



## Cyberark PVWA Authentication via Duo WebSDK v4 Technical Documentation

Name of Company: Cisco Duo

Website: [www.duo.com](http://www.duo.com)

Name of Product: Duo WebSDK

Version: Duo Web SDK v4

Date: July 2024

## SOLUTION OVERVIEW

Multifactor Authentication is crucial to secure access to Privileged Accounts stored in the CyberArk Vault. This integration allows the CyberArk GUI Password Vault Web Access (PVWA) to take advantage of Duo Authentication via the Duo Web SDK. All the feature rich Duo Authentication Capabilities provided by the browser-based Duo prompt are available to CyberArk, such as policies that limit access to users and devices that meet comply with software and security policy restrictions.

## INTEGRATION FLOW AND DESCRIPTION

The integration requires a Context which is the First Factor in the authentication process. This Context is the Duo ID of the user. The windows account that the user is logged in with can be used as the context to sign the request sent to Duo, if the context cannot be obtained for the user (in case the computer is not in the domain, or PasswordVault IIS is not configured with ASP impersonation and Windows Authentication enabled as shown later in this guide) the flow for the authentication will present a screen to enter the Duo user name to then continue.

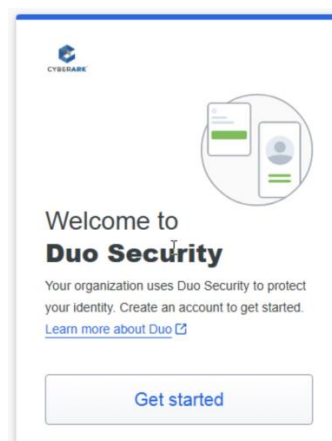
This authentication will work with CyberArk PAS v9/v10+. However only the classic UI logon screen (UseV10LogonPage set to No) is supported.

If the user (either obtained by the Windows logged-in user, or entered) doesn't exist in Duo it will start the User/Device enrollment process.

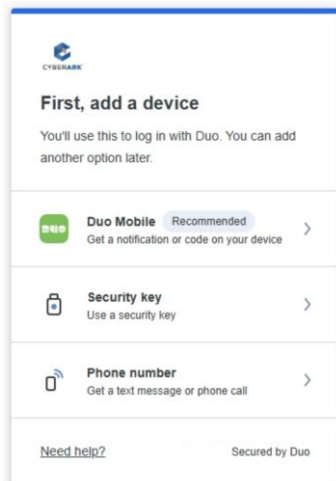
- 1) The user and device enrollment workflow will look like this:



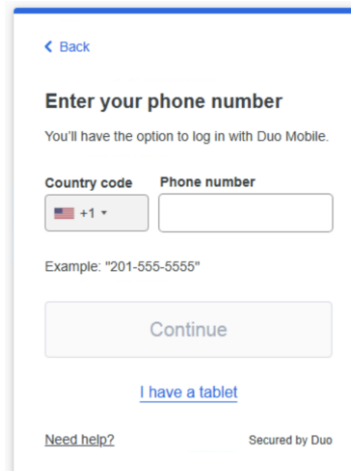
The user selects Duo Security from the Authentication Methods available to their CyberArk instance.



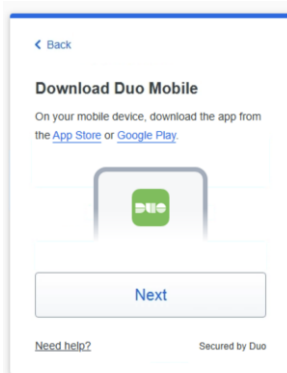
The Context is assumed from the Windows account the user is logged in.



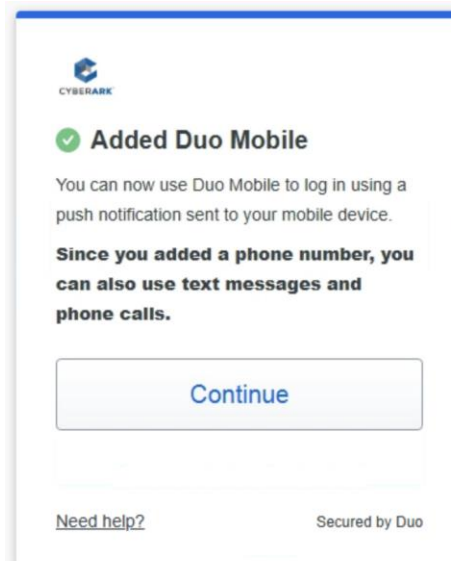
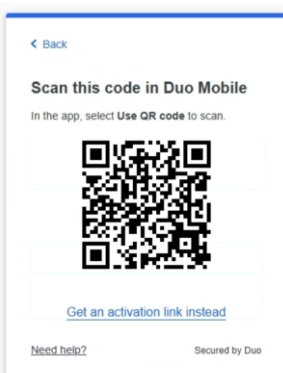
If the user doesn't exist in Duo it will start the enrollment process.



