EnterpriseAlert® 9

Configuring ADFS and WAP for Enterprise Alert®

PHONE +49 331 29878-0 · FAX +49 331 29878-22 · WWW.DERDACK.COM
AMTSGERICHT POTSDAM, HRB 13071 · MANAGING DIRECTOR: MATTHES DERDACK · EU VAT ID: DE 812 810 585

1 MOBILE APP AUTHENTICATION

For maximum security, the authentication mechanism used between the Derdack Mobile App (for Windows Phone, Android or iPhone) and the Enterprise Alert® backend is based on the following three modes.

Secure authentication with Derdack-specific credentials	Users can be authenticated using Derdack-specific app credentials. The implemented authentication workflow is the Authorization Code Flow according to the OAuth2 protocol.
Secure SSO authentication	Besides the authentication with Derdack-specific credentials, users can also be authenticated via SSO, which is based on Active Directory Federation Services (ADFS) and the Web Application Proxy (WAP) provided by Microsoft. The implementation of ADFS and WAP adds Single Sign-On (SSO) as well as Multi-Factor Authentication (MFA) capabilities to your Derdack App authentication workflow.
Mixed mode authentication	Depending on the deployment scenario and security policies that apply to your business environment, users have the option of either logging in using Derdack-specific credentials or by entering in their Active Directory domain credentials.

The authentication mode can be configured on the User Authentication page of the Enterprise Alert® Web App.

If SSO or mixed mode authentication needs to be used, then Active Directory Federation Services (ADFS) needs to be configured for use by Enterprise Alert®. Furthermore, if a Web Application Proxy (WAP) needs to be used to route communication between the Mobile Apps and the Enterprise Alert® backend, then this will need to be configured as well.

The steps required for enabling ADFS and/or WAP to work together with the authentication flow between the Derdack Mobile Apps and the Enterprise Alert® backend are described in the following sections.

PHONE +49 331 29878-0 · FAX +49 331 29878-22 · WWW.DERDACK.COM
AMTSGERICHT POTSDAM, HRB 13071 · MANAGING DIRECTOR: MATTHES DERDACK · EU VAT ID: DE 812 810 585

1.1 SSL Requirements

The Derdack App communicates with the Enterprise Alert® backend (Mobile API) using HTTPS i.e. communications are only supported using SSL. This ensures that the communications are secure and safe from prying eyes. It also means that you will need to install and configure an SSL certificate in IIS where Enterprise Alert® is installed.

This can either be done via the Enterprise Alert® Web App *System* > *User Authentication* > *Set Public DNS* > *Generate & Install Self-Signed Certificate in IIS*, whereby a self-signed certificate will be generated and automatically installed in IIS based on the DNS with which you have opened the Web App, or you can install your own certificate manually in IIS.

We recommend using a certificate purchased from a public CA as opposed to a self-signed certificate, otherwise users will have to install the self-signed certificate on their mobile devices before the Derdack App can be used.

Now the Derdack App is smart enough in that it can automatically download the self-signed certificate from the Enterprise Alert® backend, but this can only be done if HTTP is activated. This can be problematic if you are using the Web Application Proxy (WAP), since the WAP only supports communications over SSL. If you still would like to use the WAP as well as a self-signed certificate, you can get the mobile users to install the certificate by for example sending the certificate via email.

PHONE +49 331 29878-0 · FAX +49 331 29878-22 · WWW.DERDACK.COM
AMTSGERICHT POTSDAM, HRB 13071 · MANAGING DIRECTOR: MATTHES DERDACK · EU VAT ID: DE 812 810 585

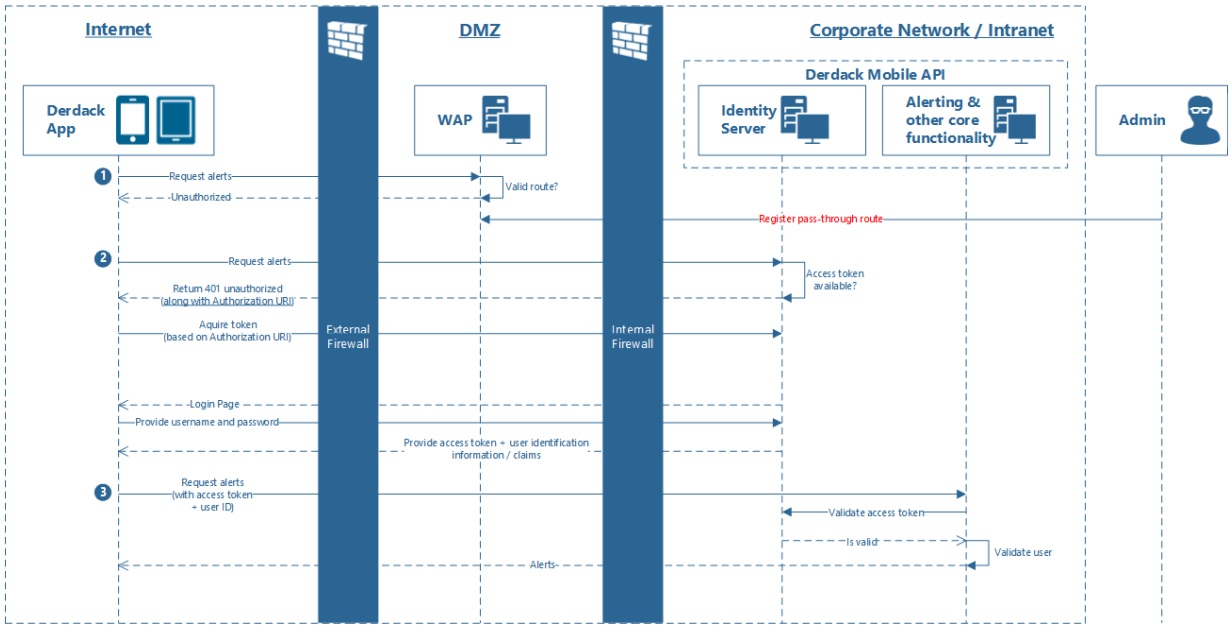
1.2 Secure authentication using Derdack-specific credentials

If you would like to use the Derdack Mobile App with Derdack-specific accounts only, you will need to set the authentication mode to "Enterprise Alert®" in the Enterprise Alert® backend as described in "Enabling User Authentication for Derdack-specific credentials in Enterprise Alert®".

