Securing the Admin App (v1.x)

Overview

This section collects best practices related to security and the ODS / API Admin App. As with every application in your enterprise, you should ensure that the Admin App is part of your organization's holistic security approach, is included in periodic external security audits, and so forth. The information herein contains considerations specific to the Admin App.

Contents

General Configuration

Access

Admin App is an administrative application targeted towards IT administrators within a local or state education agency. Ideally, you should configure the system to be accessible only to the small group that requires administrative access.

Hosting

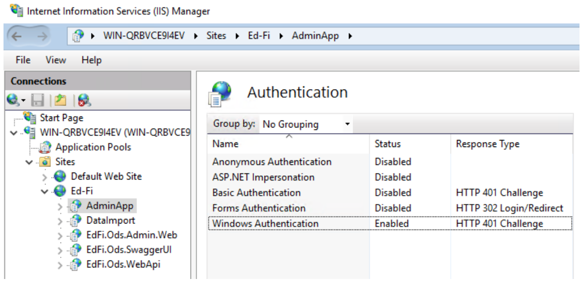
If practical, the Admin App should be available only to internal, private, or VPN-access-only networks.

On-Premises Deployments

Restricting Access to Active Directory Groups or Users

To limit, please follow the instructions below:

1.) Please ensure that "Windows Authentication" is **enabled** and "Anonymous Authentication" is **disabled** as described in the section below.



Name	Status	Response Type
Anonymous Authentication	Disabled	
ASP.NET Impersonation	Disabled	
Basic Authentication	Disabled	HTTP 401 Challenge
Forms Authentication	Disabled	HTTP 302 Login/Redirect
Windows Authentication	Enabled	HTTP 401 Challenge

2.) Open Admin App's web.config file (i.e., "C:\ed-fi\Admin App v1.7\EdFi.Ods.AdminApp.Web\web.config). Find the **<system.web>** section and insert an authorization block as shown below, customizing for your scenario:

```
  <system.web>
    <authorization>
      <allow roles="DOMAINNAME\Administrators, DOMAINNAME\AdminApp Users" />
      <allow users="DOMAINNAME\Administrator, DOMAINNAME\user1" />
      <deny users="*" />
    </authorization>
  ...
  </system.web>
```

This authorization block will restrict usage to the Active Directory groups (as within the <allow roles=""> block) and/or users (as within the <allow users=""> block). A restart may be needed within Internet Information Services if changes don't immediately take effect.

Please see "How to implement Windows authentication and authorization in ASP.NET" from Microsoft Support on more details on limiting access via Active Directory and Web.config.

Updating NTLM for On-Premises Admin App Deployments

ⓘ The NTLM issue noted below only applies to on-premises deployments. Admin App instances deployed via the Ed-Fi ODS / API Cloud Deployment for Azure offering on Ed-Fi Exchange are not impacted by the vulnerability.

In on-premises deployments, the Admin App uses Active Directory via Internet Information Service (IIS) as the primary authentication method for the application. IIS can use various authentication providers such as NTLM, Kerberos, and others to communicate with Active Directory to authorize users. NTLM in general — and NTLMv1 in particular — have known vulnerabilities that may impact on-premises users of Admin App v1.0 as part of standard IIS configuration. This is not an issue with the Admin App per se, but is important to be aware of.

Some organizations have disabled NTLM throughout their enterprise. If your organization has done that, then you're not impacted by this vulnerability.

However, many organizations require NTLM for older devices and applications. If your organization has not disabled NTLM, the Ed-Fi Alliance strongly recommends that IIS be configured to use the Negotiate protocol for on-premises deployments.

Technical Details

There are two types of authentication providers widely available and used with IIS:

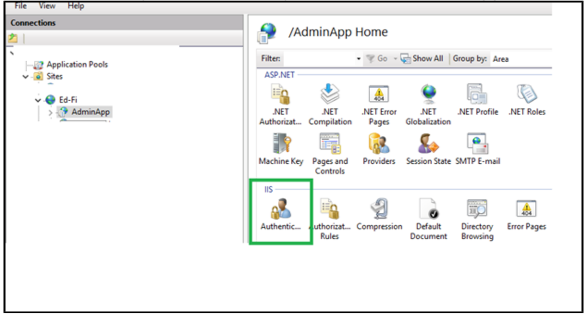
- **The Negotiate Provider.** This provider will attempt to use Kerberos for authentication if it is available. Kerberos provides strong authentication for client-server applications such as the Admin App.
- **The NTLM Provider.** This provider will use NTLM only. NTLM is an older protocol with disadvantages that include relatively weak cryptography, no mutual authentication, and no multi-factor authentication. Further, NTLMv1 is open to a relay attack vector, which is an easy-to-exploit vulnerability.

Other providers exist, but these are currently installed and available in most typical installations. **Because of the disadvantages with NTLM, it is strongly recommended that your organization configures IIS to use the Negotiate protocol**.

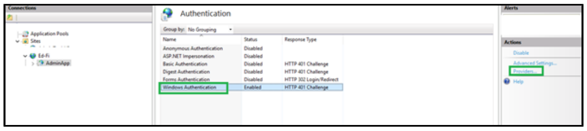
IIS Configuration

The following are steps to configure IIS to use the Negotiate protocol:

1. Open IIS Manager. In IIS Manager, select the AdminApp website, then click Authentication:



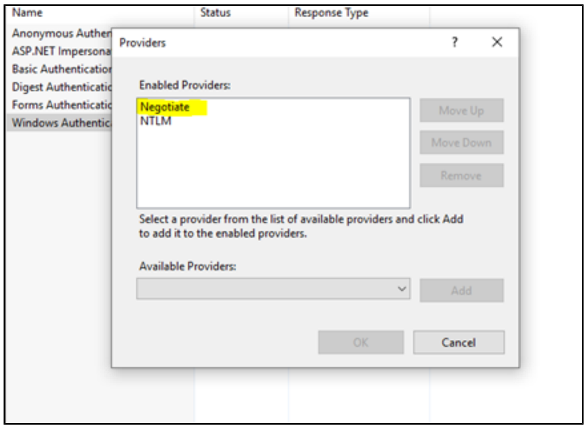
2. Select Windows authentication. Click the Providers... link under Actions:



Name	Status	Response Type
Anonymous Authentication	Disabled	
ASP.NET Impersonation	Disabled	
Basic Authentication	Disabled	HTTP 401 Challenge
Digest Authentication	Disabled	HTTP 401 Challenge
Forms Authentication	Disabled	HTTP 302 Login/Redirect
Windows Authentication	Enabled	HTTP 401 Challenge

⚠ Ensure that when you use Windows Authentication, **Anonymous Authentication is not enabled**.

3. In order to set up Kerberos, make sure the **Negotiate** provider is at the top of the providers list. The Negotiate provider supports Kerberos protocol. It uses NTLM as backup when Kerberos fails, which minimizes the attack surface while ensuring backward compatibility with older devices.



Negotiate
NTLM

Select a provider from the list of available providers and click Add to add it to the enabled providers.

Further Reading on IIS and NTLM

- Microsoft Developer: Setting Up Kerberos Authentication for a Website in IIS
- Microsoft TechNet: Two easy ways to pick Kerberos from NTLM in an HTTP capture
- Help Net Security: Critical Microsoft NTLM vulnerabilities allow remote code execution on any Windows machine
- Stack Overflow: NTLM vs Kerberos (discussion)