Then it will proceed to the mobile device enrollment process.



The mobile device is validated.

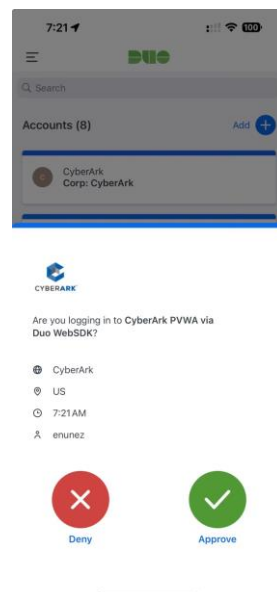
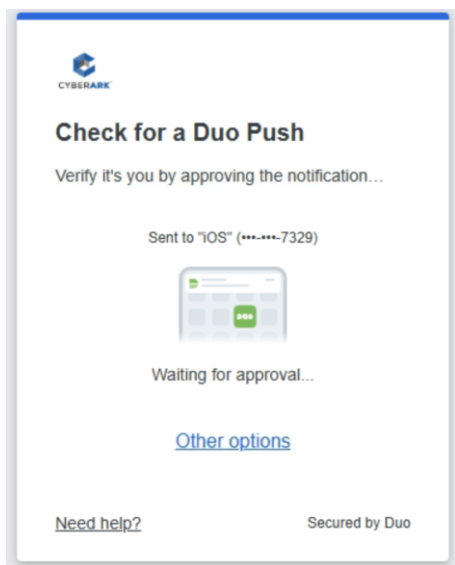


The user enrollment is complete.

The user and the mobile device is now enrolled in Duo :

The screenshot shows the Duo Users management page. On the left is a dark sidebar with navigation links: Dashboard, Policies, Applications (3), Users (5), Add User, Pending Enrollments, Bulk Enroll Users, Import Users, Directory Sync, Bypass Codes, 2FA Devices (1), Groups (0), Administrators (1), and Reports. The main content area has a search bar at the top. Below it, the 'Users' section displays summary cards: 5 Total Users (with a Trash (0) button), 2 Not Enrolled, 2 Inactive Users (last 30 days), 0 User Bypass, and 0 Locked Out. There are buttons for Directory Sync, Import Users, Bulk Enroll Users, and Add User. Below the summary cards is a table of users with columns: Username, Name, Email, Status, and Last Login. Two users are listed: 'john' (Active, Never authenticated) and 'mike' (Active, Jul 11, 2018 3:58 PM). The 'john' row is highlighted with a yellow background.

2) When a user is already enrolled, the login flow will look like this:



If Push is selected it will show up on the user's mobile device. After the user responds to the push he/she will be logged in to CyberArk.

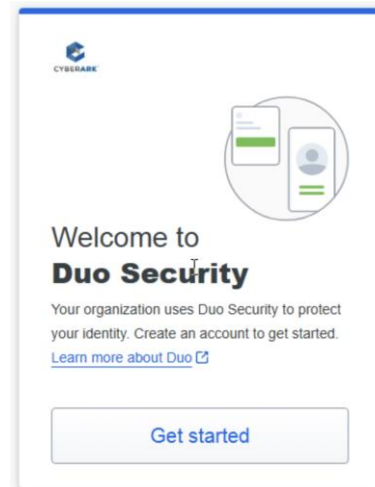
- 3) If the user is not logged into a Windows domain, no previous user context exists. Therefore, the authentication will prompt for a user name and start the enrollment process if necessary:



Enter Duo Username:

If the authentication can't obtain the Duo user ID from the context (Windows/AD user), it prompts for it.



If the user does not exist in Duo with a 2FA device it will trigger the enrollment process described earlier. Otherwise it will continue the regular login process.

The page to enter the username can be customized by the administrator by editing the file `usernameprompt.aspx` in your PVWA server under `inetpub\wwwroot\PasswordVault\auth\duosecurity\duo`.

