Entrust

ISAPI Filter 13.0 Technical Integration Guide

Document issue: 1.0

Date of Issue: April 2025

TOC

About

About this guide

This guide describes how to install, configure and use the Entrust ISAPI Filter solution. It describes post-installation configuration tasks that you may want to perform before you start using the solution, as well as advanced configuration methods to tailor the solution to your needs.

It also describes ways of customizing the solution, such as adding your own logo and colors to the login pages, changing the font and layout of the Web forms, and modifying text and error messages that are displayed to the user.

Topics in this chapter:

Revision information

Table 1: Revisions in this document

Document issue and date	Section	Description
1.0 March 2025		First issue of this document

Documentation conventions

The following documentation conventions are used in Entrust guides:

Table 2: Typographic conventions

Convention	Purpose	Example
Bold text (other than headings)	Indicates graphical user interface elements and wizards.	Click **Next**.
Italicized text	Used for book or document titles.	*Entrust Identity Enterprise Administration Guide*
Underlined blue text	Used for Web links.	For more information, visit our Web site at www.entrust.com.
`Courier` type	Indicates installation paths, file names, Windows registry keys, commands, and text you must enter.	Use the `entrust-configuration.xml` file to change certain options for Verification Server.
Angle brackets < >	Indicates variables (text you must replace with your organization's correct values).	By default, the `entrust.ini` file is located in `<install_path>/conf/security/entrust.ini`.
Square brackets `[courier type]`	Indicates optional parameters.	`dsa passwd [-ldap]`

Note and Attention text

Throughout this guide, there are paragraphs set off by ruled lines above and below the text. These paragraphs provide key information with two levels of importance, as shown below.

Note:
Information to help you maximize the benefits of your Entrust product.

Attention:
Issues that, if ignored, may seriously affect performance, security, or the operation of your Entrust product.

Related documentation

This section describes related reading material that may be used in conjunction with this guide.

- Entrust ISAPI Filter Release Notes
- Entrust Identity Enterprise Release Notes
- Entrust Identity Enterprise Deployment Guide
- Entrust Identity Enterprise Database Configuration Guide
- Entrust Identity Enterprise Directory Configuration Guide
- Entrust Identity Enterprise Installation Guide
- Entrust Identity Enterprise Administration Guide
- Entrust Identity Enterprise Programming Guide for the Java Platform
- Entrust Identity Enterprise Programming Guide for the .NET Framework
- Entrust Pocket Token Administration Guide
- Entrust Mini Token Administration Guide
- Entrust Identity as a Service Online Help

Obtaining additional documentation

Entrust product documentation, white papers, technical notes, and a comprehensive Knowledge Base are available through Entrust TrustedCare Online. If you are registered for our support programs, you can use our Web-based Entrust TrustedCare Online support services at:

https://trustedcare.entrust.com/

Documentation feedback

You can rate and provide feedback about product documentation by completing the online feedback form. Any information that you provide goes directly to the documentation team and is used to improve and correct the information in our guides. You can access this form by:

- clicking the *Report any errors or omissions* link located in the footer of PDF documents (see bottom of this page).
- following this URL: http://go.entrust.com/documentation-feedback.

Obtaining technical assistance

Entrust recognizes the importance of providing quick and easy access to our support resources. The following subsections provide details about the technical support and professional services available to you.

Technical support

Entrust offers a variety of technical support programs to help you keep Entrust products up and running. To learn more about the full range of technical support services, visit our Web site at:

https://www.entrust.com

If you are registered for our support programs, you can use our Web-based support services.

Entrust TrustedCare Online offers technical resources including product documentation, white papers and technical notes, and a comprehensive Knowledge Base at:

https://trustedcare.entrust.com/

If you contact Customer Support, please provide as much of the following information as possible:

- your contact information
- product name, version, and operating system information
- your deployment scenario
- description of the problem
- copy of log files containing error messages
- description of conditions under which the error occurred
- description of troubleshooting activities you have already performed

E-mail address

The e-mail address for Customer Support is:

support@entrust.com

Professional Services

The Entrust team assists organizations around the world to deploy and maintain secure transactions and communications with their partners, customers, suppliers and employees. Entrust offers a full range of professional services to deploy our solutions successfully for wired and wireless networks, including planning and

design, installation, system integration, deployment support, and custom software development.

Whether you choose to operate your Entrust solution in-house or subscribe to hosted services, Professional Services will design and implement the right solution for your organization's needs. For more information about Professional Services please visit our Web site at:

https://www.entrust.com/services

Training

Through a variety of hands-on courses, Entrust delivers effective training for deploying, operating, administering, extending, customizing and supporting any variety of Entrust digital identity and information security solutions. Delivered by training professionals, Entrust's professional training services help to equip you with the knowledge you need to speed the deployment of your security platforms and solutions. Please visit our training website at:

https://entrust.com/resource-center/training

1

About the ISAPI Filter solution

The ISAPI Filter solution uses Entrust Identity Enterprise Server and Identity as a Service to provide strong second-factor authentication to Microsoft Outlook Web Access (OWA), Remote Desktop Web Access (RD Web Access), Integrated Windows Authentication (IWA), SharePoint, and generic forms-based authentication types. The solution consists of the filter component and the authentication application component.

You can use the ISAPI Filter with the Entrust Identity Enterprise Server and Identity as a Service authentication methods to allow only valid users access to a Web application.

Topics in this chapter:

Terminology

Some terms used frequently in this guide are defined here.

Table 3: Terminology

Term	Description
DMZ	This is the demilitarized zone or demarcation zone of a network.
Entrust Identity Enterprise Authentication Web application	This is an ASP.NET application that is distributed as part of the Entrust ISAPI Filter solution. It communicates with Entrust Identity Enterprise Server and Identity as a Service to provide second-factor authentication.
IIS	Internet Information Services. This is the Microsoft Web server available with Microsoft Windows Server.
ISAPI	Internet Server Application Programming Interface. This is the API for Microsoft Windows IIS.
ISAPI Filter	This is a DLL distributed as part of the Entrust ISAPI Filter solution. It works with the Entrust Identity Enterprise Authentication Web application to provide the functionality of this solution. In this guide, “Filter” refers to the solution as a whole, and “filter” refers to the filter component of the solution.
IWA	Integrated Windows Authentication
OWA	Outlook Web Access, renamed to Outlook Web App in Exchange 2016. This is the Microsoft Exchange Server Web mail service.
RBA	Risk-based authentication.
PVN	Personal verification number.

First-factor authentication methods

The Entrust ISAPI Filter solution supports the following first-factor authentication methods.

Microsoft Integrated Windows Authentication (IWA)

IWA is included with Microsoft IIS. It uses the current Windows client user information to authenticate the user. If the authentication fails, it prompts the user for the correct user name and password.

Anonymous access must be disabled on IIS for the ISAPI Filter to work with Integrated Windows Authentication.

Microsoft Outlook Web Access or Outlook Web App (OWA)

OWA is the Microsoft Exchange Server Web mail service. It permits users to access their email and other Outlook content from a browser, when they do not have access to the Outlook desktop application.

Microsoft Remote Desktop Web Access

Remote Desktop Web Access (RD Web Access), formerly Terminal Services Web Access, enables users to access RemoteApp and Desktop Connection through a Web browser or through the **Start** menu on a computer running Windows 7.

Forms-based authentication

Forms-based authentication uses HTML forms to establish the identity of a user trying to access pages on a Web site. When the user's browser sends a request for a restricted page, the Web server looks for a cookie included with the request. If this cookie establishes the identity of the user, the Web server allows the user to access the page. If the request does not include a cookie, the Web server presents an HTML form on which the user must fill in the requested login information. If the user correctly fills in the form, the Web server returns a cookie to the client, and allows the user to access the page. The user can send future requests, with the cookie, and does not need to fill in a login form repeatedly.

Entrust Identity Enterprise and Identity as a Service password authentication

Users can use their Entrust Identity Enterprise and Identity as a Service passwords to authenticate to the ISAPI Filter. For information about setting the ISAPI Filter solution to use Entrust Identity Enterprise and Identity as a Service password for first-factor authentication, see “Configuring first-factor authentication” on page 159.

Second-factor authentication methods

Entrust Identity Enterprise and Identity as a Service normally present the user name and the password first, and the second-factor challenge second. You can configure the ISAPI Filter to reverse this authentication order in some cases. See the installation Step 13 on page 81.

Supported second-factor authentication method	Notes
Grid	Presents the user name and the password on one page, and the grid challenge on the next page. Grid authentication can be used with a personal verification number (PVN) if your system is set up to require it. See “Configuring grid authentication” on page 169.
Token	Supports use of dynamic passwords generated by hardware tokens, both response-only and challenge-response tokens. Typically, tokens present the user name and password on one page, and the token challenge on the next page. You can configure the ISAPI Filter to present the authentication in one step, or skip the password prompt for generic form-based authentication. Token authentication can be used with a personal verification number (PVN) if your system is set up to require it. See “Configuring token authentication” on page 170.
Mobile smart credential (IA)	Supports use of a mobile smart credential, which is contained within the Entrust Identity Mobile Smart Credential app. This app is available for Android, BlackBerry, and iOS devices. See “Configuring mobile smart credential authentication” on page 172.
Mobile soft token (TVS) authentication	For mobile soft token authentication, an out-of-band authentication challenge is sent to the user’s mobile device. The user selects **Confirm** to access the protected resource. If the authentication request was not initiated by the user or appears fraudulent, the user can select **Cancel** to deny access to the resource, or select **Concern**, in which case the authentication request is canceled.

Supported second-factor authentication method	Notes
Out-of-band one-time password (OTP)	Supports use of a one-time password that is sent to the user when needed or when a predefined threshold is reached. Entrust Identity Enterprise allows users to have multiple OTPs. Since OTPs can be used only once, the user's supply of OTPs is reduced with each authentication. When the user's supply of OTPs falls below a threshold, Entrust Identity Enterprise automatically generates and sends a new supply of OTPs. The operation and refresh threshold policies are defined in Entrust Identity Enterprise. OTP authentication can be used with a personal verification number (PVN), if your system is set up to require it. See "Configuring out-of-band OTP authentication" on page 177.
Temporary PIN	Supported when used to replace grid or token authentication. When a user loses their grid or token, they can still authenticate using a temporary PIN. Temporary PIN authentication can be used with a personal verification number (PVN), if your system is set up to require it.
Knowledge-based (question and answer)	Supports use of user-configured questions and answers for second-factor authentication. See "Configuring knowledge-based authentication" on page 180.
Risk-based authentication (RBA)	Supports checking the IP address and additional client information (for example persistent browser cookies) of the user logging in. See "Configuring risk-based authentication" on page 185.
Step-up	A user who has previously authenticated using a standard authentication mechanism tries to access a protected resource that requires a higher authentication level, and is redirected to Entrust Identity Enterprise or Identity as a Service for a stricter level of second-factor authentication. See "Configuring step-up authentication" on page 198.

Supported second-factor authentication method	Notes
Passkey/FIDO2	A Passkey/FIDO2 token can be used for second-factor authentication for user ID log in or Passkey log in. When the user attempts to authenticate, a challenge is sent to the Passkey/FIDO2 token. The Passkey/FIDO2 token signs the challenge with a private key associated with the application to allow the user to log in. See "Configuring Passkey/FIDO2 authentication" on page 182
Policy-based	When set to policy-based authentication, the second-factor authentication method used is governed by the Entrust Identity Enterprise Authentication Types policy. Two users might see different authenticators when they log in; for example, one might see knowledge-based prompts, and the other may see a token challenge page. Entrust Identity Enterprise and Identity as a Service determine which second-factor authentication type to present to a user, based on which challenge types are currently supported and valid for that user. See "Configuring policy-based authentication" on page 183.

ISAPI solution components

The Entrust Identity Enterprise ISAPI solution has the following main components.

ISAPI Filter

This is a DLL distributed as part of the Entrust ISAPI Filter solution. It works with the Entrust Identity Enterprise Authentication Web application to provide the functionality of this solution. It uses the Microsoft Internet Server Application Programming Interface (ISAPI).

The ISAPI Filter is designed for installation on an IIS server. IAll Web traffic related to the protected application must pass through the server on which the ISAPI Filter is installed. The ISAPI Filter acts as a gatekeeper that inspects the URLs being accessed. If the URL being accessed is protected and the session is not authenticated, then the ISAPI Filter redirects the user to the Entrust Identity Enterprise Authentication Web application for authentication.

Entrust Identity Enterprise Authentication Web Application

This application distributed as part of the Entrust ISAPI Filter solution. It communicates with Entrust Identity Enterprise Server to present and validate second-factor authentication to the user. This is an ASP.NET application that displays the HTML forms to users, so they can enter authentication credentials.

The Entrust Identity Enterprise Authentication Web application (authentication application) is installed on an IIS server only. The authentication application packaged with the solution is called `IdentityGuardAuth`.

The authentication validation service is a sub-component of the authentication application. It is a Web service invoked by the ISAPI Filter before allowing access to a protected URL, to validate that the user has authenticated at the required authentication level.

Entrust Identity Enterprise Server

When a user tries to access a protected resource, the ISAPI Filter communicates with the Entrust Identity Enterprise Server. This server establishes the identity of the user using various methods; question-and-answer, grid cards, or dynamic password tokens, for example. For a list, see “Second-factor authentication methods” on page 23.

Security concepts

The following are some security concepts that you need to know before you begin the installation.

Protected host

The protected host is a domain in your network that you want to protect with the ISAPI Filter. You specify it during installation of the ISAPI solution. The information is stored in the filter configuration file `IdentityGuardFilterConfiguration.xml` and can be modified later. For more information, see “Configuring a protected host” on page 223.

You can configure more than one protected host. For more information, see “Handling multiple hosts on one server” on page 232.

The `ProtectedHost` element in the `IdentityGuardFilterConfiguration.xml` file can contain several protected URLs. Each of these represents a protected resource on your network. You specify these during installation. You can modify these URLs after installation, if needed.

If you select Outlook Web Access as the first-factor authentication type during installation, the installer does not prompt you to enter protected URLs. It automatically configures the protected URLs, based on your OWA setup. You can modify these URLs after installation, if needed.

You can use wild card characters to specify the protected URLs. They can be a single Web page, or several. For more information see “Adding or removing protected and unprotected URLs” on page 226.

Authentication levels

Each protected URL can have a level assigned to it. Each level is assigned a Entrust Identity Enterprise second-factor authentication method, such as grid or token. The ISAPI Filter solution treats Level 1 as the standard level, Level 2 as stricter, and so on.

See “Defining an authentication level” on page 163 and “Configuring a protected host” on page 223 for more information about using authentication levels, and for examples.

Step-up authentication

In this situation, a user has been authenticated and has access to a protected resource. The user requests access to another resource that requires a higher level of authentication. If step-up authentication has been configured, the user is presented with an additional challenge before being permitted to access the higher-level resource.

For example, a user accesses a personal banking Web site, which is at Level 1. The user is presented with an Entrust Identity Enterprise grid challenge. Upon successful authentication, the user can access all the resources at the personal banking level, which is Level 1.

The user now attempts to pay a bill from the checking account. Since the bill-payment page is at Level 2, the user is presented with an Entrust Identity Enterprise token challenge. Upon successful authentication, the user steps up to Level 2 and can access all the resources at both the personal banking level and the bill-payment level.

For more information see “Configuring step-up authentication” on page 198.

Policy-based authentication

In policy-based authentication, the second-factor authentication method used is governed by the Entrust Identity Enterprise **Authentication Types** policy. For example, two users might see different authenticators when they log in: knowledge-based authentication for one user, and token authentication for the other user. Entrust Identity Enterprise determines which second-factor authentication type to present to a user, based on which challenge types are supported and valid for that user.

For more information see “Configuring policy-based authentication” on page 183.

First-factor ID

For each protected host (ProtectedHost) and protected URL (ProtectedURLs) under a host, you can specify a different first-factor authentication type.

The filter component has a first-factor type assigned to the URL and the authentication application defines that type. The first-factor type is entered in the configuration files for the ISAPI Filter and the authentication application. The first-factor ID is an attribute that links these two files, permitting them to map the user in the filter to the definition in the authentication application.

Each protected URL must have a first-factor ID assigned to it.

For more information see “Adding or removing protected and unprotected URLs” on page 226.

Failover

The ISAPI Filter solution gives you the ability to use failover in two ways.

Failover for Entrust Identity Enterprise Servers

This solution allows you to set up a failover architecture by increasing the number of Entrust Identity Enterprise Servers. Each server contains a copy of the Entrust

Identity Enterprise authentication Web service that can take over, should your preferred server fail, or otherwise become unavailable. For more information see “Configuring failover for Entrust Identity Enterprise Servers” on page 250.

Failover for multiple authentication Web application servers

Multiple Entrust ISAPI Filter and authentication Web applications can also be installed. For more information, see “Configuring failover in an IIS environment” on page 254 and “Configuring failover in an IIS environment” on page 254.

Supported installation configurations

The solution supports Microsoft Internet Information Services (IIS) configuration. You can use multiple Entrust Identity Enterprise Servers in the configuration.

IIS configuration

Attention:
You must install the ISAPI Filter and the authentication application on the same IIS server. The integration does not operate if the ISAPI Filter and the authentication application are installed on different IIS servers.

In this configuration the filter and authentication application components are both installed on an IIS server. The authentication application provides first-factor authentication, using one of the supported first-factor authentication methods:

- Outlook Web Access (OWA)
- Integrated Windows Authentication
- Entrust Identity Enterprise Password Authentication
- Generic Forms Based Authentication
- Generic Forms Based Passwordless Authentication
- Remote Desktop Web Access

The authentication application allows Entrust Identity Enterprise to provide second-factor authentication.

Microsoft Internet Information Services (IIS) server is the Microsoft Web server that is available with various Microsoft Windows operating systems. If you are installing this solution with OWA, IIS is referred to as the Client Access Server.

Typical solution architectures

The following sections show typical system architectures used with the Entrust ISAPI Filter solution. The components in blue are a part of this solution.