If you would like to use a Web Application Proxy (WAP) as gateway between the Mobile App (typically located outside your organization on the Internet) and the Enterprise Alert® backend (typically located inside your organization behind the corporate firewall), you will then need to enable a pass-through route in your WAP described in "Configuring a pass-through route in the Web Application Proxy (Optional)".

1.2.1 Authentication Workflow for Derdack-specific Account

The basic flow of information in this authentication scenario is visualized in the following diagram:



(1) First of all, the Derdack App attempts to retrieve alerts over the Internet from the Derdack Mobile API located in corporate intranet. It tries to do this using HTTPS / SSL communication. Insecure communication using HTTP is not supported.

Assuming that a Web Application Proxy (WAP) is installed, the WAP validates whether there is a registered route for the requested URL. If not, the request is simply returned as unauthorized.
Note that the installation or use of the WAP is optional. However, it does provide an additional layer of security in that it only routes requests for routes that are registered.

Once the administrator has configured a route or so-called *Published Web Application* with the Derdack Mobile API endpoint in the WAP using pass-through preauthentication, authentication can be successfully completed.

(2) If the Derdack App then attempts to retrieve alerts, an unauthorized response will still be sent back, but this time round, the WAP returns an authorization URI with which the App can authenticate the user. The App redirects to this URI and the user enters in their credentials on a login screen provided by so-called identity server running within the Derdack Mobile API.

PHONE +49 331 29878-0 · FAX +49 331 29878-22 · WWW.DERDACK.COM
AMTSGERICHT POTSDAM, HRB 13071 · MANAGING DIRECTOR: MATTHES DERDACK · EU VAT ID: DE 812 810 585

Once the user has been authenticated by the identity server, an access token as well as user identification information are returned to the App.

(3) The Derdack App then attempts to retrieve the alerts again, but this time round, it also includes the access token and user information in the request.

The Derdack Mobile API the validates the access token and user information and then responds with the corresponding alert information associated with the user.

Any further requests made using the Derdack App skip steps 1 & 2 and proceed as described in step 3.

1.2.2 Enabling User Authentication for Derdack-specific credentials in Enterprise Alert®

On the Enterprise Alert® machine, double click on the Enterprise Alert® icon on the desktop or open a browser and type in the following URL to load the portal:

http://localhost/EAPortal/

Log in using the following credentials:

- User name: Administrator
- Password: *<the password you entered during the setup process, otherwise leave empty>*

After you have logged in, open the User Authentication settings under *System* > *User Authentication.*

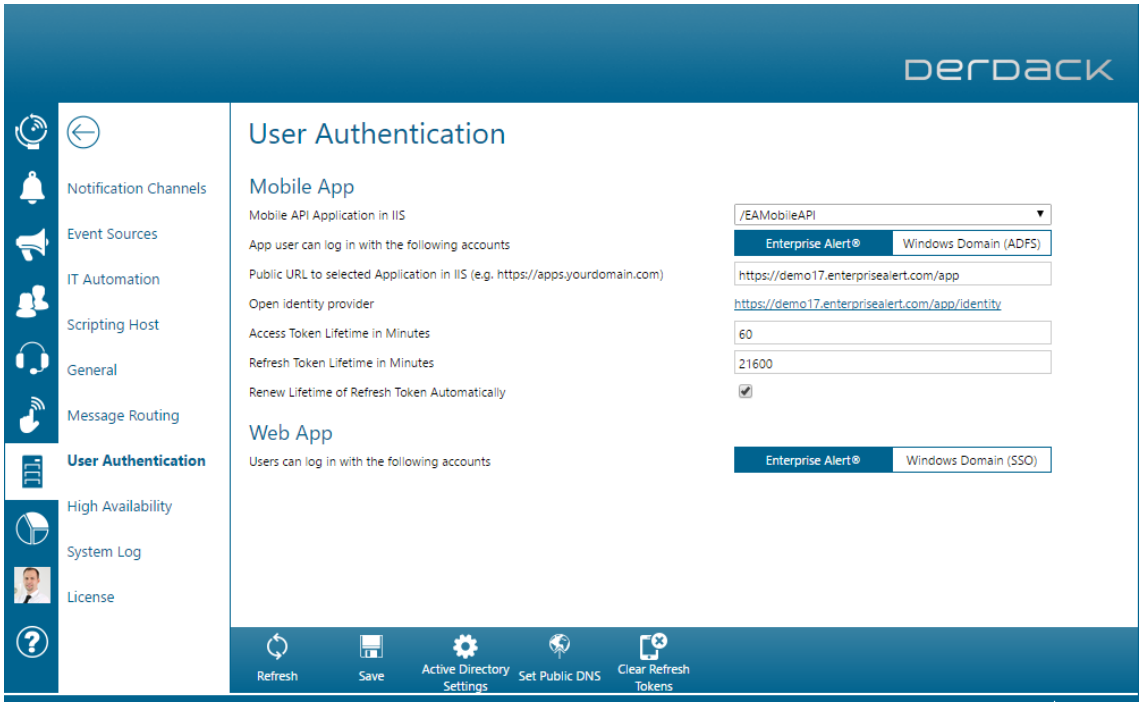
Next, ensure that only the *Enterprise Alert®* option is selected for the setting *App user can log in with the following accounts*.