## INTEGRATION STEPS

- 1) This step is relevant if you have v1 of the Duo WebSDK plugin and you wish to upgrade to v4 of the Duo WebSDK. Otherwise, you can skip to step 2). The caching mechanism of IIS has a known limitation where all binaries of v1 must be deleted, with no reference in `web.config`. The following steps will outline how to uninstall all v1 binary and reset IIS so the new v4 binary can be installed:

A. Delete the two dll files from "pvwa" bin folder: `CyberArk.Authentication.DuoSecurity.dll` and `DuoWeb.dll`

B. Remove "Duosecurity" Auth Method in PVWA -> Administration.

C. Edit `pvwa\web.config` and delete all references to old duo entries:

```
<add key="DuoSecurity-IntegrationKey" value="" />
```

```
<add key="DuoSecurity-SecretKey" value="" />
```

```
<add key="DuoSecurity-APIHostName" value="" />
```

```
<add key="DuoSecurity-ApplicationKey" value="" />
```

```
<add name="DuoSecurity"
type="CyberArk.Authentication.DuoSecurity,CyberArk.Authentication.DuoSecurity"
preCondition="managedHandler" />
```

D. Delete "duosecurity" folder in pvwa\auth folder.

F. Reset IIS and check PVWA is up and running.

- 2) Login to your Duo account and find the Admin Panel. Add the Web SDK to the list of protected applications in order generate the ClientID, Client Secret, and API hostname:

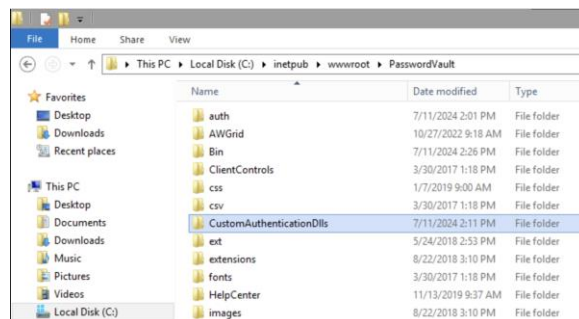
## CyberArk PVWA via Duo WebSDK

See the [Duo Web SDK Documentation](#) to integrate Duo into your custom web application.

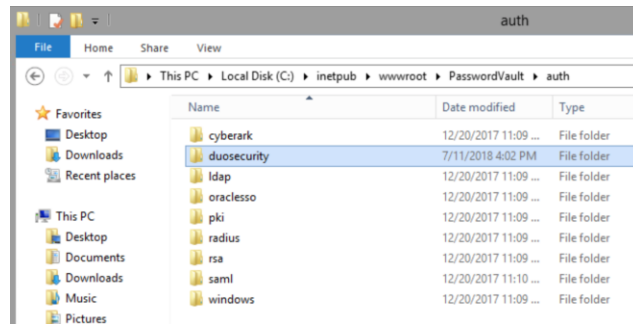
### Details

|                                                              |                                 |                      |
|--------------------------------------------------------------|---------------------------------|----------------------|
| Client ID                                                    | <div>XXXXXXXXXXXXXXXXXXXX</div> | <a href="#">Copy</a> |
| Client secret                                                | <div>XXXXXXXXXXXXXXXXXXXX</div> | <a href="#">Copy</a> |
| Don't write down your client secret or share it with anyone. |                                 |                      |
| API hostname                                                 | <div>XXXXXXXXXXXXXXXXXXXX</div> | <a href="#">Copy</a> |