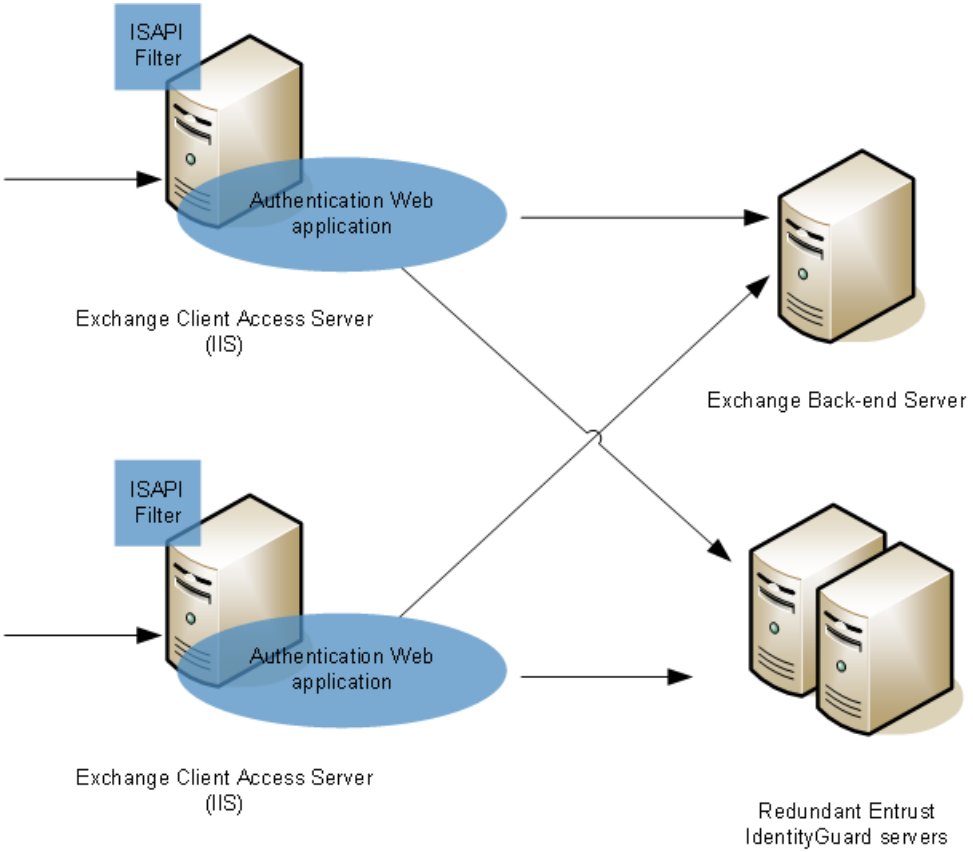
Typically the IIS servers running the solution are located inside your firewall, or in a DMZ to isolate network traffic. The Entrust Identity Enterprise Server is typically deployed inside your firewall. See the *Entrust Identity Enterprise Deployment Guide* for more information on deploying the server.

- "OWA architecture" on page 32
- "ARR architecture" on page 33
- "ARR architecture" on page 33
- "ARR architecture" on page 33
- "UAG/ADFS architecture" on page 34

OWA architecture

In this architecture, the Exchange front-end server hosts the protected Web pages and also hosts the Entrust Identity Enterprise authentication application.

Figure 1: Entrust ISAPI Filter for OWA on an IIS server



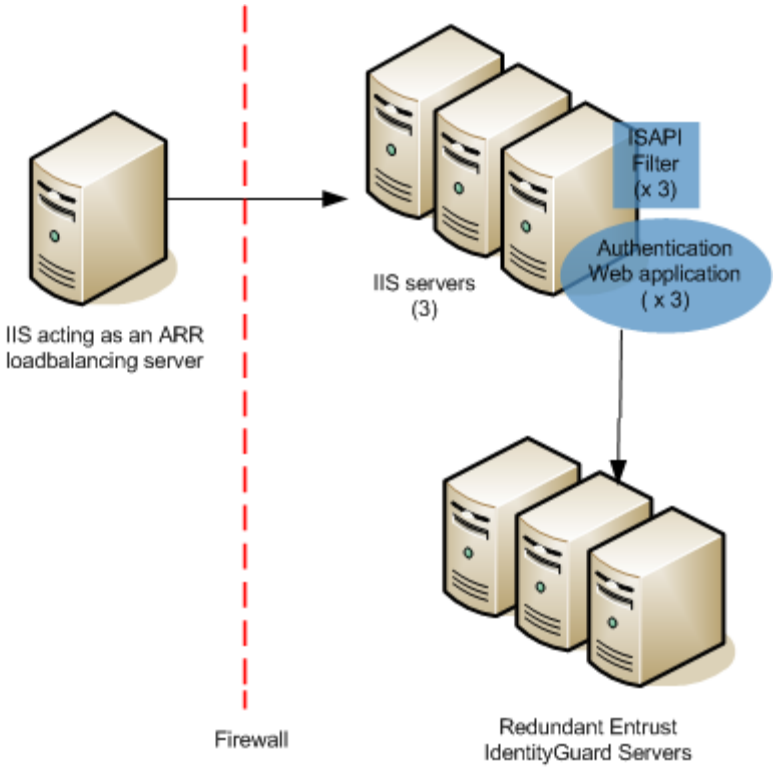
ARR architecture

With this architecture:

- there is a single IIS server acting as a router to an IIS server farm behind it. To enable IIS as a router, you must install the Application Request Router (ARR) extension, available online.
- the ARR server does not have any Entrust software installed
- each IIS server in the server farm has the Entrust ISAPI Filter and corresponding authentication application installed.

For more on ARR, see http://www.iis.net/downloads/microsoft/application-request-routing.

Figure 2: Entrust ISAPI Filter in an ARR configuration



Note:
The ISAPI Filter in the ARR architecture does not support IP and certificate risk-based authentication (RBA).

UAG/ADFS architecture

ISAPI Filter cannot be used with UAG directly because UAG does not support filters.With this release. However, UAG can be configured to use Active Directory Federation Services (ADFS) as the authentication server and ADFS can, in turn, be protected by the ISAPI filter.

The user experience is as follows:

1. A user navigates to the UAG portal: https://<hostname>/<portal>/.
2. The user is redirected to a IWA dialog box to provide first-factor authentication.
3. After entering valid credentials, the user is redirected to the second-factor authentication page.
4. After successfully completing a second-factor authentication challenge, the user can access the UAG portal and see the published applications.

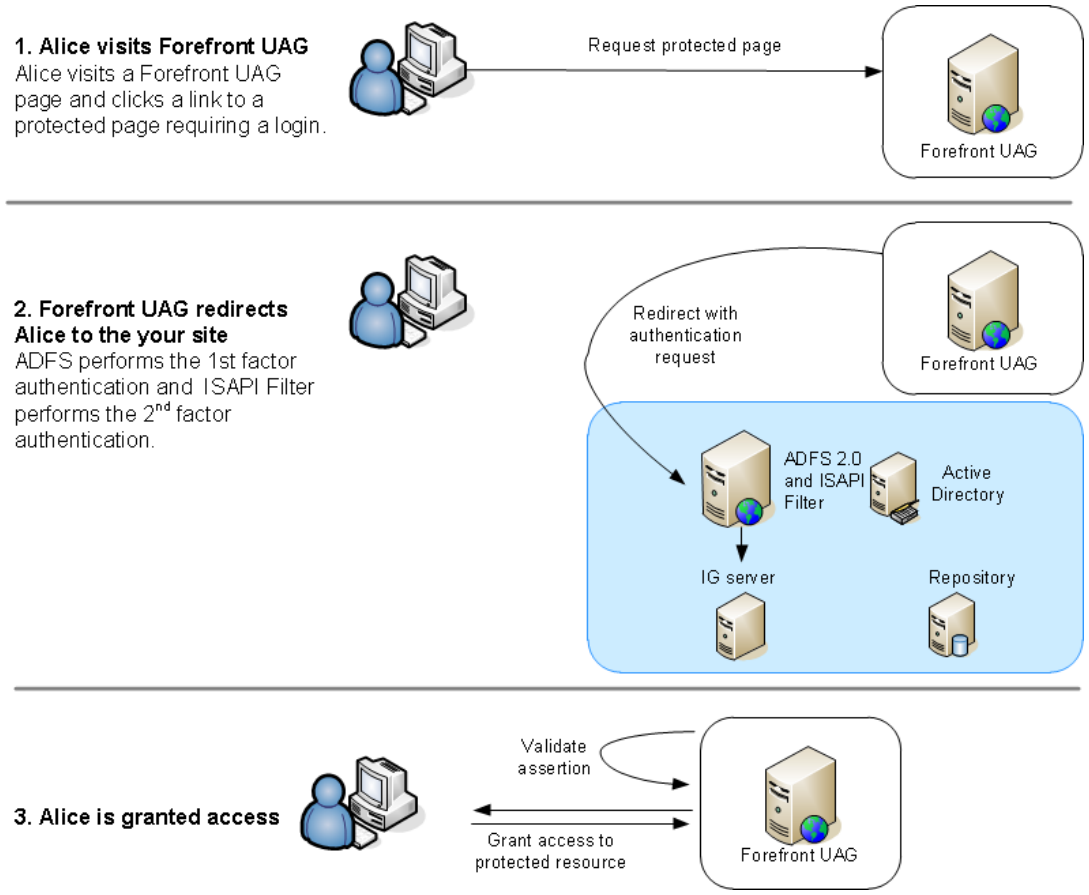
With this architecture:

- Identity Guard ISAPI filter is installed on the same IIS server as ADFS
- the "/adfs/ls" URL is protected in the filter configuration file. To do this, ensure the IdentityGuardFilterConfiguration.xml has the following entry:

```
<ProtectedURLs authlevel="1" firstfactorid="iwa">
<URL authlevel="1">/adfs/ls/</URL>
</ProtectedURLs>
```

- ISAPI Filter is configured to use IWA for first-factor authentication.

Figure 3: Entrust ISAPI Filter in a UAG/ADFS configuration



2

Preparing for installation

Complete these procedures, as needed before installing the ISAPI Filter solution.

Topics in this chapter:

Prerequisites

The Entrust ISAPI Filter Identity as a Service Authentication API uses TLS 1.2.

For Identity as a Service Authentication API, you must change the subkeys listed below to establish a successful connection between the Identity as a Service Authentication API and the Entrust ISAPI Filter.

Subkeys for this new registry key

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\.NETFramework\v2.0.50727]"SchUseStrongCry
pto"=dword:00000001
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\.NETFramework\v4.0.30319]"SchUseStrongCry
pto"=dword:00000001
[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\.NETFramework\v4.0.30319]"Sch
UseStrongCrypto"=dword:00000001
[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\.NETFramework\v2.0.50727]"Sch
UseStrongCrypto"=dword:00000001
```

All .net applications using Framework 4.8 use SSL 3.0 and TLS 1.0 by default for all SSL/TLS Handshakes.

Completing initial tasks

Complete the following tasks before you proceed to the next topic.

1. If you are installing ISAPI Filter for **Entrust Identity Enterprise Server**, ensure that Entrust Identity Enterprise Server is installed and configured. This includes the Entrust Identity Enterprise Authentication Web service. The repository must include a few users for testing purposes.
2. If you are using the IIS Application Request Routing (ARR) extension (see “ARR architecture” on page 33), ensure that you complete the following preliminary tasks:
   a. Configure an ARR loadbalancing server.
   b. Configure an IIS server farm behind the ARR server.
   c. Ensure that the ARR server is aware of each IIS server in the farm.
3. If you are installing ISAPI Filter for **Identity as a Service**, you must do the following:
   a. Add an Authentication API to Identity as a Service to generate an Application ID. You need to add the Application ID during the Entrust ISAPI Filter installation.
   b. Create a resource rule in Identity as a Service for Entrust ISAPI Filter.

See the section, “Integrate Identity as a Service ISAPI Filter” in the *Identity as a Service Administrator Online Help* for more information.

4. Set up the network components required to facilitate host/domain resolution and to route between the required subnets (such as DNS and subnet routing).
5. If you are protecting Outlook Web Access, on IIS configuration, ensure that you can successfully access OWA using the appropriate first-factor authentication.
6. If you are protecting Remote Desktop Web Access, on IIS configuration, ensure that you can successfully access RD Web Access using the appropriate first-factor authentication. If you plan to use single sign on, test to ensure that this feature is operating correctly.
7. To run ISAPI 13.0, you must first install .net Framework 4.0 or higher.
8. If you are running on a 64 bit platform, ensure that 64 bit Visual Studio C++ 2015-2022 redistributable is installed.

Confirming the preinstallation requirements

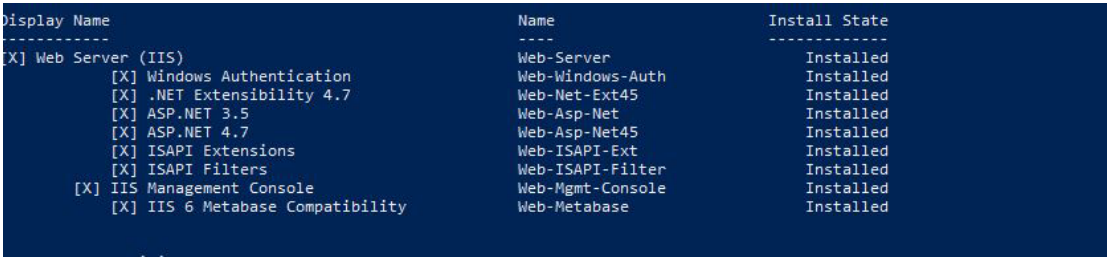
Before installing the ISAPI Filter, you can run a command in PowerShell to confirm that you have all the prerequisites before you start the installation process.

To confirmation the pre-installation prequirements

1 At the PowerShell command prompt, enter

```
Import-module servermanager; Get-WindowsFeature
Web-Server,Web-Mgmt-Console,Web-Asp-Net,Web-Asp-Net45,Web-net-Ext45,Web-ISAPI-Ex
t,Web-ISAPI-Filter,Web-Metabase,Web-Windows-Auth; Get-WmiObject -Class
Win32_Product -Filter "Name LIKE 'Microsoft Visual C++ 2019%'"
```

Your output should appear as follows:



Note:
Note: If an [X] appears in front of the prerequisite, it is already installed. If there is no [X] showing beside the requirement, then you must install it.

If some prerequisites are installed and some are missing, you should see a message indicating that no change is needed for the already installed prerequisite.

Note:
For the following output, these prerequisites are all required. If a prerequisite does not appear in the output list, then you must install it.

```
IdentifyingNumber : {A1C31BA5-5438-3A07-9EEE-A5FB2D0FDE36}
```

```
Name              : Microsoft Visual C++ 2015-2022  x64 Redistributable -
14.34.31938
Vendor            : Microsoft Corporation
Version           : 14.34.31938
Caption           : Microsoft Visual C++ 2015-2022  x64 Redistributable -
14.34.31938
```

2 If any of the prerequisites roles are missing, you can install the required component using the following command:

```
Import-module servermanager;Install-WindowsFeature
Web-Server,Web-Mgmt-Console,Web-Asp-Net,Web-Asp-Net45,Web-net-Ext45,Web-ISAPI-Ex
t,Web-ISAPI-Filter,Web-Metabase,Web-Windows-Auth
```

Your output should appear as follows:

```
Success Restart Needed Exit Code Feature Result
------- -------------- --------- --------------
True    No             Success {ASP.NET 4.5, Application Development, ASP...
```

Note:
The command in Step 2 does not install Microsoft Visual C. If Microsoft Visual C is missing, download it from the Microsoft Website and install it.

Configuring IIS for use with ISAPI Filter

Use the applicable sections below to ensure that IIS is ready to work with Entrust ISAPI Filter.

IIS 10: Adding the required role services on IIS

The following IIS role services are required on IIS 10:

- **Application Server** role
  - **.NET Framework 4.8**
  - **Web Server (IIS) Support**

    Including this feature also pulls in **ISAPI Filters**, **ISAPI Extensions**, **.NET Extensibility 4.5,** and **Windows Authentication** among others.

    Windows authentication is needed only if you are using integrated Windows authentication (IWA) for first-factor authentication.
  - **Management Tools**
    - **IIS Management Scripts and Tools**

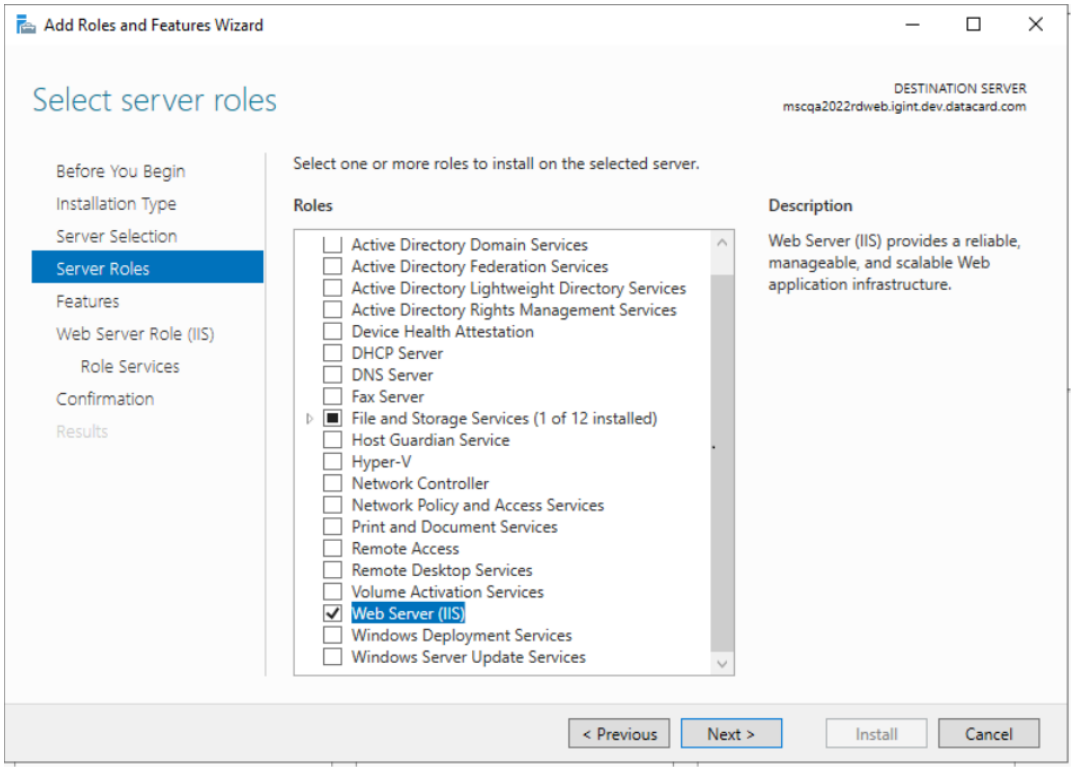
Follow the procedure below to add these roles.

To add required role services on IIS 10

1. On Windows Server 2016, 2019, or 2022, open the Server Manager.
2. In the left pane, double-click **IIS**.
3. At the top-right, from the **TASKS** drop-down list, select **Add Roles and Features**.

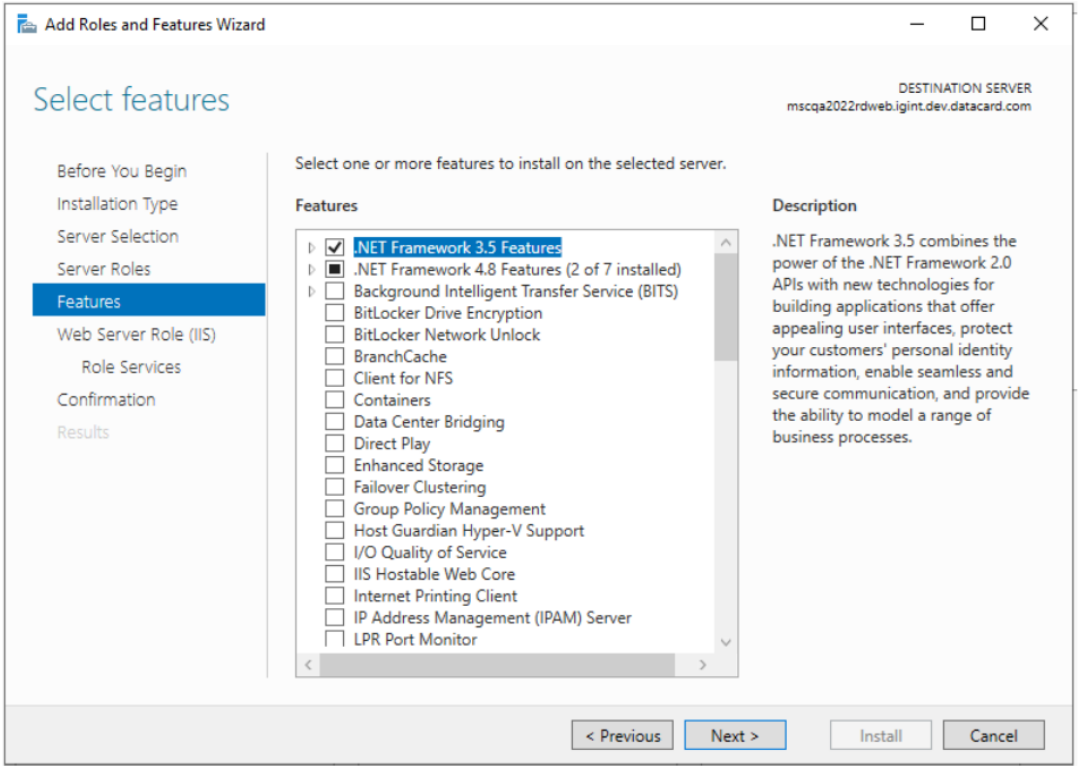
   The **Add Roles and Features Wizard** appears.
4. Click **Next**.
5. Click **Role-based or feature-based installation**.
6. Click **Next**.
7. Select your IIS server. Click **Next**.

A list of roles appears.



8 Select **Web Server (IIS)**. Click **Next**.

A list of features appears.

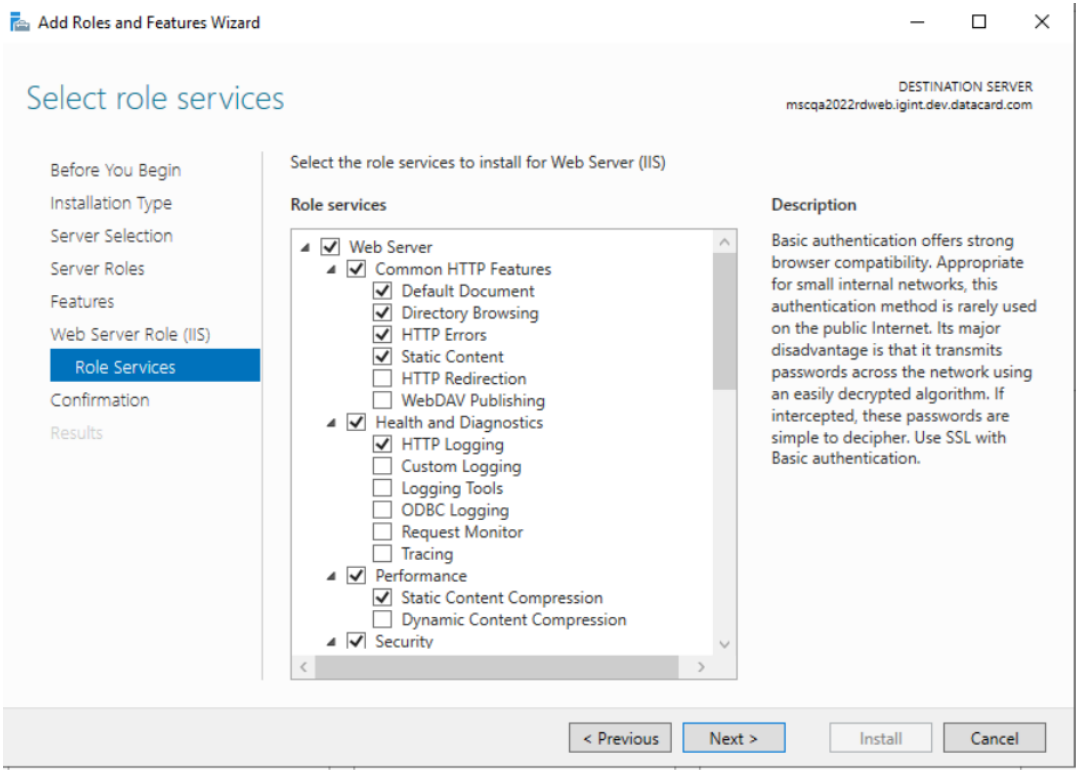


9 Ensure **.NET Framework 4.x Features** are installed based on your Operating System. Enable the check box if they are not.

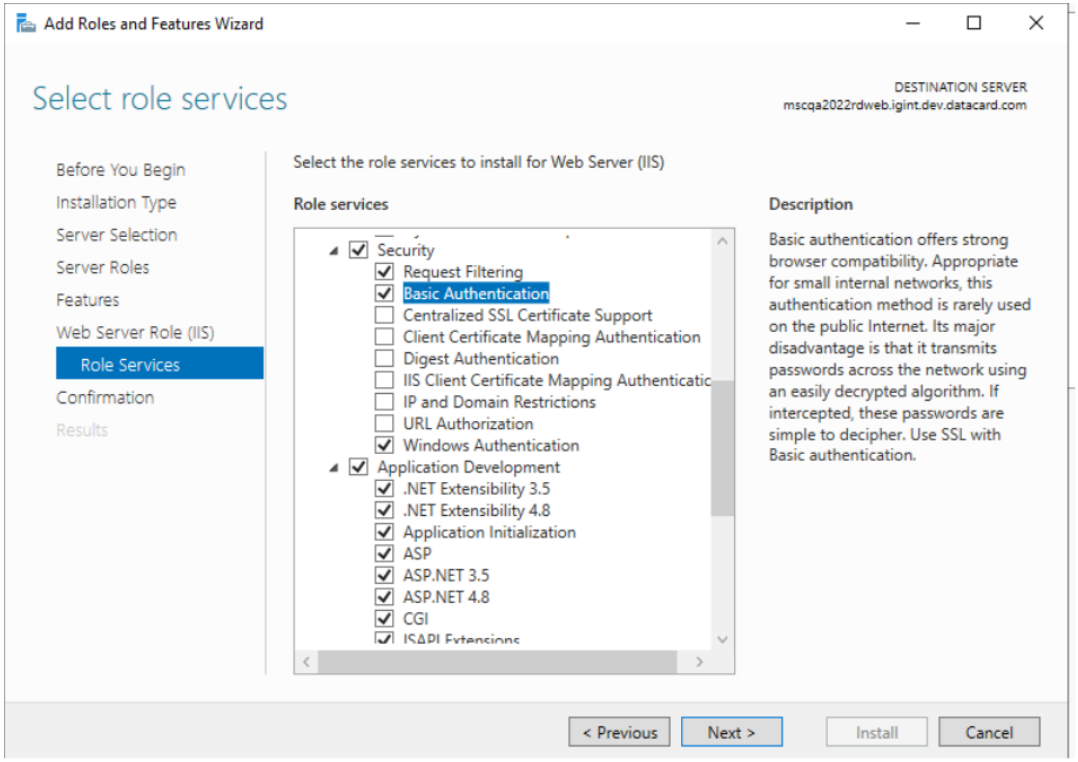
10 Click **Next**.

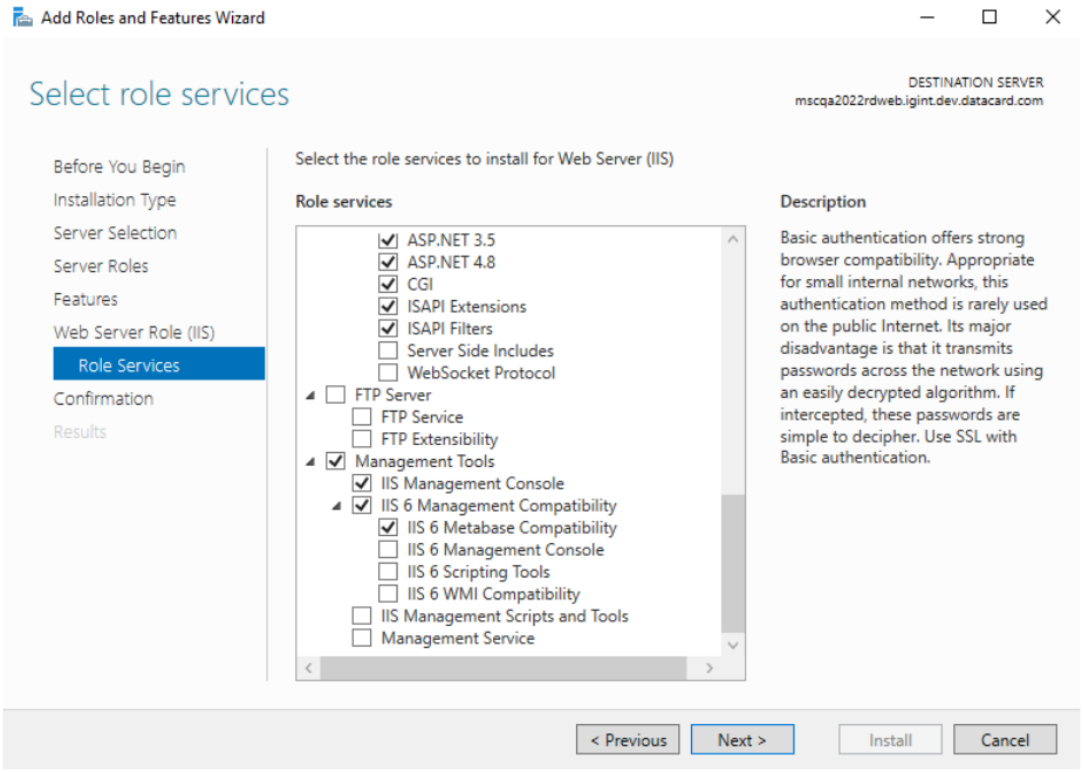
11 Click **Next** on the message that discusses the Application Server role.

The role services appear.



12 Select the roles services as shown in the following screenshots.


Add Roles and Features Wizard
Select role services
DESTINATION SERVER
mscqa2022rdweb.igint.dev.datacard.com
Before You Begin
Installation Type
Server Selection
Server Roles
Features
Web Server Role (IIS)
Role Services
Confirmation
Results
Select the role services to install for Web Server (IIS)
Role services
Security
Request Filtering
Basic Authentication
Centralized SSL Certificate Support
Client Certificate Mapping Authentication
Digest Authentication
IIS Client Certificate Mapping Authenticatic
IP and Domain Restrictions
URL Authorization
Windows Authentication
Application Development
.NET Extensibility 3.5
.NET Extensibility 4.8
Application Initialization
ASP
ASP.NET 3.5
ASP.NET 4.8
CGI
Description
Basic authentication offers strong browser compatibility. Appropriate for small internal networks, this authentication method is rarely used on the public Internet. Its major disadvantage is that it transmits passwords across the network using an easily decrypted algorithm. If intercepted, these passwords are simple to decipher. Use SSL with Basic authentication.
< Previous
Next >
Install
Cancel



13 Click **Next**.

14 Click **Install**.

15 You have now enabled the required roles and services on IIS 10.

Overview of configuring SSL for this solution

For security reasons, SSL is typically used to encrypt traffic that flows between elements of your Web solution and the solution components. SSL uses certificates to encrypt communication, and to verify that applications can trust that they are communicating to the correct end-point.

This overview describes the following SSL communication scenario:

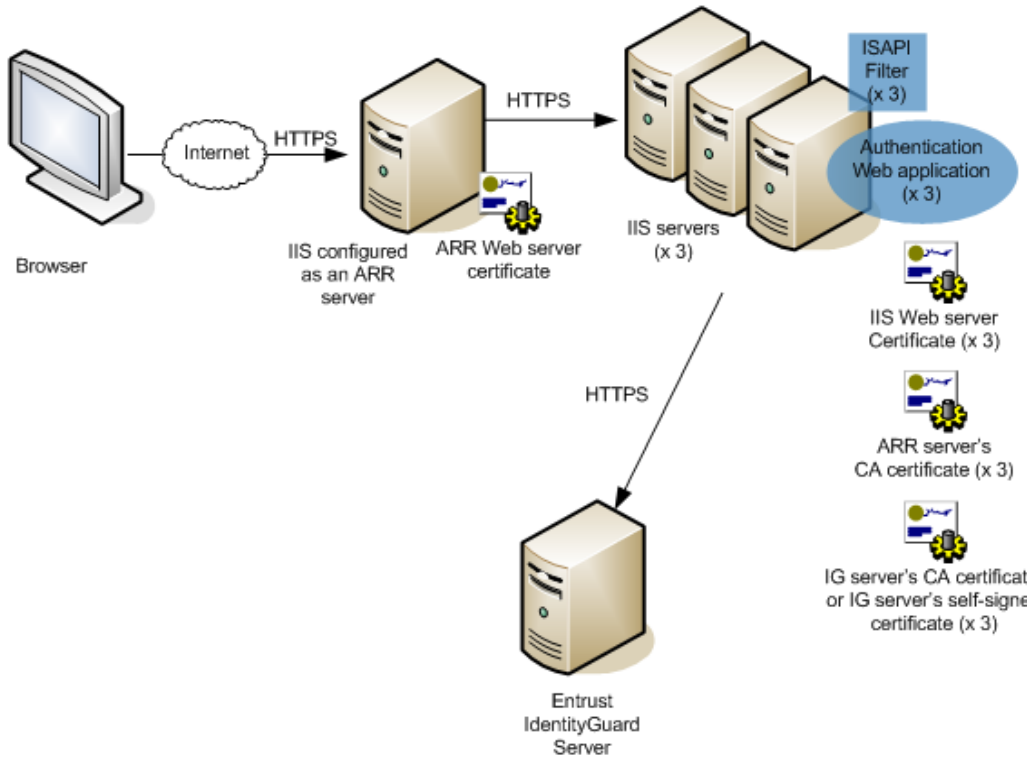
- "SSL in an IIS and ARR environment" on page 48

SSL in an IIS and ARR environment

Figure 4 shows how HTTP traffic flows between the user's browser, ARR, the IIS server farm, and Entrust Identity Enterprise Server.

Below the figure are instructions on configuring this SSL setup.

Figure 4: SSL configuration with the ISAPI Filter solution that includes ARR



To enable SSL in an ISAPI Filter solution with ARR

1. On the ARR server, do the following:
   a. Generate, purchase, and install a Web server certificate. For instructions, see "Enabling SSL on the IIS or ARR server" on page 61.
   b. Disable SSL offloading, as follows:
      – Open the IIS management console.
      – In the left pane, expand the ARR module.
      – Select your server farm.
      – Double-click **Routing Rules**.
      – Disable the **Enable SSL offloading** check box.
      When the check box is disabled, traffic travels from ARR to the IIS servers untouched over HTTPS. (When enabled, SSL traffic is decrypted and passes in clear text over HTTP between the ARR server and IIS servers. This results in errors because the ISAPI Filter expects SSL.)