Enter in the public URL of the Mobile API of the Enterprise Alert® installation and click *Save*. The application name of the Mobile API virtual directory is EAMobileAPI by default.

To test the accessibility of the identity provider (used for user authentication using Derdack-specific credentials), click on the link below the public URL you have entered.

The following screenshot shows an example configuration:

PHONE +49 331 29878-0 · FAX +49 331 29878-22 · WWW.DERDACK.COM
AMTSGERICHT POTSDAM, HRB 13071 · MANAGING DIRECTOR: MATTHES DERDACK · EU VAT ID: DE 812 810 585



1.2.3 Configuring a pass-through route in the Web Application Proxy (Optional)

ADFS authentication is not used for user authentication using Derdack-specific credentials. There is therefore nothing specific which needs to be configured in ADFS.

If the Enterprise Alert® backend runs on a server located inside your organization i.e. behind the corporate firewall, then you can use a Web Application Proxy (WAP) to provide selective access to the backend for Mobile App end users located outside of your organization or on the Internet. You can do this be specifying a pass-through route in the WAP.

To install such a route, you can use the PowerShell script *WAP_RegisterEAMobileAPI_PassThrough.ps1* located in the installation folder of Enterprise Alert® under the *AdminTools* subfolder (Default: *C:\Program Files\Enterprise Alert\AdminTools*). Note that you will have to execute the script on the WAP's server using administrator privileges.

The public URL of your Enterprise Alert® Mobile API, the backend URL of your Enterprise Alert® Mobile API and the thumbprint of the certificate used for SSL communication needs to be specified for each of the following parameters respectively: -mobileApiBackendUrl, -mobileApiPublicUrl, -sslCertThumbprint.

Example:

powershell.exe -File ".\WAP_RegisterEAMobileAPI_PassThrough.ps1"
-mobileApiBackendUrl "https://internalhost/EAMobileAPI/"
-mobileApiBackendUrl "https://publicEA/EAMobileAPI/"
-sslCertThumbprint 54464688722BF0215164B4E156241F0164121C

The script registers or updates the application "Enterprise Alert Mobile API" in your WAP.

The certificate with the specified thumbprint needs to be installed in the personal certificate store of the computer account on the server where the WAP is running.

PHONE +49 331 29878-0 · FAX +49 331 29878-22 · WWW.DERDACK.COM
AMTSGERICHT POTSDAM, HRB 13071 · MANAGING DIRECTOR: MATTHES DERDACK · EU VAT ID: DE 812 810 585

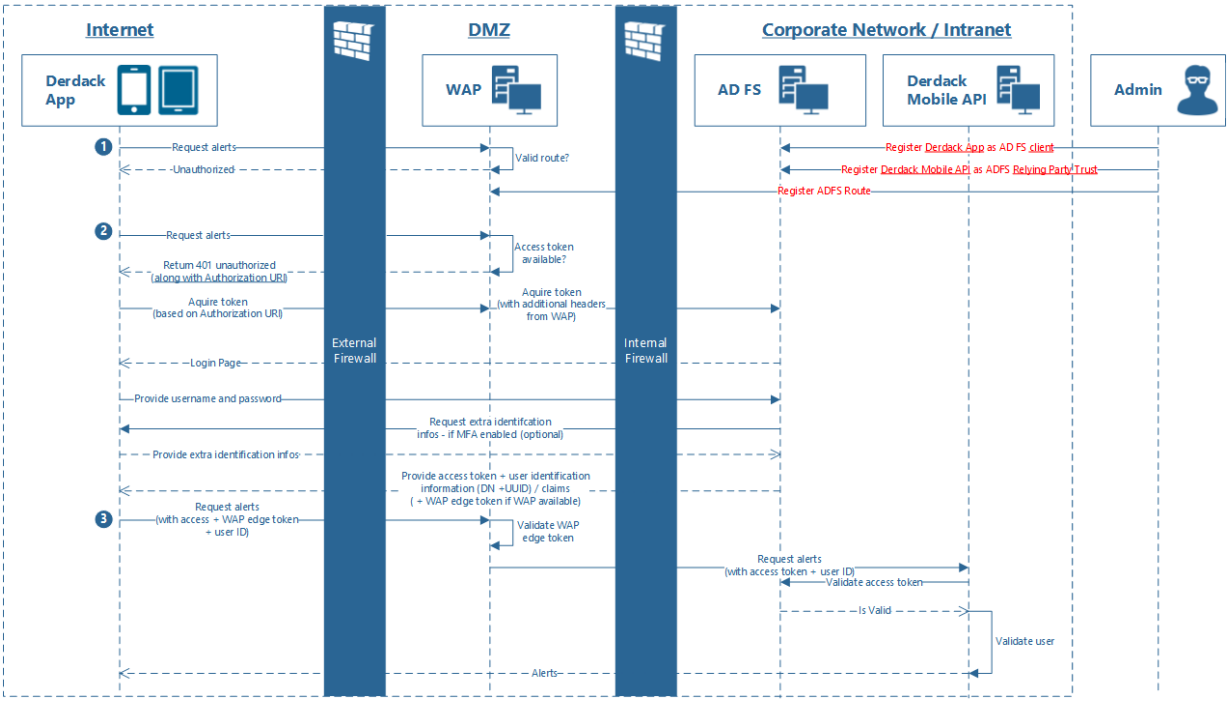
1.3 Secure SSO authentication via ADFS

If you would like to use the Derdack Mobile App with Active Directory accounts only, then you will need to complete the following steps:

1. Set the authentication mode in the Enterprise Alert® backend to activate authentication via ADFS described in "Enabling SSO authentication via ADFS in Enterprise Alert®".
2. Register the Enterprise Alert® Mobile API as an ADFS Relying Party Trust in your ADFS as described in "Registering Enterprise Alert® Mobile API as a Relying Party Trust in ADFS".
3. Register the Enterprise Alert Mobile App as an ADFS client in your ADFS as described in "Register Derdack Mobile App as ADFS Client".
4. If you would like to use a Web Application Proxy (WAP) as gateway between the Mobile App and the Enterprise Alert® backend, you will need to register an ADFS route in your WAP described in "Configuring a pass-through route in the Web Application Proxy (Optional)".