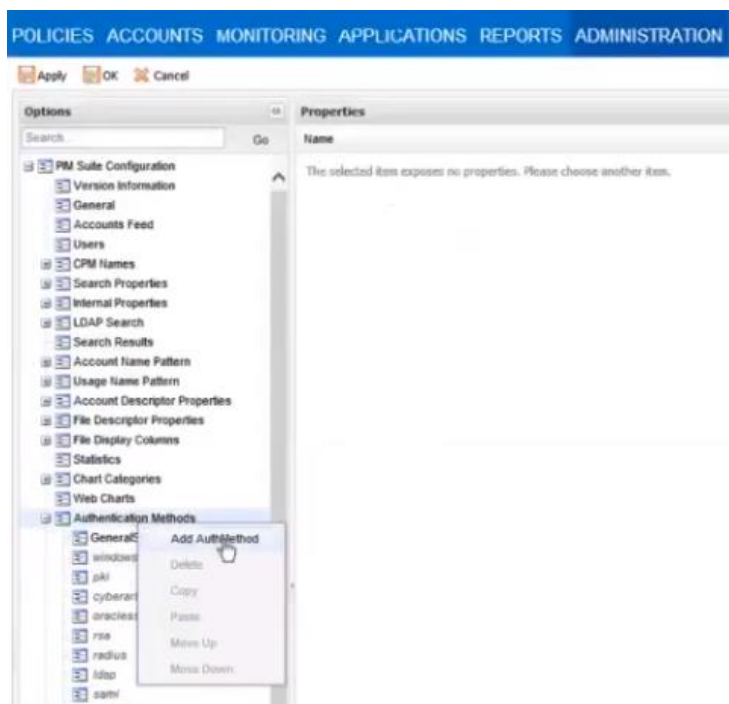
- 3) Download the distribution package from the [CyberArk Marketplace](#).
- 4) Extract the distribution package on the PVWA server:
  - a. Copy folder CustomAuthenticationDlls (the complete folder) to ...\\inetpub\\wwwroot\\PasswordVault folder.



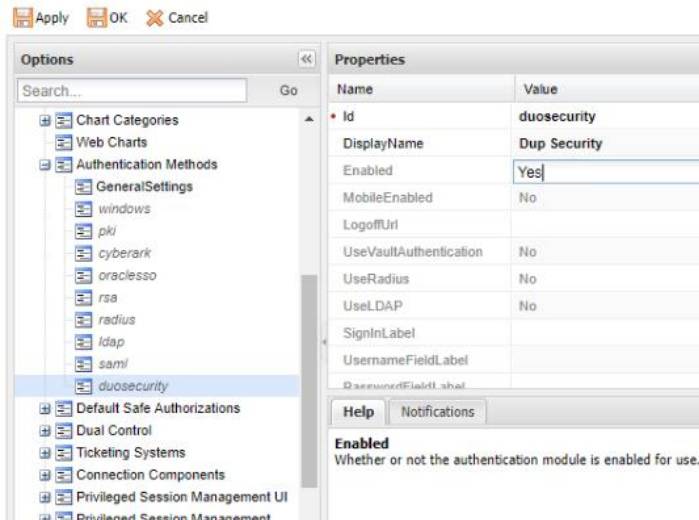
- b. Copy the **duosecurity** folder from the distribution to ...\\inetpub\\wwwroot\\PasswordVault\\auth



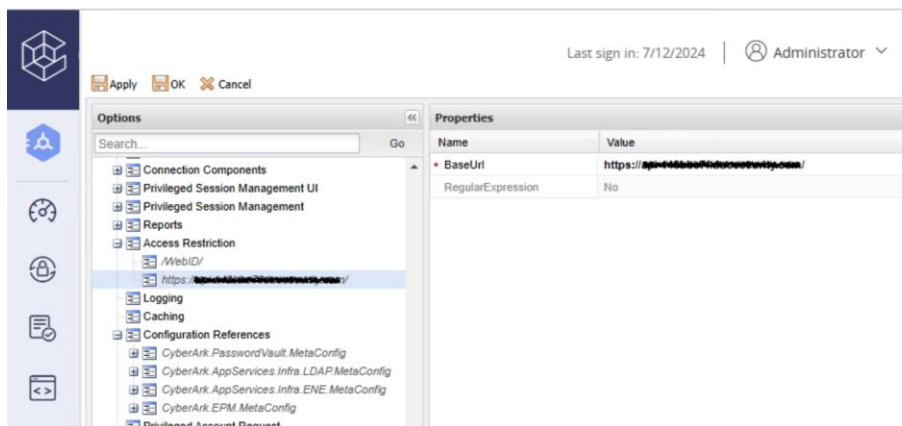
- 5) Define the **duosecurity** custom authentication method in the PVWA. Navigate to Administration -> Options, then right click on “Authentication Methods” and select “Add Auth Method”:



You must set the ID to **duosecurity**. Make sure to spell it correctly and in lowercase as this value is expected by the dll's installed earlier. Set **Enabled** to **Yes** and choose a **DisplayName** for the CyberArk login screen:



- 6) Add AllowedReferrer for the duo API hostname in the format of <https://<API-hostname>/>, in **Administration -> Configuration Options -> Options -> Access Restrictions** via right-click, then "Add AllowedReferrer"



- 7) Next, open the PVWA web.config file.
- 8) Add the keys obtained from Duo to the PVWA web.config file inside the **appSettings** section, add the following keys:

```
<add key="DuoSecurity-ClientID" value="" />
<add key="DuoSecurity-ClientSecret" value="" />
<add key="DuoSecurity-APIHostName" value="" />
<add key="DuoSecurity-ApplicationKey" value="" />
```

The DuoSecurity-ApplicationKey is an arbitrary 40 char key created by you and it will be used to sign the requests to and back from Duo.