   c. Export the trusted root CA certificate—as well as all intermediate CA certificates—from the ARR's Web server certificate to a single file. For instructions, see "Creating a certificate chain" on page 53.
   d. Copy the ARR server's CA certificate file to the IIS computers where you will install the ISAPI Filter and authentication application. (During the ISAPI Filter installation, you will be prompted for this file.)
2. On each IIS server in the server farm, do the following:
   a. Generate, purchase, and install a Web server certificate. For instructions, see "Enabling SSL on the IIS or ARR server" on page 61.
   b. Import the Entrust Identity Enterprise Server's self-signed certificate, or the root CA certificate that signed the Entrust Identity Enterprise Server's certificate. For details, see "Configuring SSL between the authentication Web application and Entrust Identity Enterprise" on page 56.
3. If the ARR server and the IIS server farm are on different domains (for example, `owa.com` and `mycorp.com`), then do the following:
   a. Install the ARR server's CA certificate file to each IIS server's Trusted Root Certification Authorities store. For instructions on importing a CA certificate file to the Trusted Root Certification Authorities store, see "Importing certificates into the local computer store" on page 57.
   b. Install each IIS server's CA certificate file to the ARR server's Trusted Root Certification Authorities store.

   If the ARR and IIS servers are on the same domain, you can skip Step 3.

   You have now configured SSL in an ARR environment.

Configuring SSL between ISAPI Filter and the authentication application

It is recommended that you use SSL between the ISAPI Filter and the authentication application when they are installed on separate machines.

Complete the following procedures.

- “Exporting certificates to Base-64 format” on page 50
- “Creating a certificate chain” on page 53
- “Importing certificates into the local computer store” on page 57

See “Overview of configuring SSL for this solution” on page 48 for details of the SSL configuration.

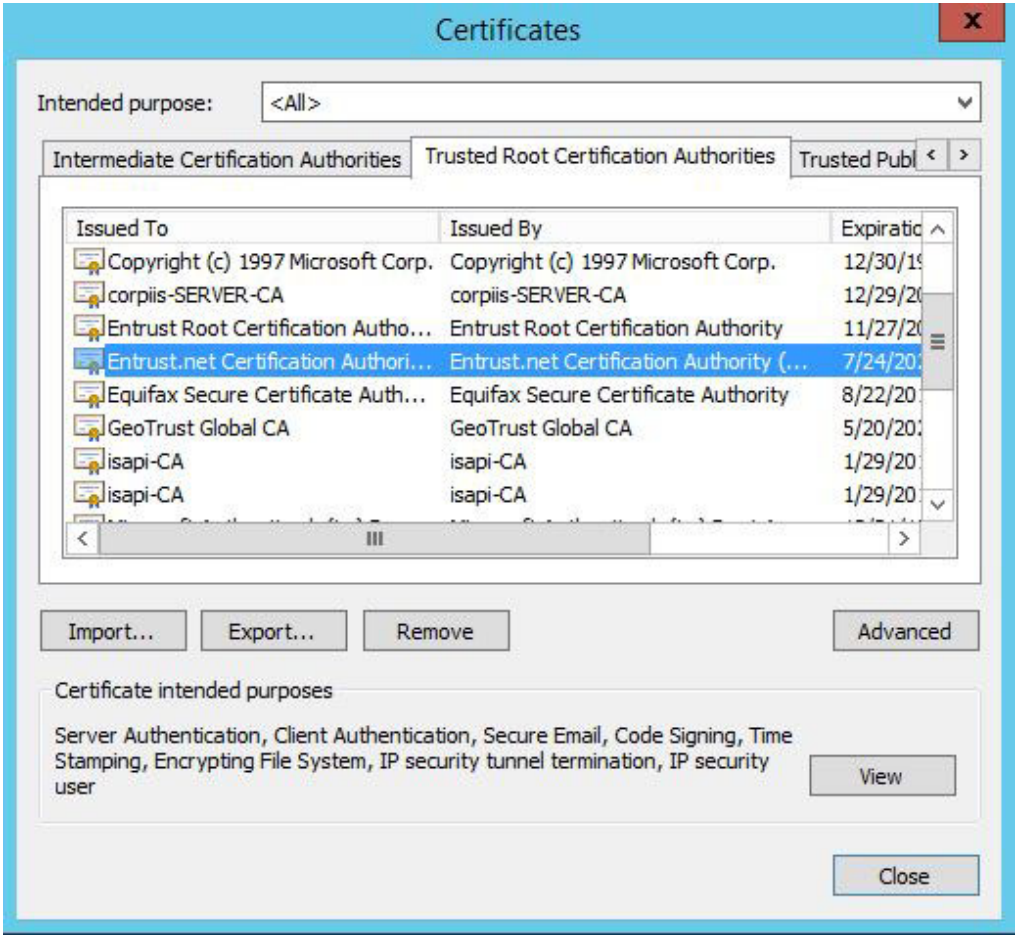
Exporting certificates to Base-64 format

The ISAPI Filter requires a trusted root CA certificate, and any intermediate CA certificates, in a PEM encoded, X.509 file format. You can use Internet Explorer to export your certificates to Base-64 `.cer` file (a PEM format).

To export certificates to Base-64

1. Start Internet Explorer on the server where you plan to install the authentication application.
2. In Internet Explorer, from the **Tools** menu, click **Internet Options**.
3. Click the **Content** tab.
4. Click **Certificates** to view a list of certificates.

5 Click the **Trusted Root Certification Authorities** tab and select the certificate you want to export.



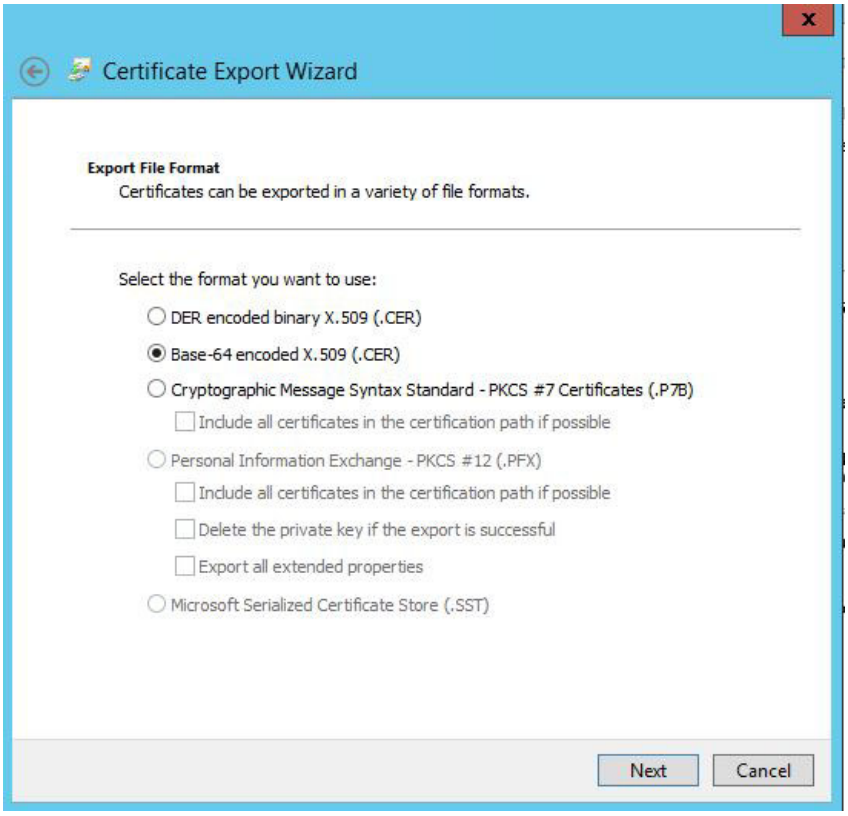
This is the root certificate for the CA that issued the SSL certificate for the IIS Web site where you plan to install the authentication application.

6 Click **Export**.

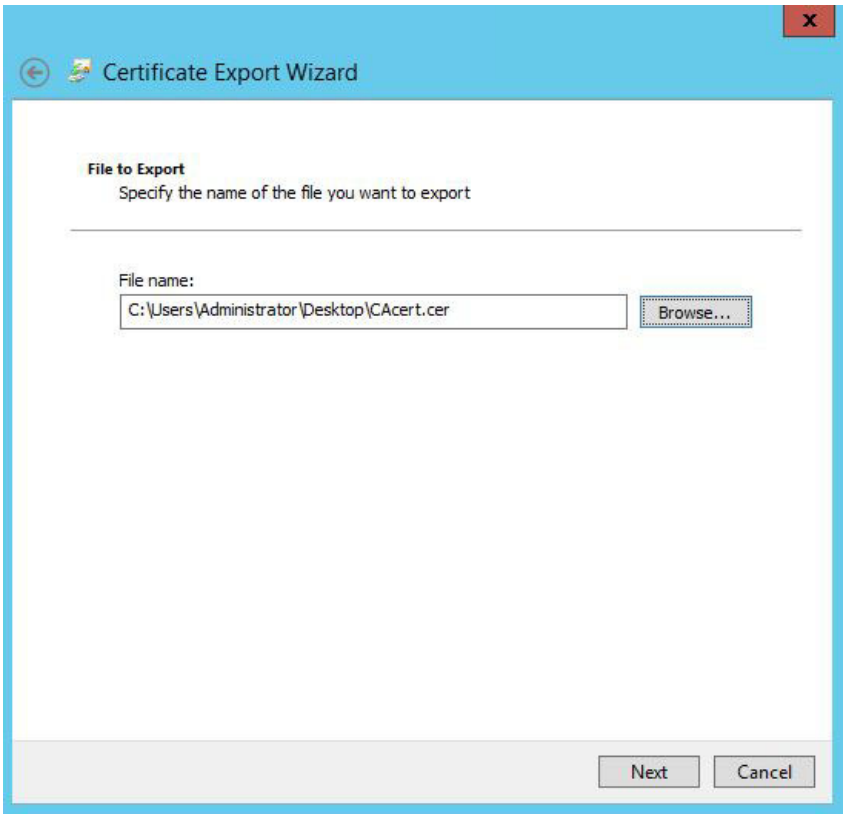
7 On the **Welcome to the Certificate Export Wizard** screen, click **Next**.

8 In the **Export File Format** screen, select **Base-64 encoded X.509 (.CER)**.

9 Click **Next**.


Certificate Export Wizard
Export File Format
Certificates can be exported in a variety of file formats.
Select the format you want to use:
DER encoded binary X.509 (.CER)
Base-64 encoded X.509 (.CER)
Cryptographic Message Syntax Standard - PKCS #7 Certificates (.P7B)
Include all certificates in the certification path if possible
Personal Information Exchange - PKCS #12 (.PFX)
Include all certificates in the certification path if possible
Delete the private key if the export is successful
Export all extended properties
Microsoft Serialized Certificate Store (.SST)
Next
Cancel

10 In the **File to Export** screen, browse to select the certificate file and then click **Next**.



11 On the **Completing the Certificate Export Wizard** screen, click **Finish**.

You have now successfully exported your certificate to the Base-64 X.509 file format.

Creating a certificate chain

When the certification path contains more than just the Web server certificate and a trusted root certificate, (that is, if there is an intermediate certificate), you must export all certificates in the path, so that the ISAPI Filter can use them. The ISAPI Filter requires only the Intermediate and root CA certificates; it does not require the SSL certificate.

To create a certificate chain

1. Export all intermediate certificates and the root certificate to the Base-64 file format. See “Exporting certificates to Base-64 format” on page 50 for instructions.
2. Edit the certificate files so that all certificates appear in the same file.

   Enter the certificates in order from the lowest Intermediate certificate in the chain up to the root CA certificate. The root CA certificate must be entered last. Ensure the file contains no empty lines.

   For example:

```
-----BEGIN CERTIFICATE-----
MIIDcjCCAlqgAwIBAgIEP2RuyzANBgkqhkiG9w0BAQUFADAxMQswCQYDVQQGEwJD..
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
mTA+d9P0nRQDufXAmZ8fni2ZsCPcejmqqjLFqc2WdrDC66YjDDt/V8Uf8KHURdbA..
-----END CERTIFICATE-----
```

You now have one certificate file, containing the CA certificate and any intermediate CA certificates, to use during the ISAPI Filter installation.

Configuring SSL between the authentication Web application and the OWA login service

This step applies only when you are installing the ISAPI Filter on the OWA server.

Skip this step:

- If you have SSL configured on OWA with a publicly trusted root certificate. The certificate is already trusted.

Import the trusted root certificate of the OWA domain into the local computer store of the computer where you plan to install the authentication application. This allows the authentication application to trust the OWA login service.

Copy the certificate to the computer where you plan to install the authentication application. Import it into the local computer store by following the procedure in “Importing certificates into the local computer store” on page 57.

Configuring SSL between the authentication Web application and Entrust Identity Enterprise

You must import the appropriate certificate or certificates into the local computer store of the computer where you plan to install the authentication application. This allows the authentication application to trust Entrust Identity Enterprise. The certificate to import depends on the setup of the Entrust Identity Enterprise Servers.

Entrust Identity Enterprise Servers	Certificates to import
One server with self-signed certificate	Certificate of the Entrust Identity Enterprise Server
One or more servers certified by a root CA	Root certificate of the root CA that signed the Entrust Identity Enterprise Servers' certificates
One or more servers with self-signed certificates	Certificates of all the Entrust Identity Enterprise Servers You must import a certificate from each server.

Copy the certificate to the computer where you plan to install the authentication application. Import it into the local computer store by the following the procedure in "Importing certificates into the local computer store" on page 57.

Importing certificates into the local computer store

To allow secure communication between the components for this solution, complete the following steps to import your trusted root CA certificates or self-signed Entrust Identity Enterprise Server certificates into the proper locations on the local computer store of the computer hosting the authentication application.

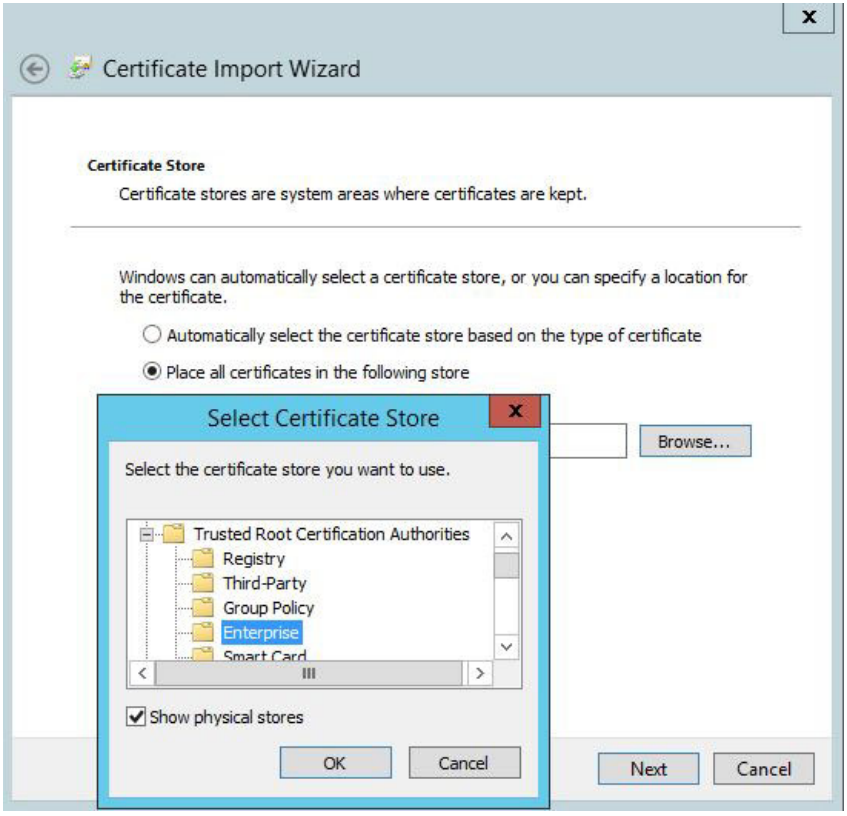
To import root CA certificates or self-signed Entrust Identity Enterprise Server certificates into the local computer store

1. Ensure you have copied the certificate file (`.cer`) to the computer where you will be importing the certificates.
2. Double-click the `.cer` file.
3. Click **Install Certificate**.

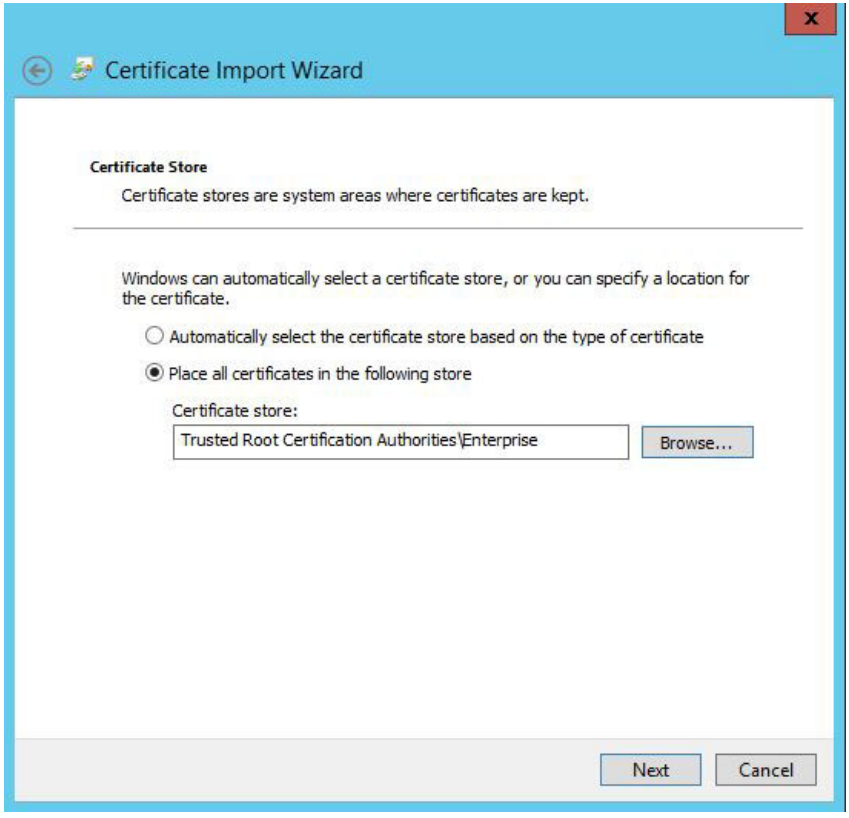
   The **Certificate Import Wizard** appears.

4 In the **Certificate Import Wizard**, select **Local Machine** and then click **Next**.

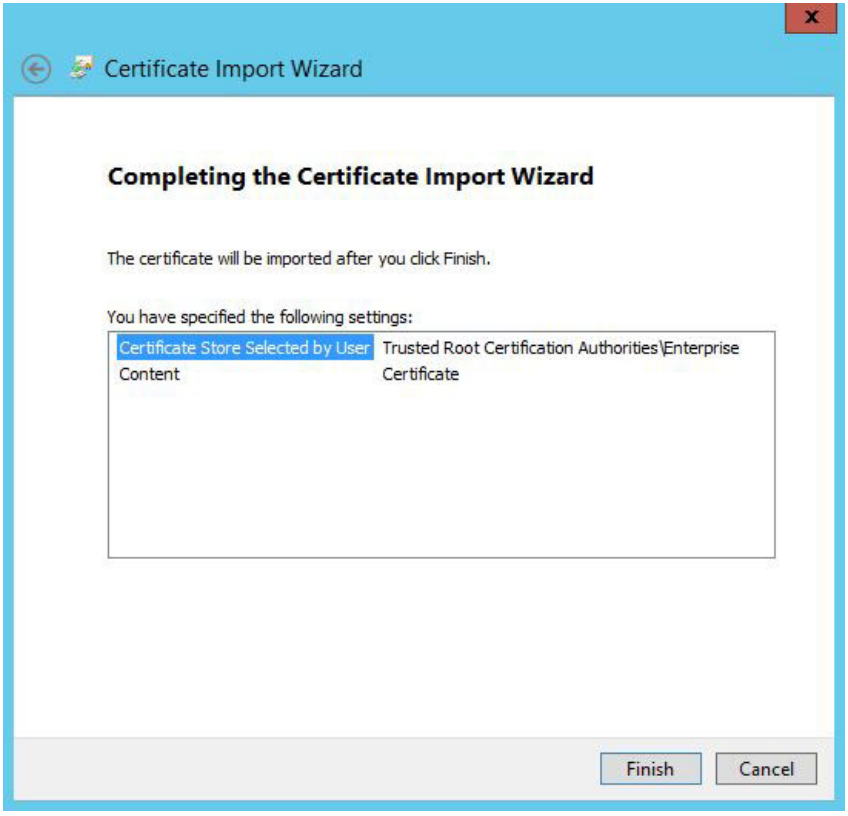
5 Select **Place all certificates in the following store**, and click **Browse** to access the certificate store.



6 Select the **Enterprise** certificate store and then click **OK**.



7 Click **Next**. The **Completing the Certificate Import Wizard** page appears.



8 Click **Finish**.

9 Select **Trusted Root Certification Authorities** > **Local Computer**, and click **OK**.

10 Click **Next** in the **Certificate Store** screen.

11 In the **Completing the Certificate Import Wizard** screen, click **Finish**.

You have successfully imported your certificate to the local computer store.

Enabling SSL on the IIS or ARR server

It is recommended that you enable Secure Sockets Layer (SSL) on the IIS server that hosts the authentication application. It is also recommended that you enable SSL on the ARR server, if you are using one.

You can skip this step if you have already enabled SSL on IIS or ARR.

To enable SSL on IIS 10

1. Create a certificate request for an SSL server certificate:
   a. Open IIS Manager from the Server Manager **Tools** menu.
   b. Under **Connection tasks**, select **Connect to localhost**.
   c. In the middle pane, double-click **Server Certificates**.
   d. In the **Actions** pane, click **Create Certificate Request** to launch the **Request Certificate** wizard.
   e. Follow the steps in the **Request Certificate** wizard to create a new certificate request.
2. Send the certificate request to a company such as Entrust, that can generate a Web server certificate for you. Go to www.entrust.net to submit the certificate request.
3. Import the Web server certificate into IIS:
   a. Open IIS Manager.
   b. Under **Connection tasks**, select **Connect to localhost**.
   c. In the middle pane, double-click **Server Certificates**.
   d. In the **Actions** pane, click **Complete Certificate Request** to launch the **Complete Certificate Request** wizard.
   e. Follow the steps in the **Complete Certificate Request** wizard to import the Web server certificate.
4. Add the HTTPS port to your Web site:
   a. Open IIS Manager.
   b. In the left pane, expand your server > **Sites** and select your Web site (for example, **Default Web Site**).
   c. In the **Actions** pane, click **Bindings**.

      The **Site Bindings** dialog box appears.
   d. Click **Add**.

      The **Add Site Binding** dialog box appears.
   e. In the **Type** drop-down list, select **https**.

f In the **IP address** drop-down list, select the IP address for the SSL-enabled Web site.

g In the **Port** field, enter the SSL port. By default, the SSL port is port 443.

h In the **SSL certificate** drop-down list, select the Web server certificate that you imported into IIS.

i Click **OK** to close the **Add Site Binding** dialog box.

j Click **OK** to close the **Site Bindings** dialog box.

5 To configure the Web site to require SSL:

a Open IIS Manager.

b Select your Web site (such as **Default Web Site**).

c From the middle pane, double-click **SSL Settings**.

d In the **SSL Settings** pane, select **Require SSL**.

e In the **Actions** pane, click **Apply**.

Configuring SharePoint server to work with ISAPI Filter in the IIS-only configuration

If you are planning to install ISAPI Filter on an IIS server running SharePoint, complete the following steps before installing ISAPI Filter.

- "Configuring SharePoint to use Integrated Windows Authentication" on page 63
- "Ensuring SharePoint URLs use fully qualified host names" on page 64

Configuring SharePoint to use Integrated Windows Authentication

The ISAPI Filter installed on IIS does not support SharePoint when it is configured for forms-based authentication or anonymous authentication.

You must ensure your SharePoint site is configured to use Integrated Windows Authentication (IWA).

Note:
SharePoint 2013 does not provide built-in support for client certificate authentication and, hence, ISAPI filter does not support certificate-based RBA.

This topic contains the following procedures:

- "To configure SharePoint Server 2013 and 2016 to use Integrated Windows Authentication" on page 63

To configure SharePoint Server 2013 and 2016 to use Integrated Windows Authentication

1. Open the SharePoint Central Administration Console.
2. Click the **Security** link.
3. Under **General Security**, click **Specify authentication providers**.
4. Click **Default zone**.
5. Enable the required options, as described in the table.

If you are using...	Do this...
SharePoint 2013 and 2016	• For **Claims Authentication Type**, select **Enable Windows Authentication**, select **Integrated Windows authentication**, and then select **NTLM**.

6 If you want to use the client integration feature, then for **Enable Client Integration?** click **Yes**.

If you want to use the **Enable Client Integration** feature, you must configure the ISAPI Filter to use persistent cookies after you install the ISAPI Filter. See "Configuring the filter to use persistent cookies with SharePoint" on page 104 for details.

Ensuring SharePoint URLs use fully qualified host names

When protecting your SharePoint site with ISAPI Filter, you must ensure that the SharePoint URLs contain fully qualified host names. When setting up the SharePoint site, this information is contained a field called the **Host Header**.

ISAPI Filter uses the cookie domain, which relies on the domain information picked up from the host header.

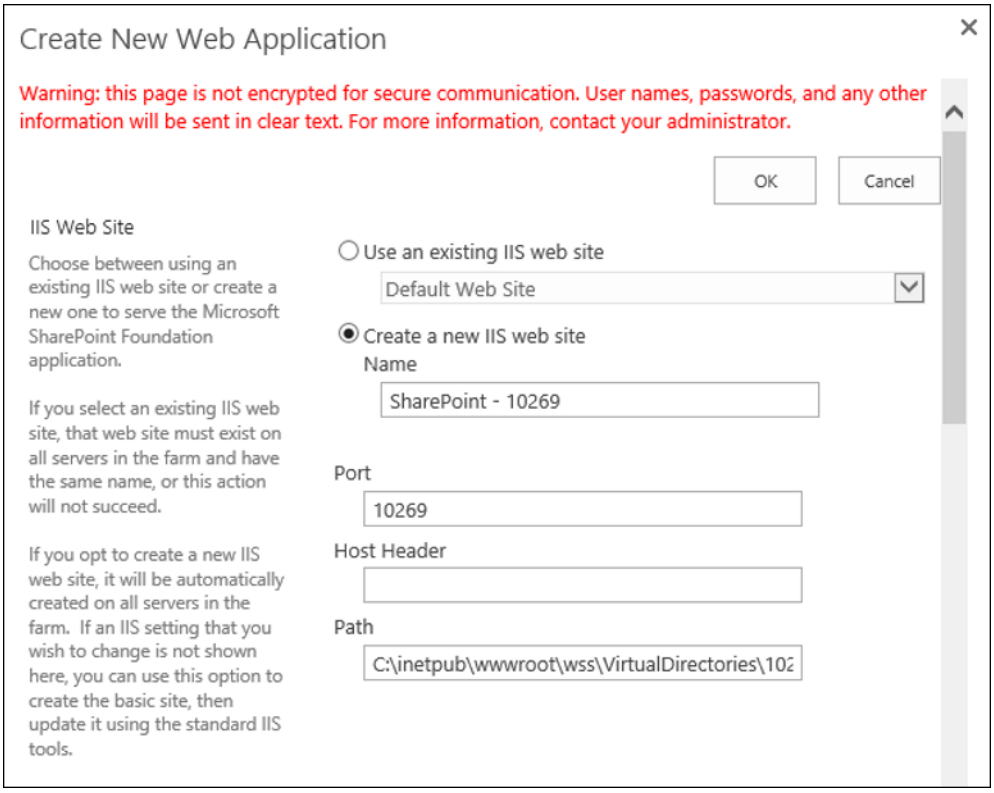
See the appropriate section below:

- "If you are creating a new SharePoint site" on page 64
- "If you have an existing SharePoint site" on page 65

If you are creating a new SharePoint site

If you are creating a new SharePoint site, you must include the fully qualified host header during creation.

See the SharePoint documentation and your SharePoint administrator for the details.



If you have an existing SharePoint site

If you have an existing SharePoint site, you must add a fully qualified domain name to the SharePoint URLs, if they are not already there. Rely on the advice of your SharePoint administrator in choosing the appropriate procedure for your situation.

Note:
It is recommended that you back up your SharePoint site before performing these procedures. See your SharePoint documentation, and talk to your SharePoint administrator for advice.

To prepare SharePoint to work with ISAPI Filter, follow one of these procedures:

- "To create an extended Web site for the SharePoint site" on page 66
- "To modify the existing SharePoint Web site by providing a new mapping" on page 67

To create an extended Web site for the SharePoint site

1 Launch the SharePoint Central Administration Application.

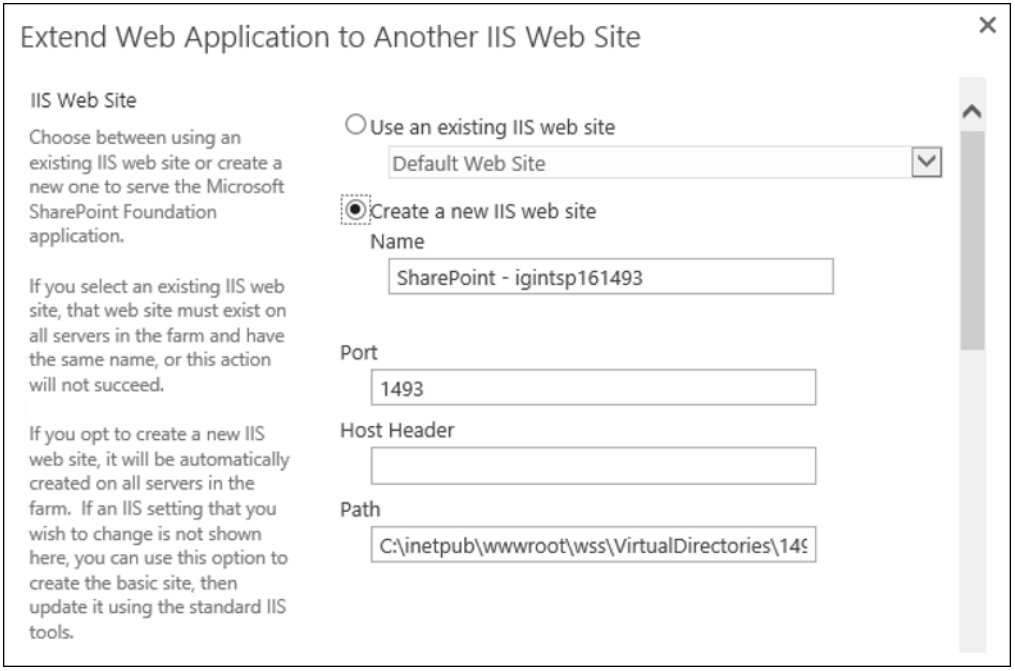
2 Click the **Application Management** tab.

The **Application Management** page appears.

3 Do one of the following, as described in the table.

If you are using...	Do this...
SharePoint 2013 and 2016	1 Click **Manage Web Applications**. 2 Select an existing Web application. 3 Click **Extend**.

The **Extend Web Application to Another IIS Web Site** page appears.



4 From the **Web Application** drop-down menu, select the site you want to extend.

   If it is not displayed in the list, select **Change Web Application**, and select the Web application from the **Select Web Application** page that appears.

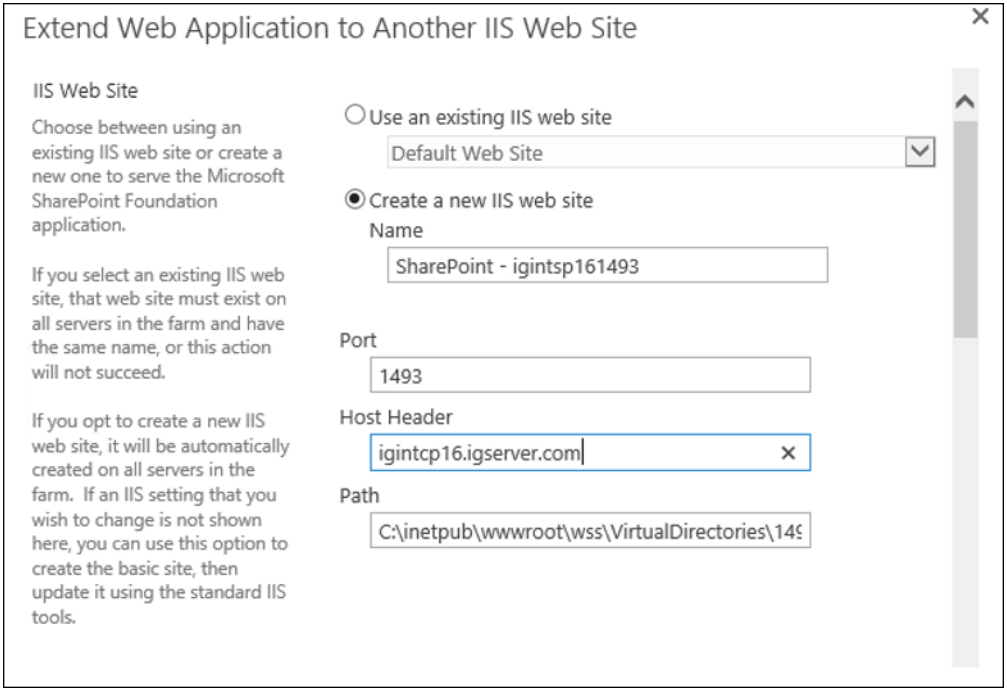
5 Verify that the correct SharePoint site is now displayed on the **Web Application** list.

6 Select **Create a new IIS Web site**.

7 Under **Port**, check the port number. The default ports are 80 and 443.

8 In **Host name**, enter the fully qualified SharePoint host name in the form `sitename.domain.com`.

   For example: `sharepoint.anycorp.com`.



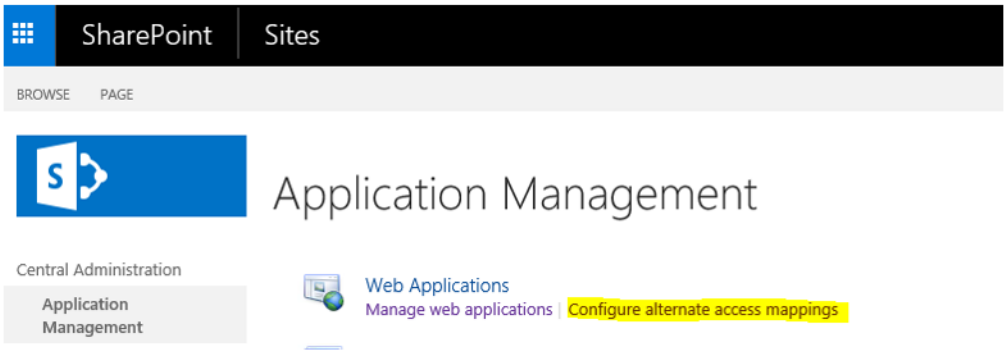
9 Click **OK**.

To modify the existing SharePoint Web site by providing a new mapping

1 Launch the SharePoint Central Administration Application.

2 Click the **Application Management** tab.

The **Application Management** page appears.



3 Click **Configure Alternate Access Mappings.**

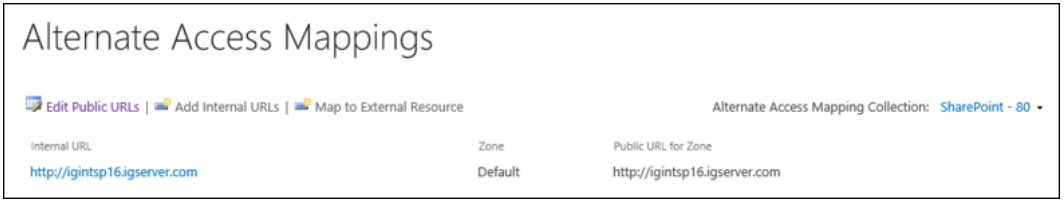
The **Alternate Access Mappings** screen appears.

4 Check that the mapping collection you want to modify is displayed, and select it.

If it is not displayed, on the **Alternate Access Mapping Collection** drop down menu, click **Change alternate access mapping collection**.

On the **Select an Alternate Access Mapping Collection** screen that appears, select the SharePoint site for which you are changing mappings.

5 On the **Alternate Access Mapping** screen, click **Edit Public URLs**.

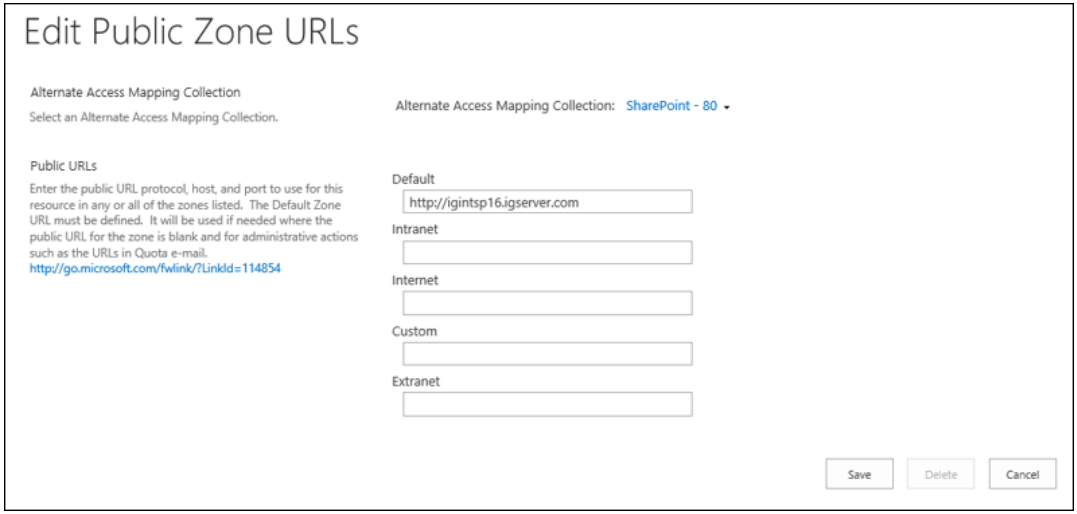


6 In the **Public URLs** section, add new URLs or edit existing URLs in any of the following based on your SharePoint setup.

- Default
- Intranet
- Internet
- Custom
- Extranet

The new or updated URL must be of the format:

[http|https]://sitename.domain.com:<portnumber>/



7 Click **Save**.

After installing the ISAPI Filter on SharePoint 2016 or 2013 and accessing the resource or created Web site, you may see the XML Parsing Error on the browser. To fix this issue, do the following:

1 Disable the **Client Object Model Permission Requirement** option by following the steps in SharePoint administration portal.

2 Log in to the **SharePoint Administration Portal**.

3 Click A**pplication Management > Manage web applications**.

4 Select the site you created.

5 Click **Authentication Providers**.

6 Click **Default**.

7 Uncheck the box **Require Use Remote Interfaces permission**.

8 Click **Save** to apply the changes.

Configure ISAPI Filter for Passkey/FIDO2 authentication

Passkey/FIDO2 authentication requires users to respond to the notification sent to their mobile device or passkey token.

Topics in this section:

Note:
This procedure assumes that you have already integrated ISAPI Filter with IDaaS. See Integrate ISAPI Filter Adapter in the I*dentity as a Service Technical Integrations Guides* for more information.

Configure ISAPI Filter for Passkey/FIDO2 with Identity as a Service

To use passkey/FIDO2 authentication, the user must match the relying party ID on the IDaaS page with the ISAPI Filter configuration file.

To configure ISAPI Filter for Passkey/FIDO2 with IDaaS

1. Configure Passkey/FIDO2 for multifactor authentication. See Modify Passkey/FIDO2 authenticator settings in the *IDaaS Administrator Help*.
2. Select **Enable Passkey/FIDO2 Allowlist** and include the **Relying Party ID** you configured when you installed the ISAPI Filter Adapter for Identity as a Service. See Modify Passkey/FIDO2 authenticator settings in the *IDaaS Administrator Help*.
3. Create a custom user login Authentication Flow to enable Passkey/FIDO2 for second-factor authentication. See Create authentication flows in the *IDaaS Administrator Help*.
4. Create a resource rule that includes the Authentication Flow that enables Passkey/FIDO2 for second-factor authentication. See Create a resource rule in the *IDaaS Administrator help*.