1.3.1 Authentication Workflow via ADFS

The basic flow of information in this authentication scenario is visualized in the following diagram:



(1) First of all, the Derdack App attempts to retrieve alerts over the Internet from the Derdack Mobile API located in corporate intranet. It tries to do this using HTTPS / SSL communication. Insecure communication using HTTP is not supported.

Assuming that a Web Application Proxy (WAP) is installed, the WAP validates whether there is a registered route for the requested URL. If not, the request is simply returned as unauthorized.
Note that the installation or use of the WAP is optional. However, it does provide an additional layer of security in that it only routes requests for routes that are registered and secondly, it performs certain preauthentication operations in conjunction with AD FS.

Once the administrator has registered the Derdack App as well as the Derdack Mobile API as ADFS Relying Party Trust in ADFS and then furthermore configured a route with the Derdack Mobile API endpoint and ADFS Relying Party Trust in the WAP, authentication can be successfully completed.
The route itself or so-called *Published Web Application* will need to be configured to use AD FS preauthentication and not pass-through preauthentication.
It goes without saying that ADFS will only generate access tokens for third party APIs that it trusts, which is why the registration of the Derdack Mobile API as Relying Party Trust is necessary. Registering the Derdack App as an ADFS client is also necessary so that ADFS only generates access tokens for those clients, who are authorized to work together with the "Relying Party Trusts". The Derdack App identifies itself as a valid ADFS client by sending a client ID to ADFS when authentication occurs.

(2) If the Derdack App then attempts to retrieve alerts, an unauthorized response will still be sent back, but this time round, the WAP returns an authorization URI with which the App can authenticate the user. The App redirects to this URI and the user enters in their credentials on a login screen provided by ADFS (the login screen is not part of the Derdack solution and Derdack has no influence over how users are actually authenticated by ADFS – how Windows domain authentication over the Internet via ADFS is implemented is best left over to Microsoft).

Once the user has been authenticated, an access token and a so-called edge token for preauthentication in the WAP are returned to the App along with information for identifying the user, specifically the distinguished name (DN) of the user as well as the user ID represented as a universal unique identifier (UUID).

(3) The Derdack App then attempts to retrieve the alerts again, but this time round, it also includes the tokens and user information in the request.

The WAP then validates the edge token and if valid, the request is passed on to the Derdack Mobile API. Finally, the Derdack Mobile API validates the access token and user information and then responds with the corresponding alert information associated with the user.

Any further requests made using the Derdack App skip steps 1 & 2 and proceed as described in step 3.

1.4 Enabling SSO authentication via ADFS in Enterprise Alert®

On the Enterprise Alert® machine, double click on the Enterprise Alert® icon on the desktop or open a browser and type in the following URL to load the portal:

http://localhost/EAPortal/

Log in using the following credentials:

- User name: Administrator
- Password: *<the password you entered during the setup process, otherwise leave empty>*

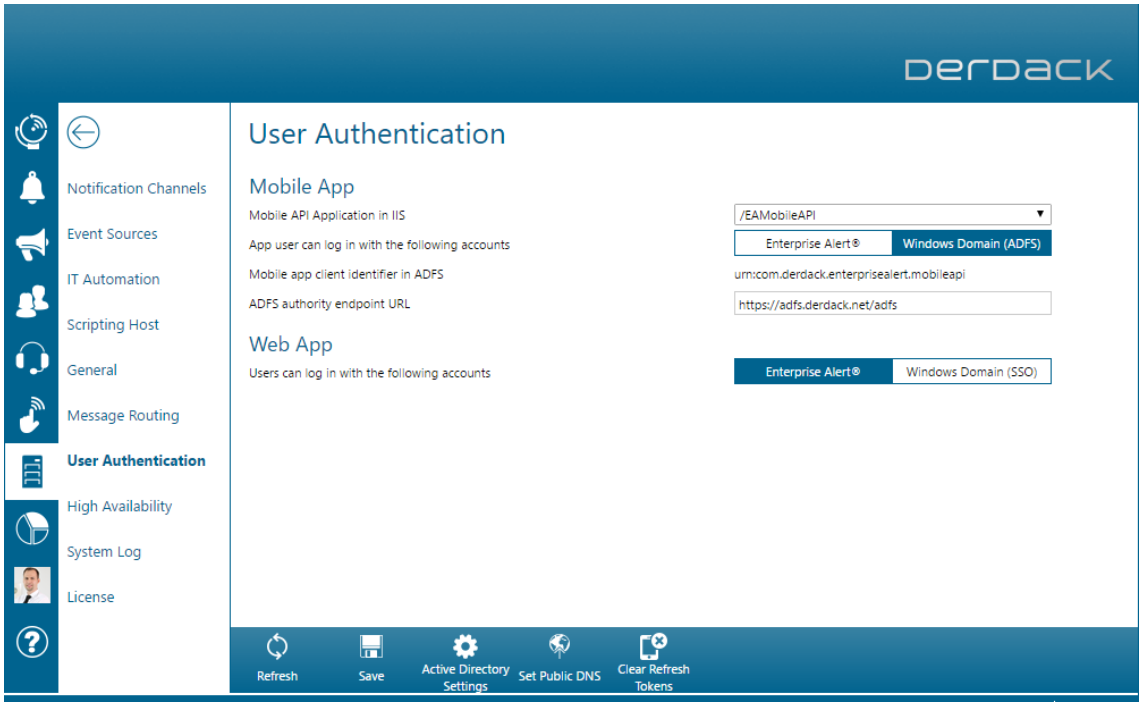
After you have logged in, open the User Authentication settings under *System* > *User Authentication.*

Next, ensure that only the *Windows Domain (ADFS)* option is selected for the setting *App user can log in with the following accounts*.