**Note:** With the integration with Duo WebSDK v4, the first 2 keys were renamed from DuoSecurity-IntegrationKey and DuoSecurity-SecretKey; please make sure to name them as pointed out on this section. If coming from an upgrade, rename the keys from the previous values.

- 9) In the modules section of the same web.config file, add the following module:

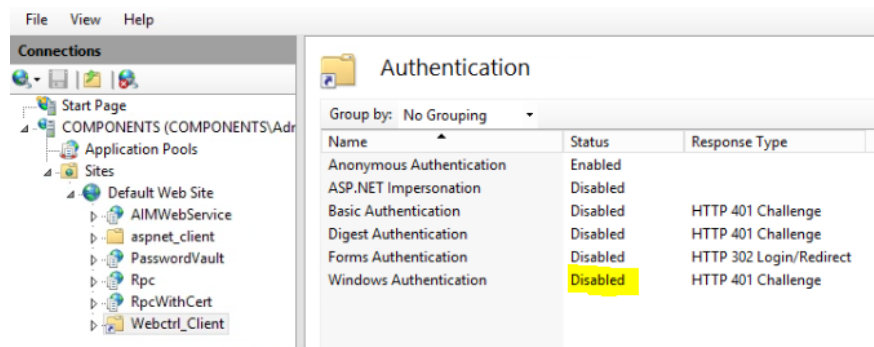
```
<add name="DuoSecurity" type="CyberArk.Authentication.DuoSecurity,CyberArk.Authentication.DuoSecurity" precondition="managedHandler" />
```

```
<modules>
  <remove name="WebDAVModule" />
  <add name="CyberArkMobileGatewayModule" type="CyberArk.MobileGatewayModule,CyberArk.MobileGatewayModule" />
  <add name="CyberArkAuth" type="CyberArk.Authentication.CyberArkAuth,CyberArk.Authentication.CyberArkAuth" />
  <add name="PKIAuth" type="CyberArk.Authentication.PKIAuth,CyberArk.Authentication.PKIAuth" />
  <add name="OAuth" type="CyberArk.Authentication.OAuth,CyberArk.Authentication.OAuth" />
  <add name="WinAuth" type="CyberArk.Authentication.WinAuth,CyberArk.Authentication.WinAuth" />
  <add name="OracleSSOAuth" type="CyberArk.Authentication.OracleSSOAuth,CyberArk.Authentication.OracleSSOAuth" />
  <add name="SAMLAuth" type="CyberArk.Authentication.SAMLAuth,CyberArk.Authentication.SAMLAuth" />
  <add name="RSAAuth" type="CyberArk.Authentication.RSAAuth,CyberArk.Authentication.RSAAuth" />
  <add name="RADIUSAuth" type="CyberArk.Authentication.RADIUSAuth,CyberArk.Authentication.RADIUSAuth" />
  <add name="LDAPAuth" type="CyberArk.Authentication.LDAPAuth,CyberArk.Authentication.LDAPAuth" />
  <add name="CyberArkGatewayModule" type="CyberArk.Authentication.GatewayModule,CyberArk.Authentication.GatewayModule" />
  <add name="WcfReadEntityBodyModule" type="CyberArk.Authentication.WcfReadEntityBodyModule,CyberArk.Authentication.WcfReadEntityBodyModule" />
  <add name="DuoSecurity" type="CyberArk.Authentication.DuoSecurity,CyberArk.Authentication.DuoSecurity" />
</modules>
```

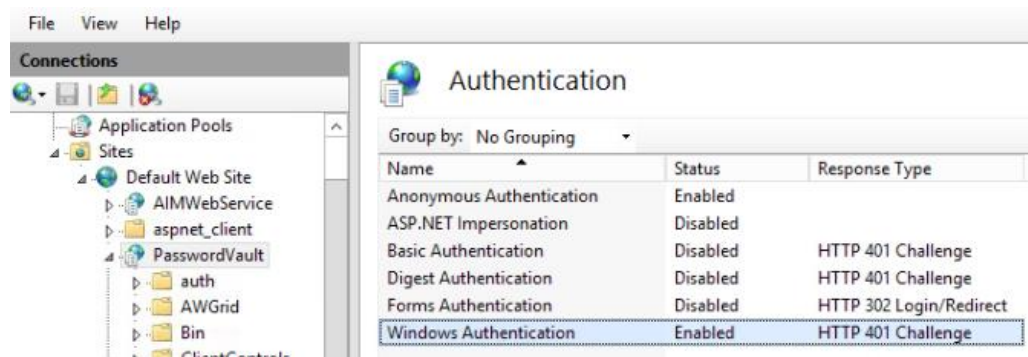
- 10) Add the following dependent assemblies to the same web.config inside the **runtime -> assemblyBinding** section:

```
<dependentAssembly>
  <assemblyIdentity name="Microsoft.IdentityModel.Tokens" publicKeyToken="31bf3856ad364e35" culture="neutral" />
  <bindingRedirect oldVersion="0.0.0.0-6.34.0.41017" newVersion="6.34.0.0" />
  <codeBase version="6.34.0.0" href="CustomAuthenticationDlls\Microsoft.IdentityModel.Tokens.dll" />
</dependentAssembly>
<dependentAssembly>
  <assemblyIdentity name="System.Text.Json" publicKeyToken="cc7b13ffcd2ddd51" culture="neutral" />
  <bindingRedirect oldVersion="0.0.0.0-5.0.0.2" newVersion="5.0.0.2" />
  <codeBase version="5.0.0.2" href="CustomAuthenticationDlls\System.Text.Json.dll" />
</dependentAssembly>
<dependentAssembly>
  <assemblyIdentity name="Microsoft.Bcl.AsyncInterfaces" publicKeyToken="cc7b13ffcd2ddd51" culture="neutral" />
  <bindingRedirect oldVersion="0.0.0.0-5.0.0.0" newVersion="5.0.0.0" />
  <codeBase version="5.0.0.0" href="CustomAuthenticationDlls\Microsoft.Bcl.AsyncInterfaces.dll" />
</dependentAssembly>
<dependentAssembly>
  <assemblyIdentity name="System.Threading.Tasks.Extensions" publicKeyToken="cc7b13ffcd2ddd51" culture="neutral" />
  <bindingRedirect oldVersion="0.0.0.0-4.2.0.1" newVersion="4.2.0.1" />
  <codeBase version="4.2.0.1" href="CustomAuthenticationDlls\System.Threading.Tasks.Extensions.dll" />
</dependentAssembly>
<dependentAssembly>
  <assemblyIdentity name="System.Text.Encodings.Web" publicKeyToken="cc7b13ffcd2ddd51" culture="neutral" />
  <bindingRedirect oldVersion="0.0.0.0-5.0.0.1" newVersion="5.0.0.1" />
  <codeBase version="5.0.0.1" href="CustomAuthenticationDlls\System.Text.Encodings.Web.dll" />
</dependentAssembly>
<dependentAssembly>
  <assemblyIdentity name="Microsoft.IdentityModel.JsonWebTokens" publicKeyToken="31bf3856ad364e35" culture="neutral" />
  <bindingRedirect oldVersion="0.0.0.0-6.34.0.41017" newVersion="6.34.0.0" />
  <codeBase version="6.34.0.0" href="CustomAuthenticationDlls\Microsoft.IdentityModel.JsonWebTokens.dll" />
</dependentAssembly>
<dependentAssembly>
  <assemblyIdentity name="Microsoft.IdentityModel.Logging" publicKeyToken="31bf3856ad364e35" culture="neutral" />
  <bindingRedirect oldVersion="5.3.0.0-6.34.0.41017" newVersion="6.34.0.0" />
  <codeBase version="6.34.0.0" href="CustomAuthenticationDlls\Microsoft.IdentityModel.Logging.dll" />
</dependentAssembly>
<dependentAssembly>
  <assemblyIdentity name="Microsoft.IdentityModel.Abstractions" publicKeyToken="31bf3856ad364e35" culture="neutral" />
  <bindingRedirect oldVersion="5.3.0.0-6.34.0.41017" newVersion="6.34.0.0" />
  <codeBase version="6.34.0.0" href="CustomAuthenticationDlls\Microsoft.IdentityModel.Abstractions.dll" />
</dependentAssembly>
<dependentAssembly>
  <assemblyIdentity name="System.Net.Http.Json" publicKeyToken="cc7b13ffcd2ddd51" culture="neutral" />
  <bindingRedirect oldVersion="5.0.0.0-5.0.0.41017" newVersion="5.0.0.0" />
  <codeBase version="5.0.0.0" href="CustomAuthenticationDlls\System.Net.Http.Json.dll" />
</dependentAssembly>
```