Enter in the public URL to your ADFS endpoint and click *Save*.

The following screenshot shows an example configuration:

PHONE +49 331 29878-0 · FAX +49 331 29878-22 · WWW.DERDACK.COM
AMTSGERICHT POTSDAM, HRB 13071 · MANAGING DIRECTOR: MATTHES DERDACK · EU VAT ID: DE 812 810 585



1.4.1 Registering the Derdack Mobile App and Enterprise Alert® Mobile API in ADFS

For authentication via ADFS, you will need to first of all register the Enterprise Alert® Mobile API as a Relying Party Trust and secondly, you will need to register the Derdack Mobile App as an ADFS client.

Registering Enterprise Alert® Mobile API as a Relying Party Trust in ADFS

To register the Enterprise Alert® Mobile API as a Relying Party Trust, you can use the PowerShell script *ADFS_RegisterEAMobileAPI.ps1* located in the installation folder of Enterprise Alert® under the *AdminTools* subfolder (Default: *C:\Program Files\Enterprise Alert\AdminTools*). Note that you will need to execute the script on your ADFS server using administrator privileges.

The public URL of your Enterprise Alert® Mobile API needs to be specified in the parameter -mobileApiUrl.

Example:

powershell.exe -File ".\ADFS_RegisterEAMobileAPI.ps1"
-mobileApiUrl https://myenterpriseAlert.com/EAMobileAPI

The script registers or updates "Enterprise Alert Mobile API" as Relying Party Trust in your ADFS.

Register Derdack Mobile App as ADFS Client

To register the Derdack Mobile App as an ADFS client, you can use the PowerShell script *ADFS_RegisterEAMobileApp.ps1* located in the installation folder of Enterprise Alert® under the *AdminTools* subfolder (Default: *C:\Program Files\Enterprise Alert\AdminTools*). Note that you will need to execute the script on your ADFS server using administrator privileges.

This script does not require any parameters.

Example:

powershell.exe -File ".\ADFS_RegisterEAMobileApp.ps1"

The script registers or updates "Enterprise Alert Mobile App" as ADFS client in your ADFS.

1.4.2 Configure an ADFS route in Web Application Proxy (Optional)

If you would like to use a Web Application Proxy (WAP) as gateway between the Mobile App (typically located outside the your organization on the Internet) and the Enterprise Alert® backend (typically located inside your organization behind the corporate firewall), you will need to enable a ADFS route in your WAP.

To install such route you can use the PowerShell script *WAP_RegisterEAMobileAPI_ADFS.ps1* located in the installation folder of Enterprise Alert® under the *AdminTools* subfolder (Default: *C:\Program Files\Enterprise Alert\AdminTools*). Note that you will need to execute the script on your WAP server using administrator privileges.

The public URL of your Enterprise Alert® Mobile API, the backend URL of your Enterprise Alert® Mobile API and the thumbprint of the certificate used for SSL communication needs to be specified for each of the following parameters respectively: *-mobileApiBackendUrl, -mobileApiPublicUrl, -sslCertThumbprint*.

Example:

powershell.exe -File ".\WAP_RegisterEAMobileAPI_ADFS.ps1"
-mobileApiBackendUrl "https://internalhost/EAMobileAPI/"
-mobileApiBackendUrl "https://publicEA/EAMobileAPI/"
-sslCertThumbprint 54464688722BF0215164B4E156241F0164121C

The script registers or updates the application "Enterprise Alert Mobile API" in your WAP.

The certificate with the specified thumbprint needs to be installed in the personal certificate store of the computer account on the server where the WAP is running.

PHONE +49 331 29878-0 · FAX +49 331 29878-22 · WWW.DERDACK.COM
AMTSGERICHT POTSDAM, HRB 13071 · MANAGING DIRECTOR: MATTHES DERDACK · EU VAT ID: DE 812 810 585

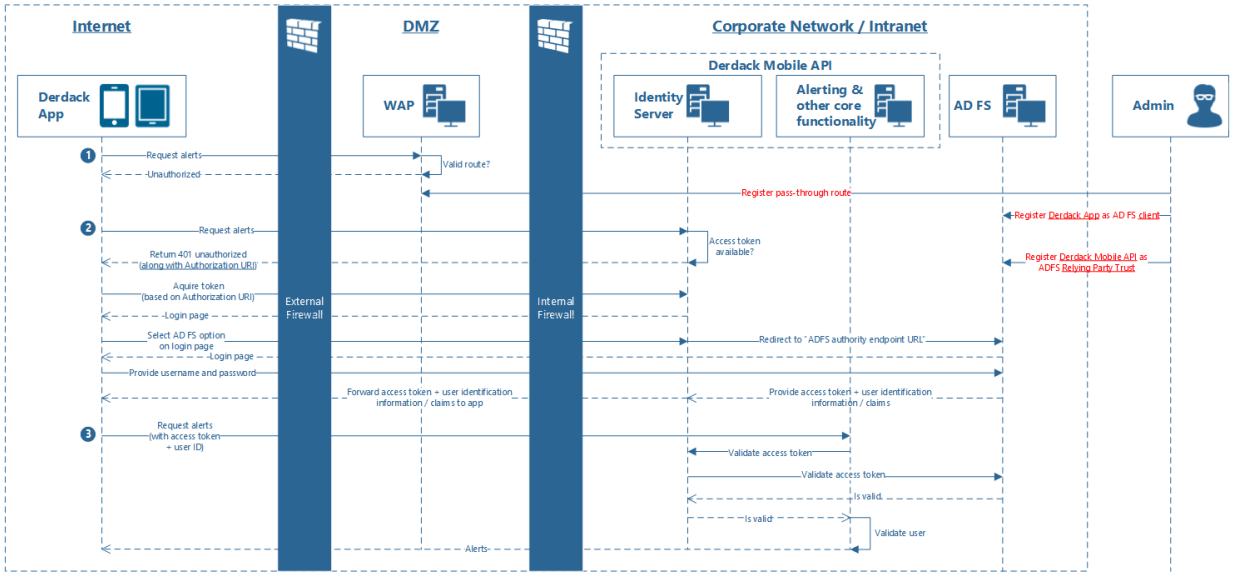
1.5 Mixed mode authentication (Derdack-Specific AND ADFS)

If you would like to use the Derdack Mobile App with mixed mode authentication (AD accounts and Derdack-specific accounts), you will need to complete the following steps.

1. Set the authentication mode in the Enterprise Alert® backend to mixed mode authentication described in "5.2. Enabling mixed mode authentication in Enterprise Alert®".
2. Register the Enterprise Alert® Mobile API as ADFS Relying Party Trust in your ADFS as described in "4.3.1 Registering Enterprise Alert® Mobile API as a Relying Party Trust in ADFS".
3. Register the Enterprise Alert Mobile App as ADFS client in your ADFS as described in "4.3.2 Register Derdack Mobile App as ADFS Client".
4. If you would like to use a Web Application Proxy as gateway between the Mobile App and the Enterprise Alert® backend. you will need to register a pass-through route in your WAP described in "3.3. Configuring a pass-through route in the Web Application Proxy (Optional)".

1.5.1 Mixed Authentication Workflow

The basic flow of information in this authentication scenario is visualized in the following diagram:



This workflow is based on a mixture of the workflows described in "Authentication Workflow for Derdack-specific Account" and "Authentication Workflow via ADFS". If the user would like to log in to the system with a Derdack-specific account, then the authentication workflow is exactly as described in "Authentication Workflow for Derdack-specific Account". Otherwise this workflow specifically concentrates on how the user can authenticate via ADFS where Derdack-specific account authentication is also enabled.

(1) Instead of the WAP performing preauthentication of the WAP edge token as was the case for the "authentication via ADFS only" configuration, the WAP simply routes traffic through the internal firewall based on its allowed routes or *Published Web Applications*. As with the previous two configurations, the use of the WAP is optional. If it is used however, a route with the pass-through preauthentication method will need to be configured and not the AD FS preauthentication method as was the case with the ADFS-only configuration.

(2) If an access token is not available, the identity server renders a login screen. The difference here is that the "Windows Authentication" button will be available. When the user selects this button, the user is redirected to the ADFS login page based on the "ADFS authority endpoint URL" of the configuration described in the next section.

After the user has entered in their Windows domain credentials, ADFS generates an access token, which the identity server then forwards back to the Derdack app. The identity server in this instance operates as a proxy between the Derdack app and ADFS.

(3) Once the access token is available, the Derdack app can finally successfully retrieve the alerts for the user. The access token which it uses to do so is validated by the identity server via ADFS.

1.5.2 Enabling mixed mode authentication in Enterprise Alert®

On the Enterprise Alert® machine, double click on the Enterprise Alert® icon on the desktop or open a browser and type in the following URL to load the portal:

http://localhost/EAPortal/

Log in using the following credentials:

- User name: Administrator
- Password: *<the password you entered during the setup process, otherwise leave empty>*

After you have logged in, open the User Authentication settings under *System* > *User Authentication.*

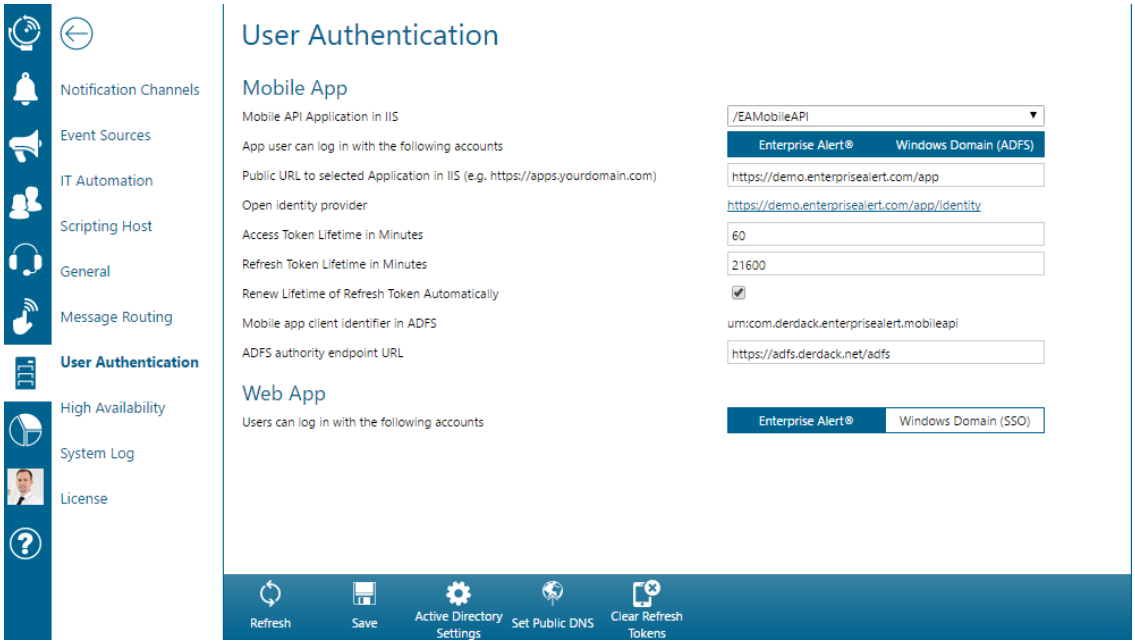
Next, select both the *Enterprise Alert®* and the *Windows Domain (ADFS)* options for the setting *App user can log in with the following accounts*.