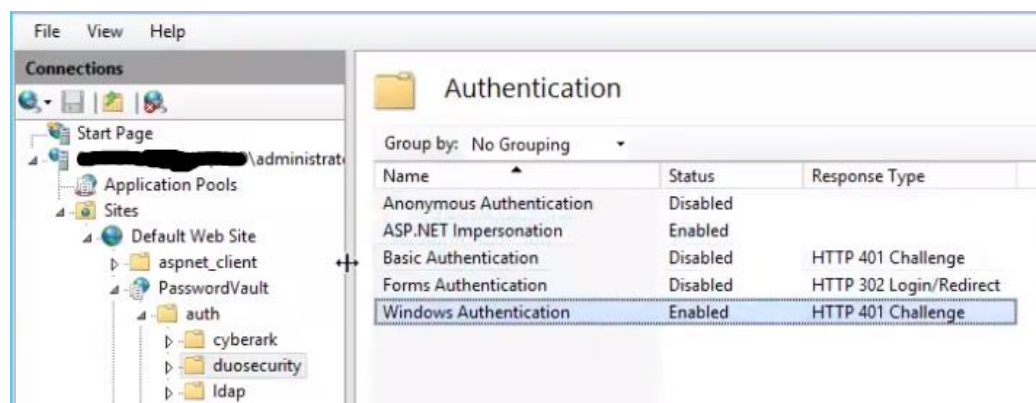
11) On the IIS manager navigate to **Webctrl\_Client** and disable **Windows Authentication**:



12) Navigate to the PasswordVault website, disable **Forms Authentication**, and enable **Windows Authentication**:



13) Navigate to **duosecurity**, enable **Windows Authentication** and **ASP.NET Impersonation**:



14) Enable web browsers to forward Microsoft Windows authentication.

**a. Google Chrome, Microsoft Internet Explorer, and Edge**

- Click **Windows Start** menu > **Settings** > **Internet Options**.
- Under the **Security** tab, go to **Trusted sites** > **Custom level**.
- Enable **Automatic logon with current username and password** and the **Enable Integrated Windows Authentication** under Advanced tab options.
- Go back to **Trusted sites** and under **Sites**, add the CyberArk PVWA URL as a trusted site.

**Note:** You may need to reset browser cache after enabling those options in Internet Options.

**b. Firefox**

- i. Locate and edit the following preferences so they include a comma-separated list of URL prefixes or domains for the location of your CyberArk PVWA server(s).
  1. network.automatic-ntlm-auth.trusted-uris
  2. network.negotiate-auth.delegation-uris
  3. network.negotiate-auth.trusted-uris

## INCORPORATING ADDITIONAL AUTHENTICATION FACTORS

If you need to enable additional authentication factors in CyberArk, to be invoked after Duo, edit the definition of **duosecurity** created on step 3. If you enable both **UseVaultAuthentication** and **UseLDAP**, users will be prompted for their LDAP password before allowing them to login to CyberArk:

POLICIES ACCOUNTS MONITORING APPLICATIONS REPORTS ADMINISTRATION

Apply OK Cancel

Options

Search... Go

Authentication Methods

- GeneralSettings
- cyberark
- ldap
- saml
- pki
- radius
- windows
- rsa
- oracle
- duosecurity

Properties

| Name                   | Value        |
|------------------------|--------------|
| Id                     | duosecurity  |
| DisplayName            | Duo Security |
| Enabled                | Yes          |
| MobileEnabled          | No           |
| LogoutUrl              |              |
| UseVaultAuthentication | Yes          |
| UseRadius              | No           |
| UseLDAP                | Yes          |
| SignInLabel            |              |

Now, after the user completes the Duo Authentication it will be challenged with his/her LDAP password:



**SIGN IN**

Specify your authentication details

John

Password

Sign in

Copyright © 1999-2018 CyberArk Software Ltd. All Rights Reserved.  
Version 10.3.0 (10.3.0.16) [About](#) | [Mobile version](#)

## USING THE CYBERARK V10 INTERFACE

The v10 logon page currently doesn't work with this integration. To access the v10 interface you will need to login to the PVWA using the classic UI logon page and then selecting the Accounts views V10 interface as your default page.