Next, enter in the public URL of the Mobile API of the Enterprise Alert® installation, the public URL to your ADFS endpoint and click *Save*. The application name of the Mobile API virtual directory is EAMobileAPI by default.

To test the accessibility of the identity provider (used for Derdack-specific authentication in Enterprise Alert®), click on the link below your entered public URL.

The following screenshot shows an example configuration:

PHONE +49 331 29878-0 · FAX +49 331 29878-22 · WWW.DERDACK.COM
AMTSGERICHT POTSDAM, HRB 13071 · MANAGING DIRECTOR: MATTHES DERDACK · EU VAT ID: DE 812 810 585

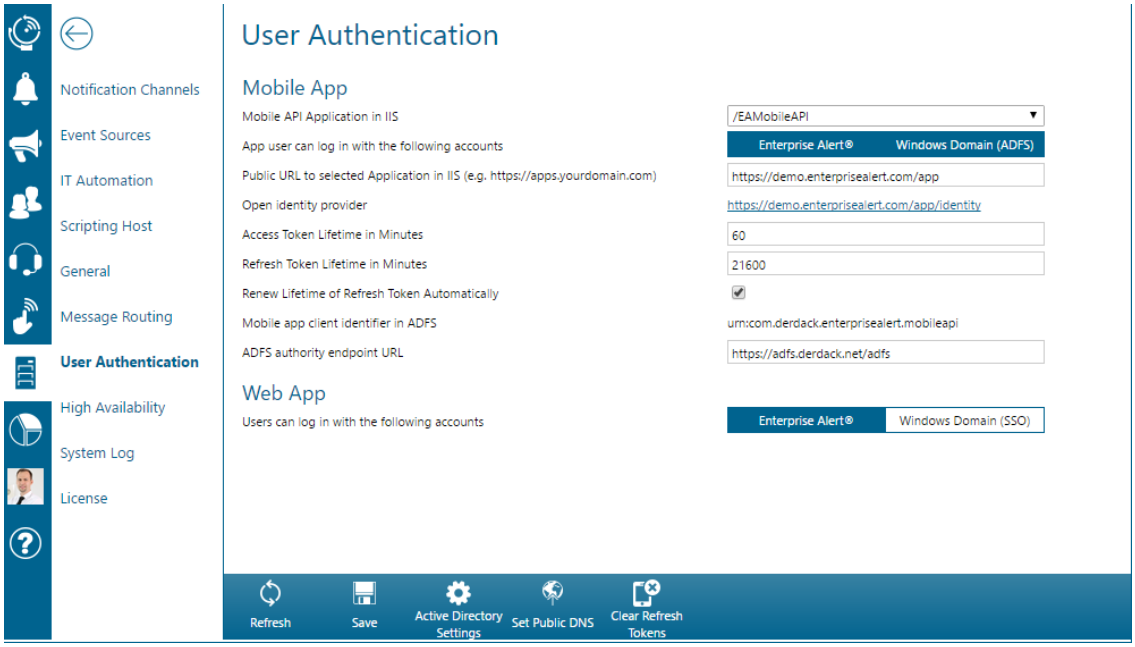


PHONE +49 331 29878-0 · FAX +49 331 29878-22 · WWW.DERDACK.COM
AMTSGERICHT POTSDAM, HRB 13071 · MANAGING DIRECTOR: MATTHES DERDACK · EU VAT ID: DE 812 810 585

1.6 Configuring Access And Refresh Token Lifetimes

1.6.1 Changing Token Lifetimes in the Enterprise Alert® Identity Provider

In order to configure the token lifetimes of the identity provider in Enterprise Alert® (used for secure authentication with Derdack-specific credentials), you can simply change the lifetime values on the User Authentication page of the Enterprise Alert® Web App.



1.7 Changing Token Lifetimes in Active Directory Federation Services

If you would like to change the token lifetimes in your ADFS, you can use the PowerShell script *ADFS_SetRefreshTokenLifetime.ps1* located in the installation folder of Enterprise Alert® under the *AdminTools* subfolder (Default: *C:\Program Files\Enterprise Alert\AdminTools*). Note that you will need to execute the script on your ADFS server using administrator privileges.

The lifetime of the refresh tokens needs to be specified (in minutes) in the –tokenLifetime parameter.

Example:

powershell.exe -File ".\ADFS_SetRefreshTokenLifetime.ps1" -tokenLifetime 1440

The script changes the global ADFS properties SsoLifetime and KmsiLifetime (keep me signed in).

Both the *SsoLifetime* and *KmsiLifetime* values need to be greater than the *TokenLifetime* of the Relying Party Trust in order to provide a refresh token for the Enterprise Alert Mobile App.

For more information on token lifetimes or using workplace joined devices, please take a look at the article 'Configure Persistent Single Sign-On' - https://technet.microsoft.com/en-us/library/mt148493.aspx

2 WEB PORTAL AUTHENTICATION

If you would like to use ADFS authentication in the Enterprise Alert Web Portal, then you will need to complete the following steps:

1. Import the Users from your Active Directory with the Enterprise Alert® Active Directory Synchronizer.
2. Register the Enterprise Alert® Web Portal as an ADFS Relying Party Trust in your ADFS as described in "Registering the Enterprise Alert® Web Portal in ADFS".
3. Set the authentication mode in the Enterprise Alert® backend to activate authentication via ADFS described in "Enabling SSO authentication via ADFS in the Enterprise Alert® Web Portal".
4. If you would like to use a Web Application Proxy (WAP) as gateway between the web browser and the Enterprise Alert® Web Portal, you will need to register an ADFS route in your WAP described in "Configure an ADFS route in Web Application Proxy (Optional)".

2.1 Registering the Enterprise Alert® Web Portal in ADFS

To register the Enterprise Alert® Web Portal as a Relying Party Trust, you can use the PowerShell script *ADFS_RegisterEAPortal.ps1* located in the installation folder of Enterprise Alert® under the *AdminTools* subfolder (Default: *C:\Program Files\Enterprise Alert\AdminTools*). Note that you will need to execute the script on your ADFS server using administrator privileges.

The public URL of your Enterprise Alert® Web Portal needs to be specified in the parameter -eaPortalUrl.

Example:

```
powershell.exe -File ".\ADFS_RegisterEAPortal.ps1"
   -eaPortalUrl https://myenterpriseAlert.com/EAPortal
```

The script registers or updates "Enterprise Alert Web Portal" as Relying Party Trust in your ADFS.

2.2 Enabling SSO authentication via ADFS in the Enterprise Alert® Web Portal

On the Enterprise Alert® machine, navigate to the physical folder where the Enterprise Alert® Web Portal web application is stored. The default path is *C:\Program Files\Enterprise Alert\WebSites\EAPortal*.

Open the *web.config* file with notepad and change the following:

- In *configuration/appSettings* change the value of *ida:AuthMode* from *None* to *Adfs* and replace the *[ADFS Host]* in the value of *idadfs:MetadataEndpoint* with the host name of your ADFS Server. After the changes the rows should be look like this:

```
<add key="ida:AuthMode" value="Adfs"/>
<add key="idadfs:MetadataEndpoint"
      value="https://myAdfsHost.com/federationmetadata/2007-06/federationmetadata.xml"/>
```

- In *configuration/system.web/authorization* change the element *<allow users="?">* to *<deny users="?">*. The authorization section should be look like this after the change:

```
<authorization>
      <deny users="?"/>
</authorization>
```


PHONE +49 331 29878-0 · FAX +49 331 29878-22 · WWW.DERDACK.COM
AMTSGERICHT POTSDAM, HRB 13071 · MANAGING DIRECTOR: MATTHES DERDACK · EU VAT ID: DE 812 810 585

After you have saved your changes to the web.config the ADFS authentication is enabled for the Enterprise Alert® Web Portal.

If you now open http://localhost/EAPortal/ you will be redirected to the log-in page of your ADFS.

After you have logged-in you will be redirected back to the Enterprise Alert® Web Portal and you are also logged-in here automatically

2.3 Configure an ADFS route in Web Application Proxy (Optional)

If you would like to use a Web Application Proxy (WAP) as gateway between the web browser (typically located outside the your organization on the Internet) and the Enterprise Alert® Web Portal (typically located inside your organization behind the corporate firewall), you will need to enable a ADFS route in your WAP.

To install such route you can use the PowerShell script *WAP_RegisterEAPortal_ADFS.ps1* located in the installation folder of Enterprise Alert® under the *AdminTools* subfolder (Default: *C:\Program Files\Enterprise Alert\AdminTools*). Note that you will need to execute the script on your WAP server using administrator privileges.

The public URL of your Enterprise Alert® Web Portal, the backend URL of your Enterprise Alert® Web Portal and the thumbprint of the certificate used for SSL communication needs to be specified for each of the following parameters respectively: *-eaPortalBackendUrl, -eaPortalPublicUrl, -sslCertThumbprint*.

Example:

powershell.exe -File ".\WAP_RegisterEAPortal_ADFS.ps1"
-eaPortalBackendUrl "https://internalhost/EAPortal/"
-eaPortalPublicUrl "https://publicEA/EAPortal/"
-sslCertThumbprint 54464688722BF0215164B4E156241F0164121C

The script registers or updates the application "Enterprise Alert Web Portal" in your WAP.

The certificate with the specified thumbprint needs to be installed in the personal certificate store of the computer account on the server where the WAP is running.

3 ABOUT

Derdack is an independent software vendor offering advanced enterprise notification and rapid response software. Derdack's premium products help clients to automate alert notifications and to communication-enable business processes and applications. Derdack is recognized for its intuitive yet inspiring software products and has customers in over 50 countries worldwide. Derdack was founded in 1999 and its corporate headquarters are in Berlin, Germany.

4 FURTHER INFORMATION

Please visit www.derdack.com or:

Corporate Blog:	https://www.derdack.com/news/
Technical Blog:	https://www.derdack.com/category/technical-blog/
YouTube:	http://www.youtube.com/derdack
Facebook:	http://www.facebook.com/derdack
LinkedIn:	http://www.linkedin.com/groups?gid=1701707
Twitter:	http://twitter.com/#!/derdack

5 CONTACT

Please visit www.derdack.com for further information on Enterprise Alert® or contact us:

Germany:	+49 (331) 29878-20 (German, English, Spanish), Fax: +49 (331) 29878-22
UK:	+44 (20) 88167095
US:	+1 (202) 4700885
Email:	info@derdack.com

5.1 Mailing Address

Derdack Corp.
4470 Cox Road, Suite 250
Glen Allen, VA 23060
USA

Derdack GmbH
Friedrich-Ebert-Straße 8
14467 Potsdam
Germany

5.2 Hours of Operation

Monday – Friday 09:00 a.m. – 06:00 p.m. Central European Time (GMT+1)

Closed Saturday and Sunday and on German and local public holidays

6 DISCLAIMER

PHONE +49 331 29878-0 · FAX +49 331 29878-22 · WWW.DERDACK.COM
AMTSGERICHT POTSDAM, HRB 13071 · MANAGING DIRECTOR: MATTHES DERDACK · EU VAT ID: DE 812 810 585