- 1) Login as Administrator and set the **UseV10LogonPage** parameter to **No**:

The screenshot shows the PVWA Administration console with the 'ADMINISTRATION' tab selected. The 'Options' pane on the left shows 'PM Suite Configuration' expanded, with 'General' selected. The 'Properties' pane on the right displays a table of system parameters:

| Name                               | Value          |
|------------------------------------|----------------|
| LoadBalancerFullPath               |                |
| PublicDataSafe                     | PVWAPublicData |
| MaxGridColumnCount                 | 100            |
| KeepIdleSessionsWithOpenConnection | No             |
| AddAntiCSRFToken                   | Yes            |
| UseV10LogonPage                    | No             |

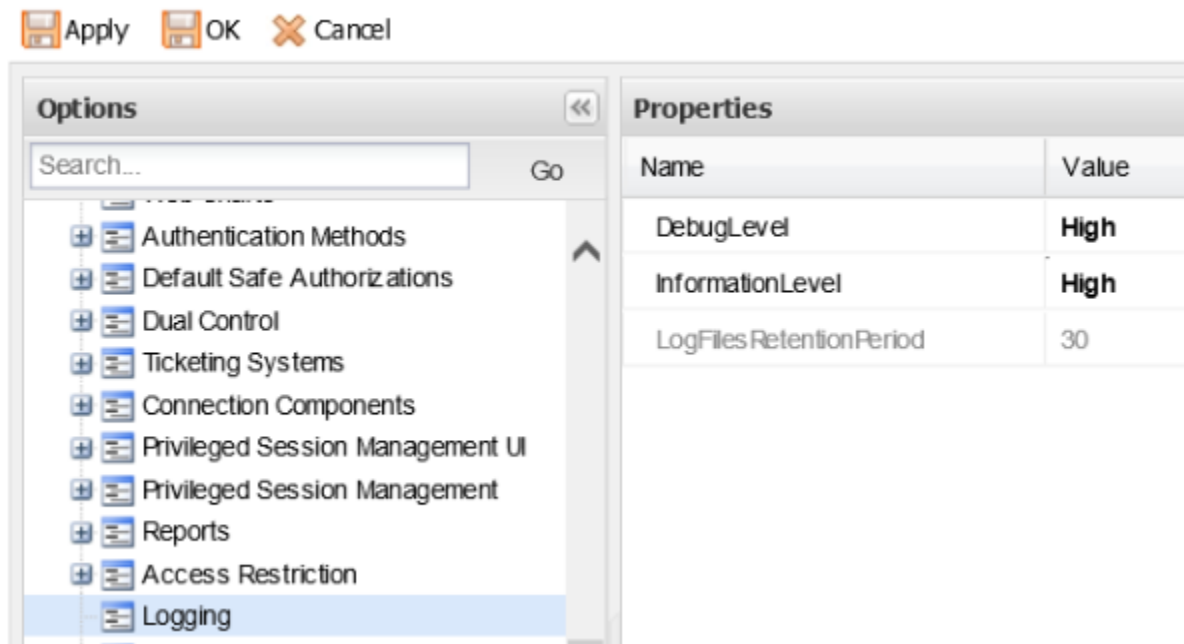
- 2) To set the Default page, select the Accounts view (Classic UI) then click on **Customize**:

The screenshot shows the 'Customize' page in the PVWA. The 'User details' section shows the user is 'Administrator'. The 'Preferences' section shows the 'Default Page' is set to 'Account views V10 interface' (indicated by a dropdown arrow). Other options include 'Objects per page' set to 'Fit page' and an unchecked checkbox for 'Clear all "connect with account" address history'. The 'Change Password' section has empty input fields for 'Current Password', 'New Password', and 'Confirm Password'. 'Save' and 'Cancel' buttons are at the bottom.

## DEBUGGING

If you're having trouble and need to gather additional information, you will find details of what's causing the issue in the **CyberArk.WebApplication.log** file located on **C:\Windows\Temp\PVWA**.

Additional debugging options are available in the Logging section of the PVWA System Configuration Options:



You can enable additional debugging for the integration by setting the following key in the PVWA web.config (in section appsettings):

```
<add key="DuoSecurity-Debug" value="true" />
```

A file named "DuoSecurityLogFile.txt" in **C:\Windows\Temp** will have debugging messages.

## SUPPORT INFO

<https://www.cyberark.com/services-support/technical-support/>