Configure ISAPI Filter for Passkey/FIDO2 with Entrust Identity Enterprise

Passkey/FIDO2 authentication requires users to respond to the notification they receive on their mobile device or token. The Relying party ID must be set to enable the following, as required:

- Passkey authentication to Entrust Identity Self-Service Module
- Self-administration to allow passkey to be registration

The relying party is associated with an origin (the allowed origin), which is either a single host or any host that belongs to a domain or associated subdomains.

Complete the following procedures using documentation available on Entrust TrustedCare:

- Configure Passkey/FIDO2 for multifactor authentication
- Register Passkey/FIDO2 token with Entrust Identity Self-Service Module

To access the documentation on Trusted Care

1. Go to https://trustedcare.entrust.com and enter your username and password.
2. Click **Products**.
3. Scroll to **Entrust Enterprise**.

Configure Passkey/FIDO2 for multifactor authentication

1. In TrustedCare, go to **Products > Entrust Enterprise > Self-Service Module**.
2. Click the **Documents** tab.
3. Click **View** to open the *Entrust Identity Self-Service Module Installation and Configuration Guide*.
4. In the table of contents, go to **Properties > FIDO2 Passkey Configuration**.
5. Follow the procedure in the section "FIDO2 Passkey Configuration" to configure Passkey/FIDO2 for multifactor authentication.

Note:
In the Passkey Relying Party ID field, enter the Relying Party ID that matches the Passkey/FIDO2 token that was also used when you installed the ISAPI Filter Adapter 13.0 for Entrust Identity Enterprise.

3

Installing Entrust ISAPI Filter

This chapter describes the procedures for installing the ISAPI Filter solution on different configurations.

This chapter describes how to apply the full-featured ISAPI Filter solution.

To upgrade an existing ISAPI Filter installation, see “Upgrading from previous versions of Entrust ISAPI Filter” on page 276.

Topics in this chapter:

Installing ISAPI Filter on an IIS server

Complete the following steps to perform a full installation and set up of the ISAPI Filter and authentication application on an IIS server.

If you want to enable logging during the installation, start the installer from a command line following the instructions in “Appendix C: Enabling logging during the installation of the ISAPI Filter solution” on page 287.

To install the Entrust ISAPI Filter on an IIS server

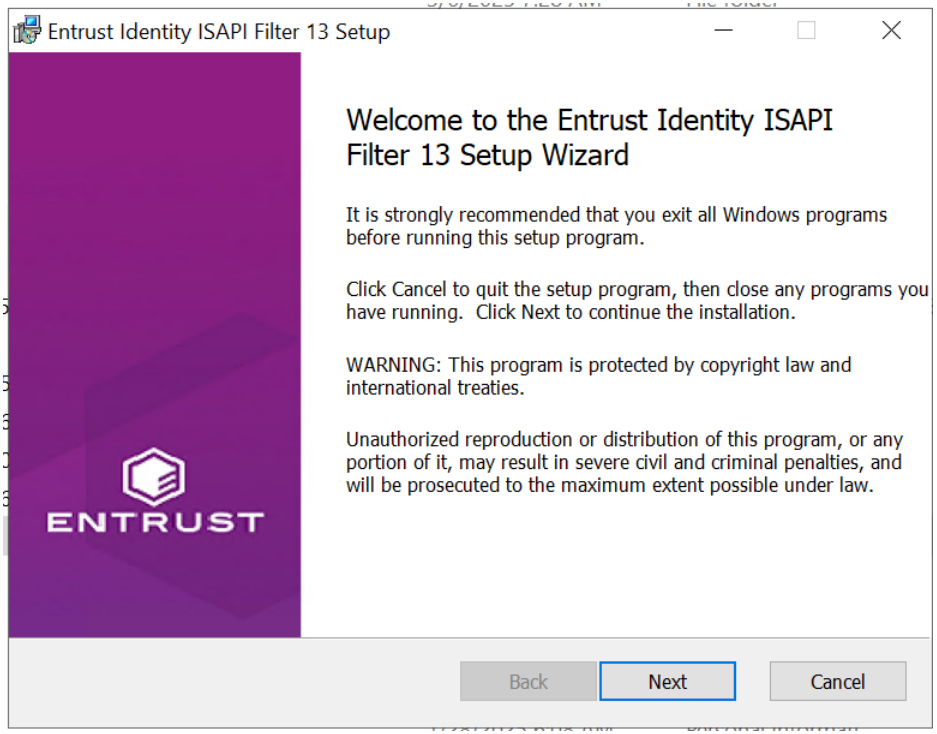
1 Download the Entrust ISAPI Filter 13.0 software from Entrust TrustedCare.

https://trustedcare.entrust.com

2 Copy the software to the IIS server.

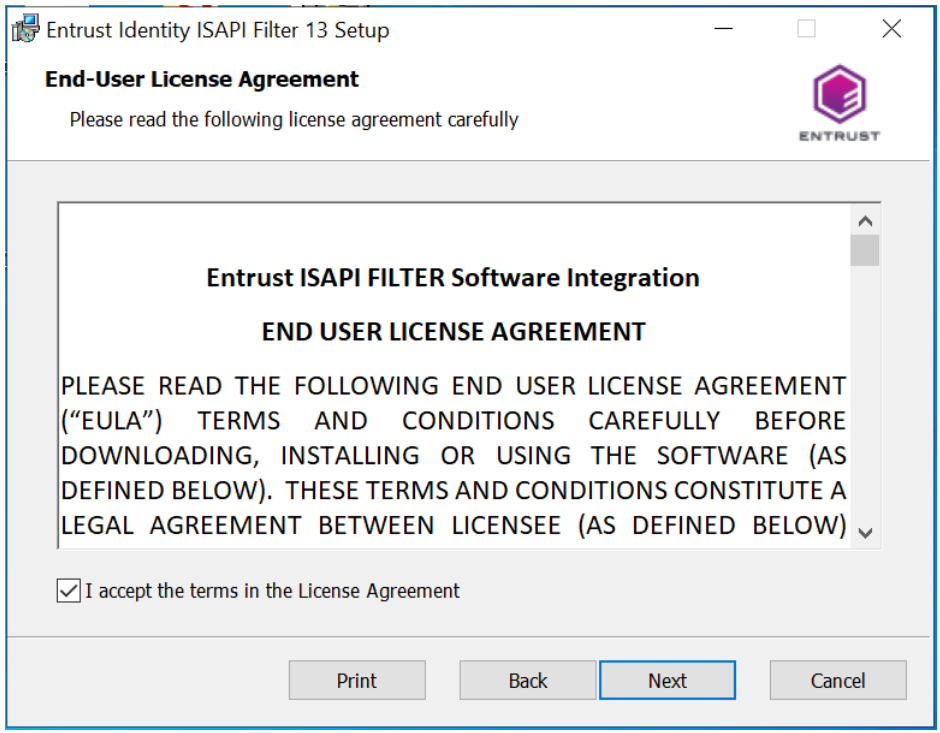
3 Double-click the `EI_ISAPI_Filter_13.0.msi` installer file.

The Entrust ISAPI Filter installation wizard appears.



4 Click **Next** to begin the installation.

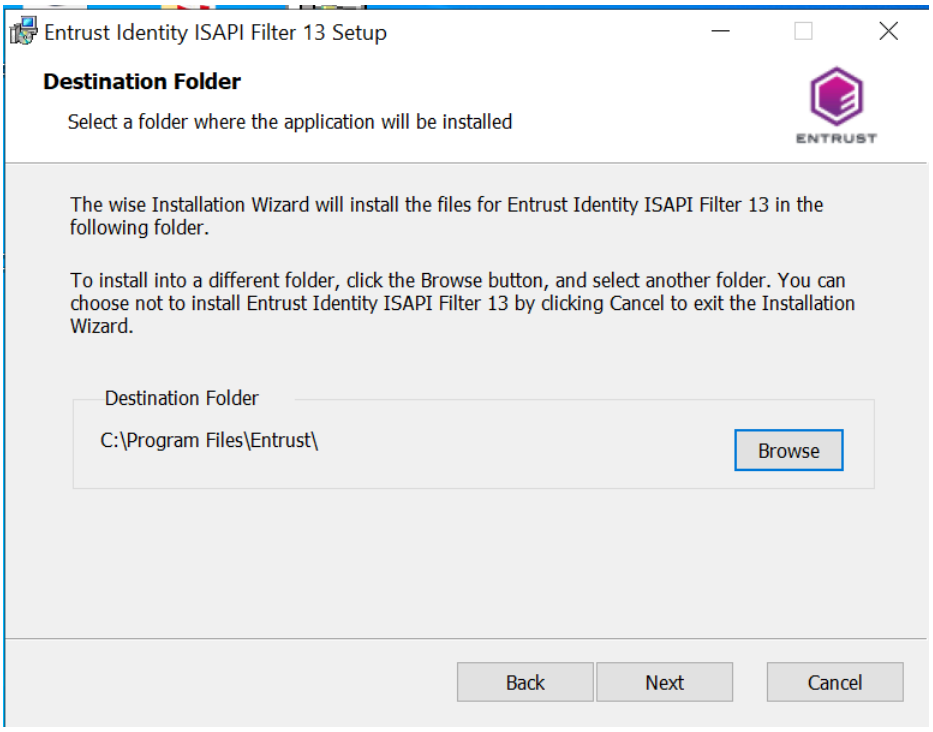
The **License Agreement** appears.



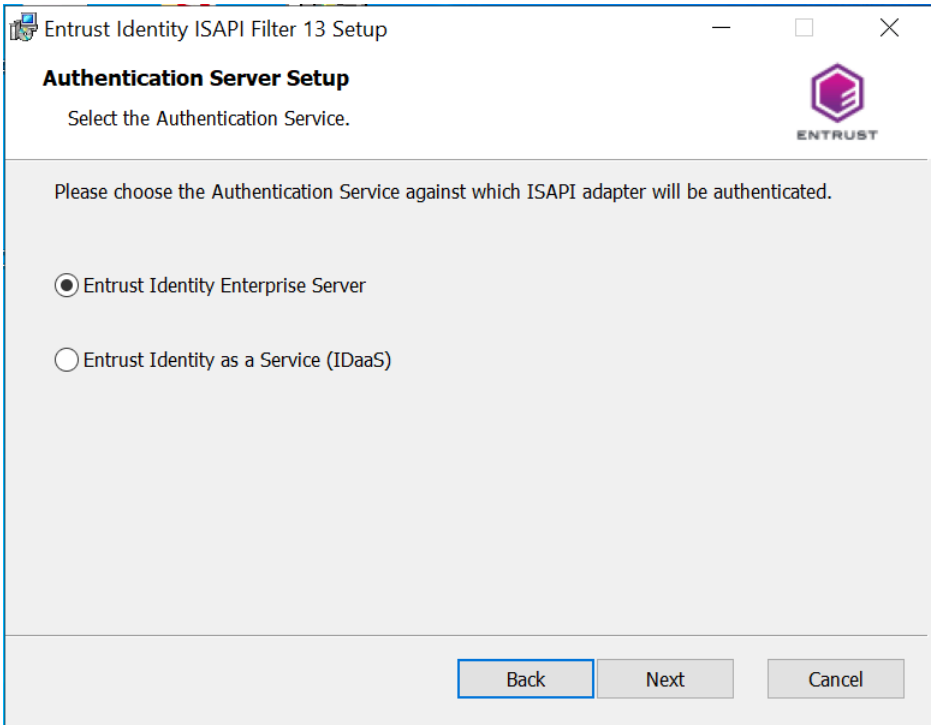
5 Read the license agreement for Entrust Identity Enterprise software carefully, and select **I accept the license agreement**.

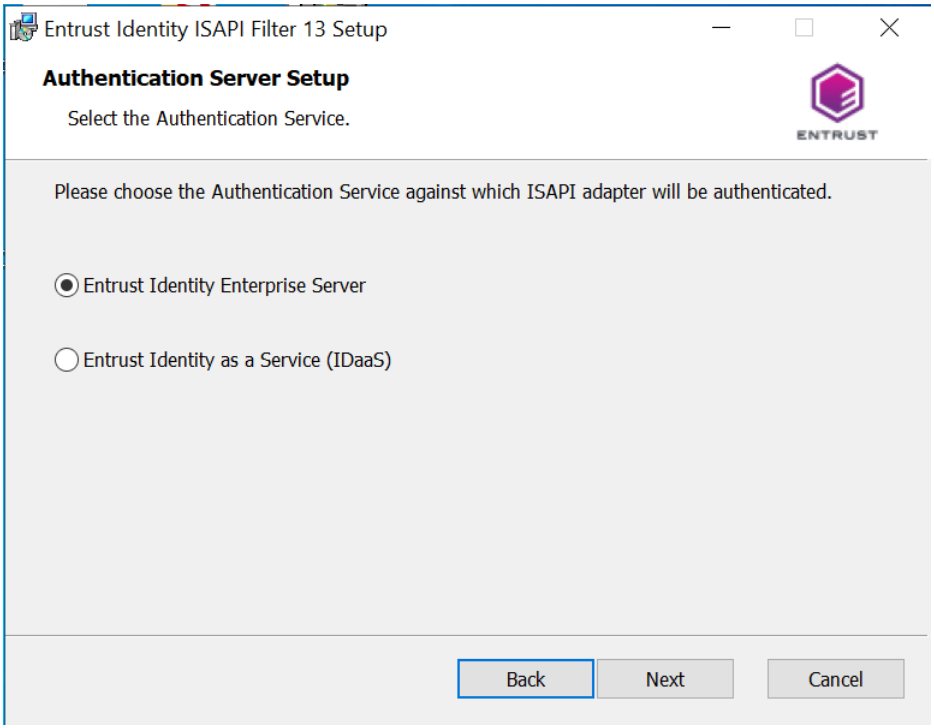
6 Click **Next**.

The **Destination Folder** screen appears.



7 Select the folder where you want to install the application, and click **Next**. The **Authentication Server Setup** page appears.


Entrust Identity ISAPI Filter 13 Setup
Authentication Server Setup
Select the Authentication Service.
ENTRUST
Please choose the Authentication Service against which ISAPI adapter will be authenticated.
Entrust Identity Enterprise Server
Entrust Identity as a Service (IDaaS)
Back
Next
Cancel



8 Select one of the following options:

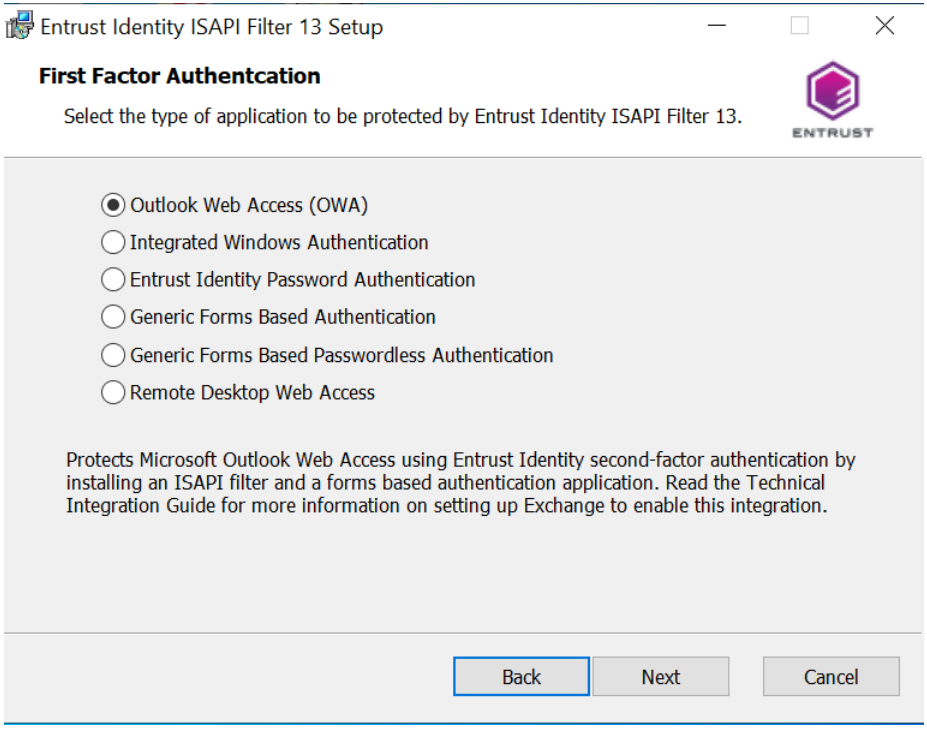
- **Entrust IdentityEnterprise Server** to install ISAPI Filter for Entrust Identity Enterprise authentication.

  -or-

- **Entrust Identity as a Service (IDaaS)** to install ISAPI Filter for Identity as a Service authentication.

9 Click **Next**.

10 The **First Factor Authentication page** appears.

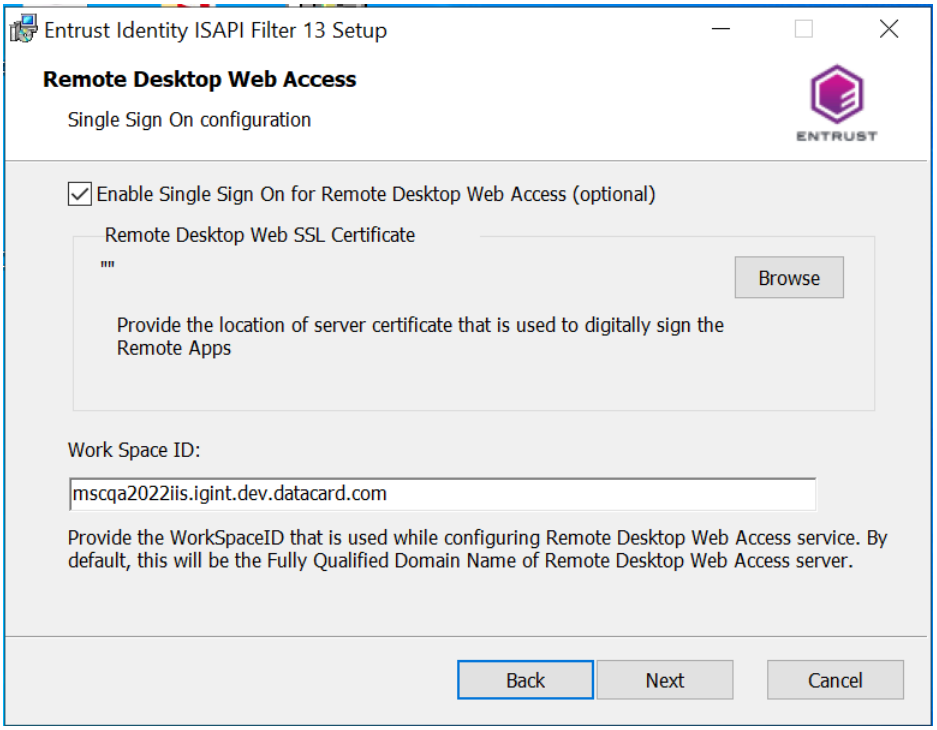


11 Select one of the following options and click **Next**:

- **Outlook Web Access (OWA)**. Proceed to Step 13.
- **Integrated Windows Authentication**. Proceed to Step 14.

  You must select this option if you are integrating with SharePoint.
- **Entrust Password Authentication**. Proceed to Step 13.
- **Generic Forms Based Authentication**. Proceed to Step 13.
- **Generic Forms Based Passwordless Authentication**. This option is “passwordless” because it authenticates a user using a token and a PVN, without a password required. Proceed to Step 14.
- **Remote Desktop Web Access**. Proceed to.Step 12

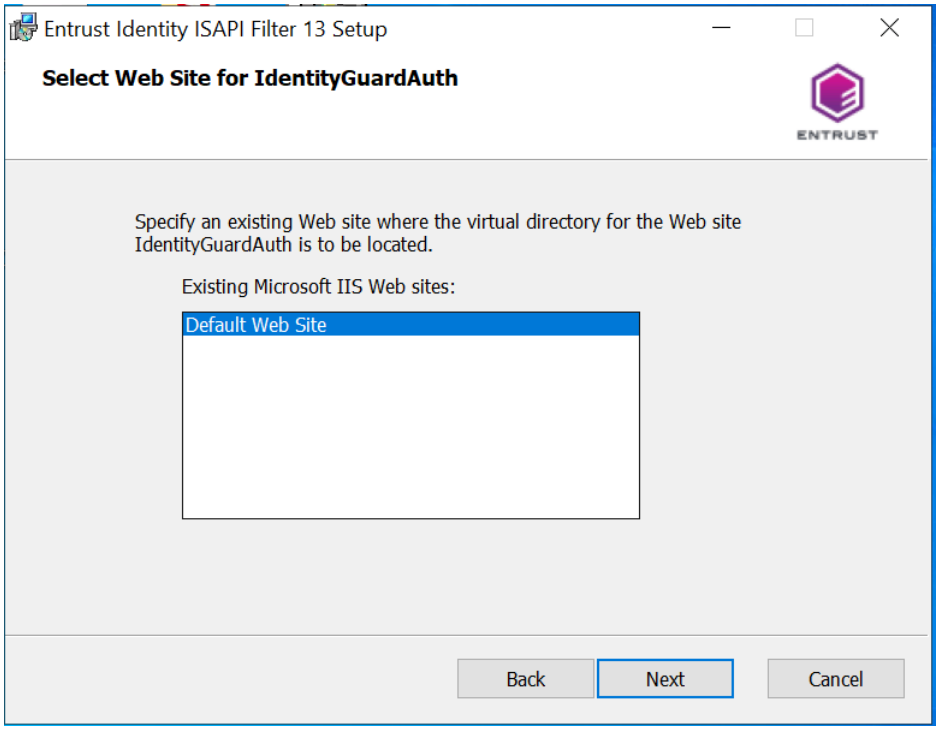
12 If you selected **Remote Desktop Web Access** as your first-factor authentication type, the **Remote Desktop Web Access** page appears:



a Select whether to enable Single Sign On (SSO) and browse for the remote desktop SSL certificate. The SSO feature requires a certificate. SSO is available with Remote Desktop Connection 7.0 or later.

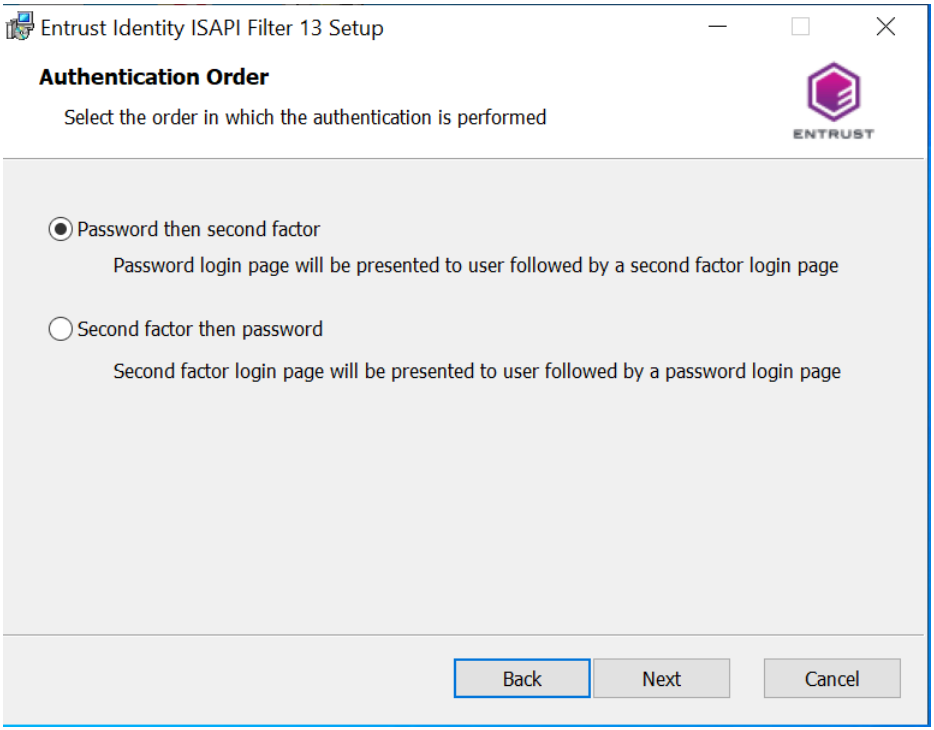
b Enter the fully qualified domain name of the Remote Desktop Web Access server.

c Click **Next**. The **Select Web Site for IdentityGuardAuth** page appears.



d Select the Web site where the virtual directory for the Web site IdentityGuard Auth is the be located and click **Next**.

13 The **Authentication Order** page appears. This page applies to Entrust Identity Enterprise Server installation to allow Entrust Identity Enterprise authentication.



On this page you can reverse the authentication order if you are installing for OWA, Entrust Identity Enterprise Password Authentication, Generic Forms Based Authentication, or Remote Desktop Web Access.

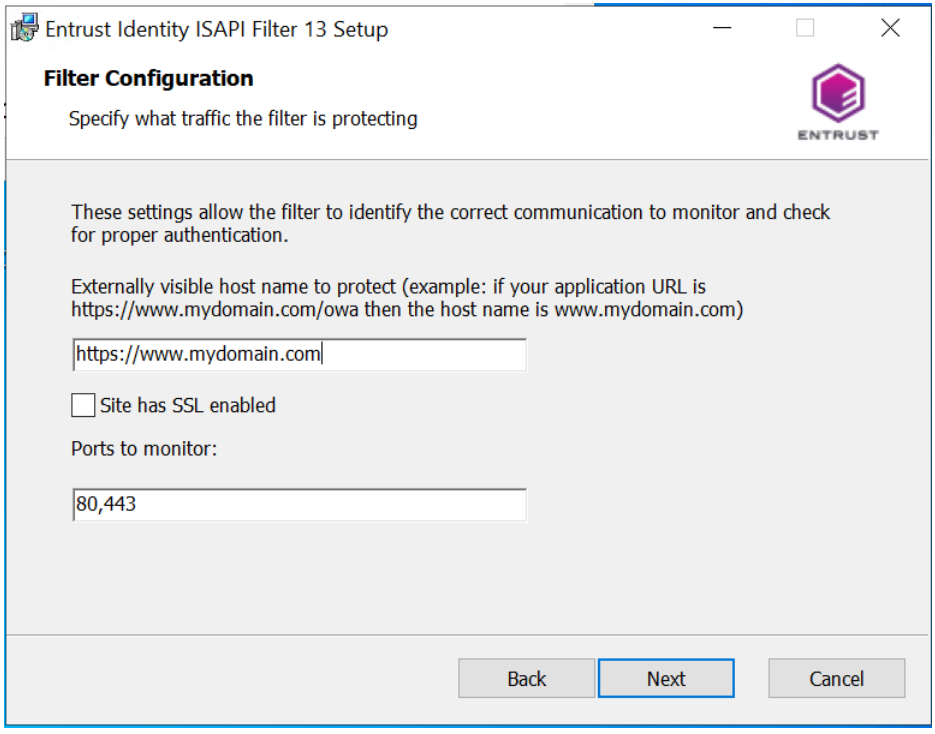
a Select one of the authentication order options.

- **Password then second factor**.
- **Second factor then password**. In this case, the **Authentication Mode** screen appears.

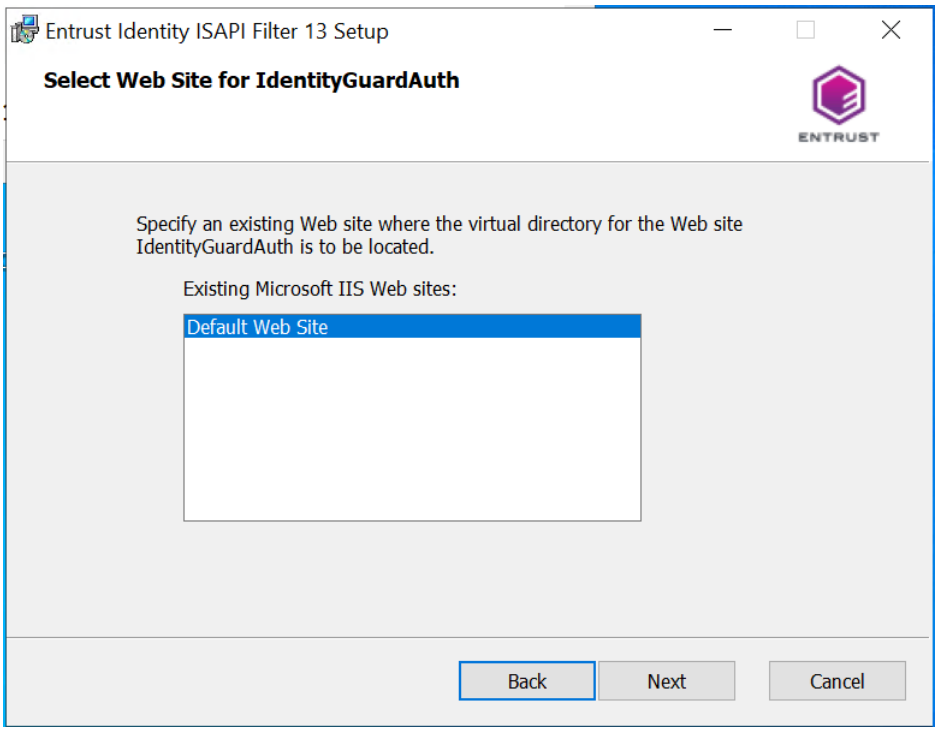
b Click **Next**.

If you selected **Password then second factor**, proceed to Step 14. Otherwise, proceed to the next step.

The **Filter Configuration** page appears.



14 Select a **Web site for the IdentityGuardAuth** virtual directory.

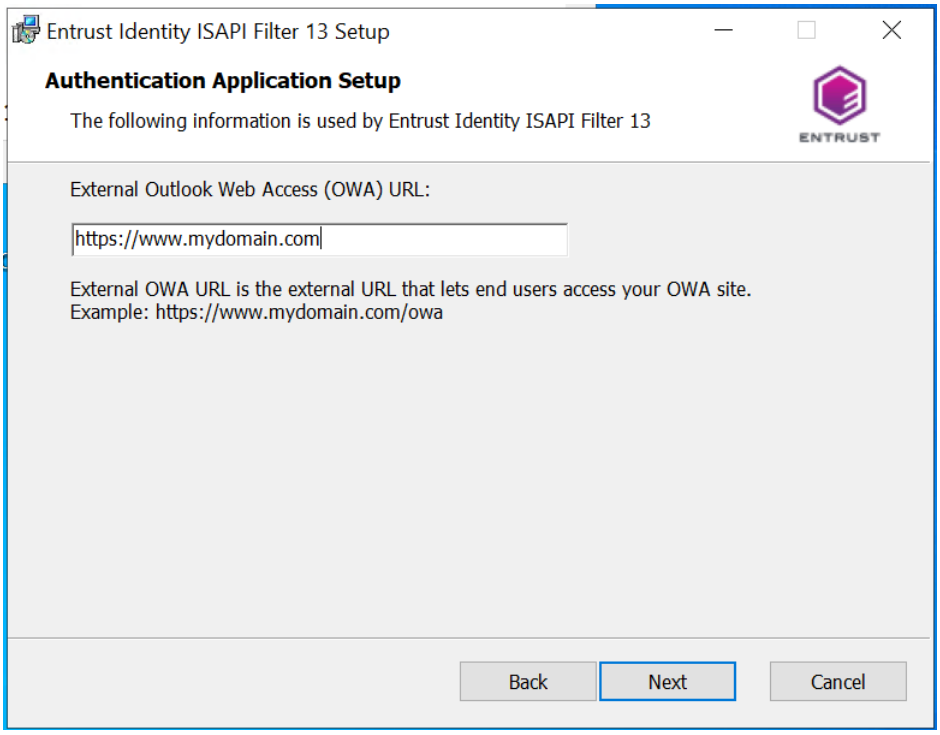


If you are using SharePoint, select the user-accessible SharePoint Web site (not the administration site).

15 Click **Next**.

If you are not installing for OWA, proceed to Step 17.

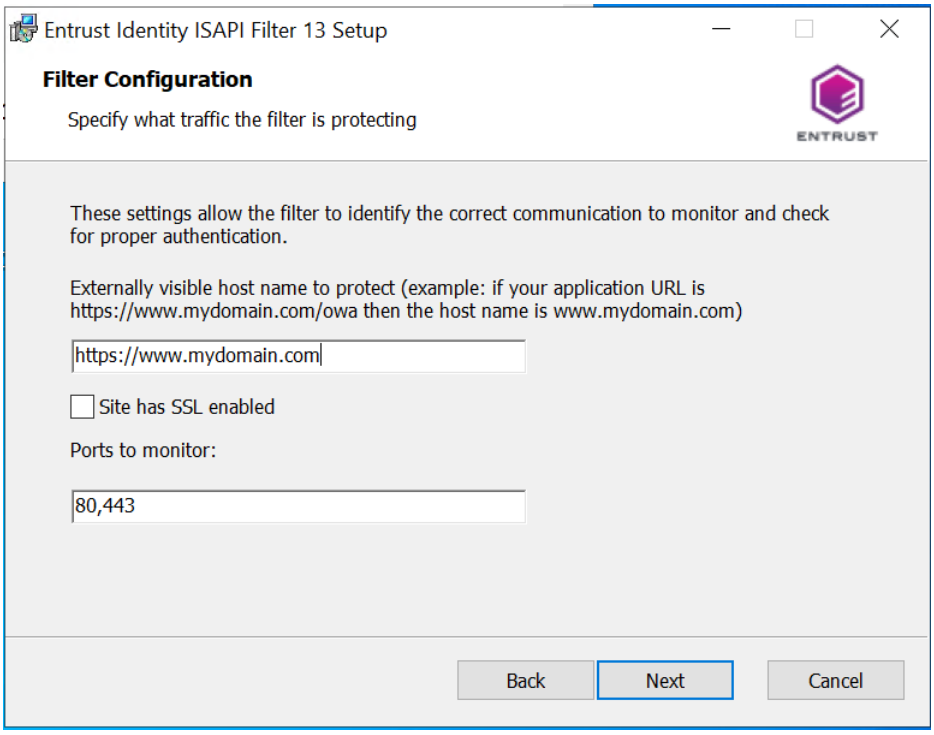
16 If you selected OWA on the **First Factor Authentication** screen, the **Authentication Application Setup** page appears.



a Enter the URL for your OWA site in the **External Outlook Web Access (OWA) URL** field.

b Click **Next**.

17 The **Filter Configuration** page appears.



a Enter the externally visible host name you want to protect. If you are deploying in an ARR environment (see "ARR architecture" on page 33), enter the host name of the ARR server.

Examples:

```
AuthApp.corp.com
```

```
ARRserver.example.com
```

b If the site uses SSL, select **Site has SSL enabled**.

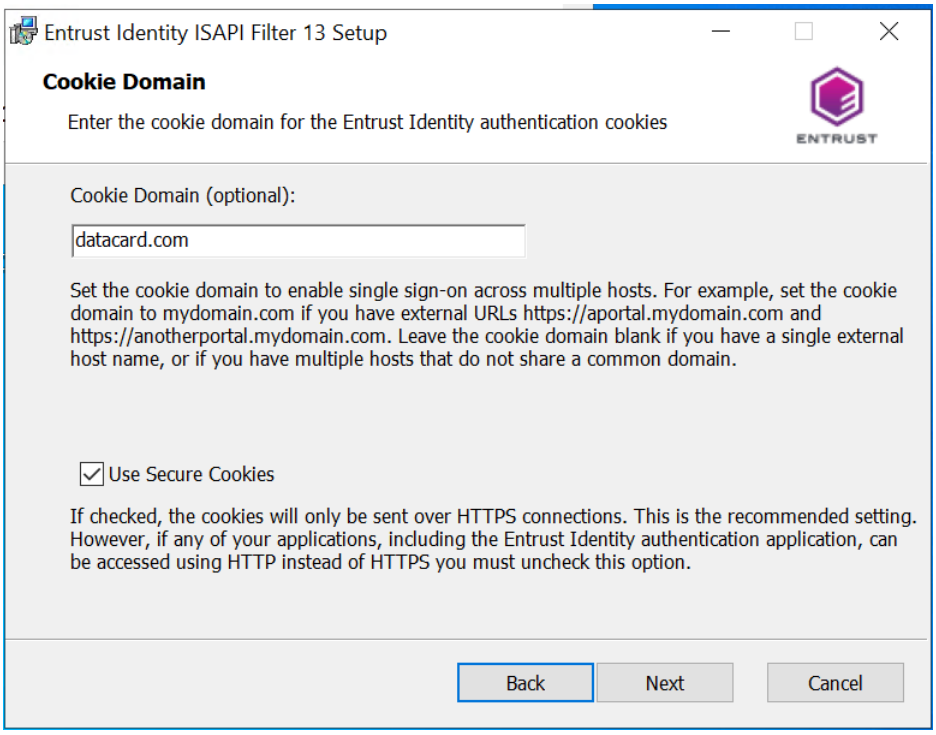
c In the **Ports to monitor**, enter a comma-separated list of ports to monitor. For example:

```
80,443,24
```

The default ports (80 and 443) are the default ports for HTTP and HTTPS. Port 24 should be specified if you are deploying in an ARR architecture. For details, see "ARR architecture" on page 33.

d Click **Next**.

18 The **Cookie Domain** page appears.



a Optional. If you want to enable single sign-on across multiple hosts, enter your cookie domain into the **Cookie Domain** field.

Leave the **Cookie Domain** field blank if any of the following conditions is true:

– You have only one external host name.

– You have multiple hosts, but they do not share a common domain.

Note:
For more information about how cookies are used in this solution, see "Appendix A: How the filter works" on page 279 and "Forms-based authentication" on page 21.

If you enter the cookie domain for Entrust Identity Enterprise cookies, the cookie domain must be the external domain that is accessed from the user's browser. If multiple subdomains are used, then enter the base domain.

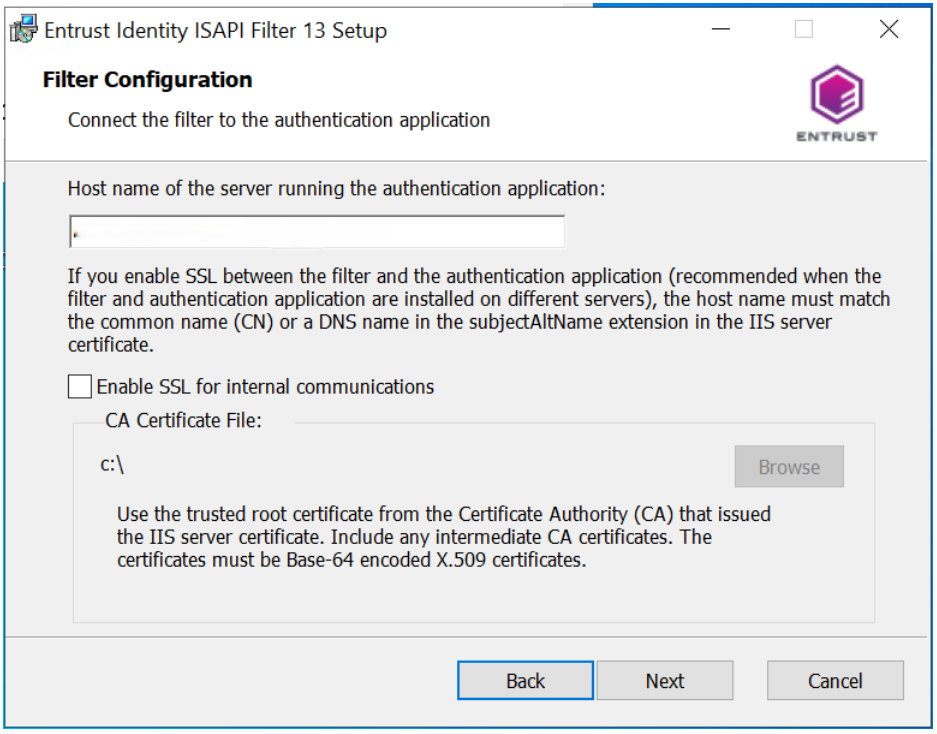
For example, if you use subdomains `owa.anycorp.com` and `sharepoint.anycorp.com`, enter `anycorp.com` into the **Cookie Domain** field.

b If your protected applications and the authentication application are all accessed only over HTTPS connections, select **Use secure cookies** to send the cookies over an HTTPS connection.

If any of your protected applications, including the Entrust Identity Enterprise authentication application, can be accessed by a user using HTTP, you must deselect **Use secure cookies**.

c Click **Next**.

19 The **Filter Configuration** page appears.



a If your Web site has SSL enabled, select **Enable SSL for internal communications**. Enabling SSL is recommended when the ISAPI Filter and the authentication application are installed on different servers.

b If you selected **Enable SSL for internal communications** in Step a, click **Browse** to locate and select:

– the trusted root Certification Authority (CA) certificate that issued the IIS server certificate (for IIS-only and OWA deployments)

OR

– the trusted root Certification Authority (CA) certificate that issued the ARR server certificate (or ARR deployments)

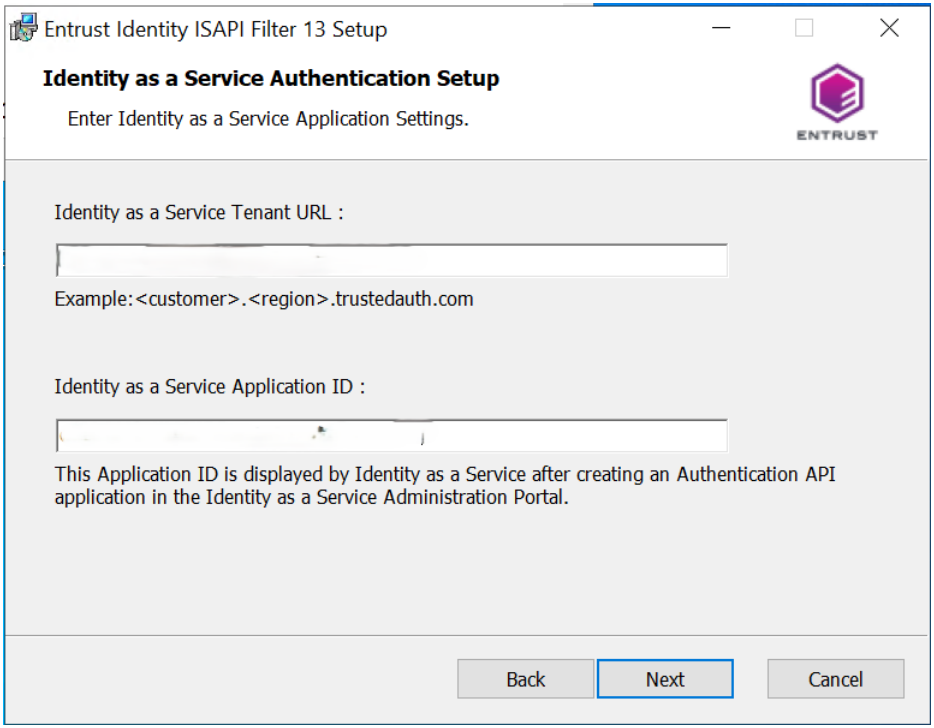
c Click **Next**.

The **Authentication Application Setup** page appears. The page you see depends on whether you are installing for Entrust Identity Enterprise Server or Identity as a Service.

– If you installing ISAPI Filter for Identity as a Service, go to Step 20.

– If you are installing for Entrust Identity Enterprise Server, go to Step 21.

20 In the Identity as a Service **Authentication Application Setup** page, complete the following:



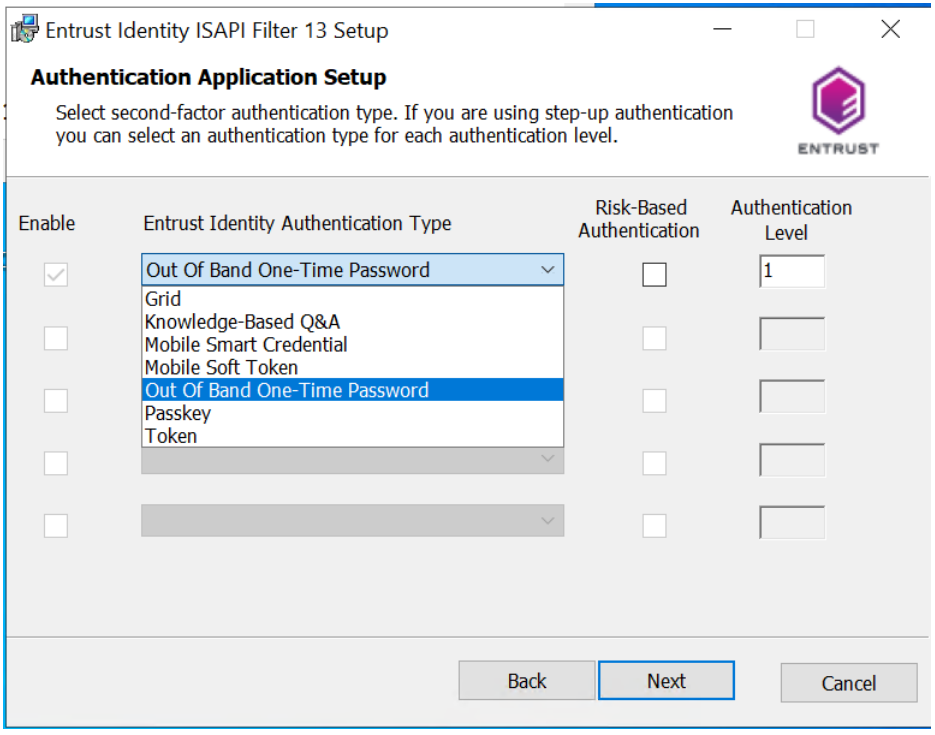
a In the **Identity as a Service Tenant URL** field, enter the Identity as a Service Tenant URL. For example, `<my_company>.<region>.trustedauthdev.com`.

b In the **Application ID** field, enter the Application ID that what generated when you created the Authentication API in Identity as a Service (see the

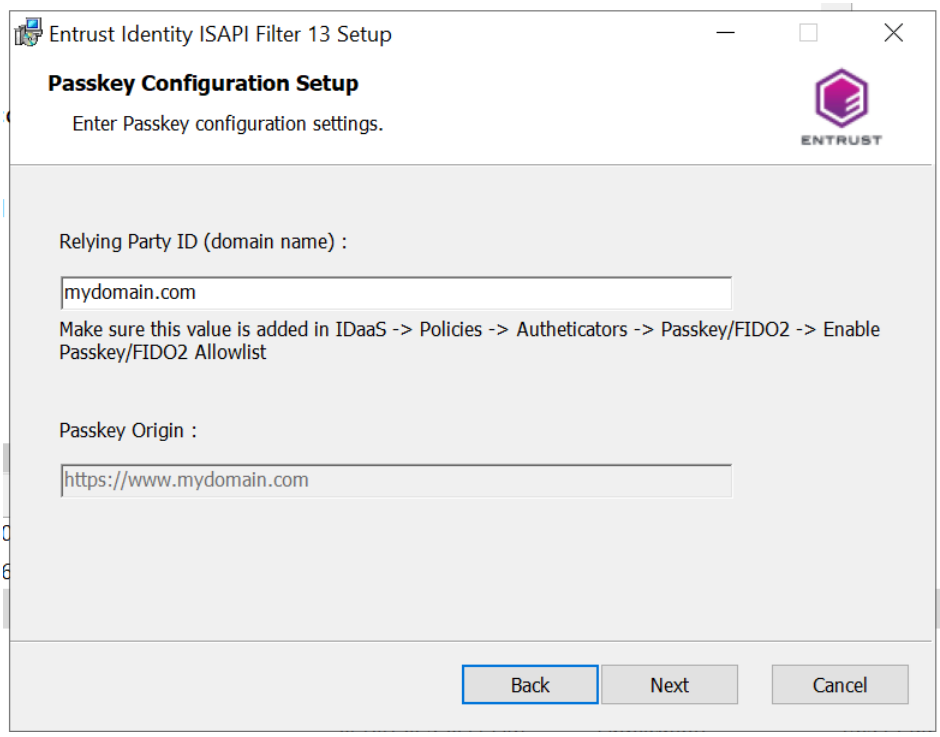
section, Integrate Identity as a Service ISAPI Filter in the *Identity as a Service Administrator Online Help*).

c Click **Next**. The **Authentication Application Setup** page appears.

d To configure passkey authentication, select **Passkey** from the **Authentication Type** drop-down list.



e The **Passkey Configuration Setup** page appears.



f In the **Relying Party ID** enter the Relying Party ID that the users have used to register the Passkey. It can be a qualified domain name (FQDN) or the domain itself. The following provides an example of valid Relying Party ID values:

`<cluster name>.<subdomain>.<domain.com>.`

`<subdomain>.<domain.com>`

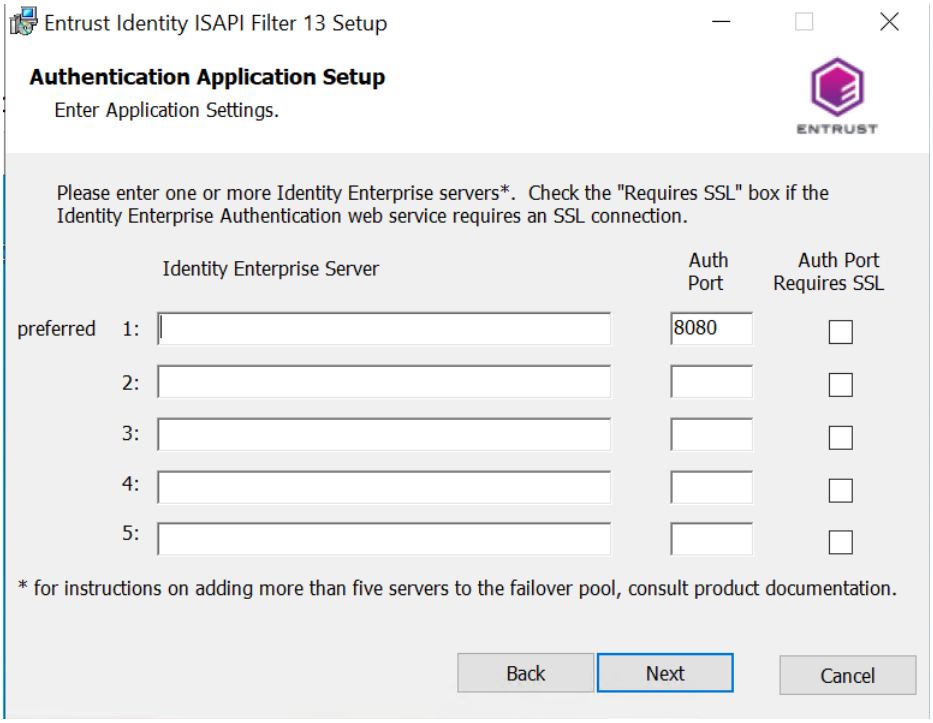
`<domain.com>`

g You must add Relying Party ID value to the Passkey/FIDO2 Allowlist in IDaaS. See Modify Passkey/FIDO2 authenticators in the *IDaaS Administrator help*. The **Passkey Origin** is populated by default

h Click **Next**.

i Proceed to Step 24.

21 In the Entrust Identity Enterprise Server **Authentication Application Setup** page, complete the following:



a Enter the host names of one or more Entrust Identity Enterprise Servers in the **IdentityGuard Server** fields.

If you need to configure more than five Entrust Identity Enterprise Servers, you can add the extra servers after installation is complete. See "Configuring failover for Entrust Identity Enterprise Servers" on page 250.

Note:
The **preferred** Entrust Identity Enterprise Server (number 1) is the Primary Entrust Identity Enterprise Server in a high availability or failover scenario.

b Enter the port number being used by the Entrust Identity Enterprise authentication service in the **Auth Port** field.

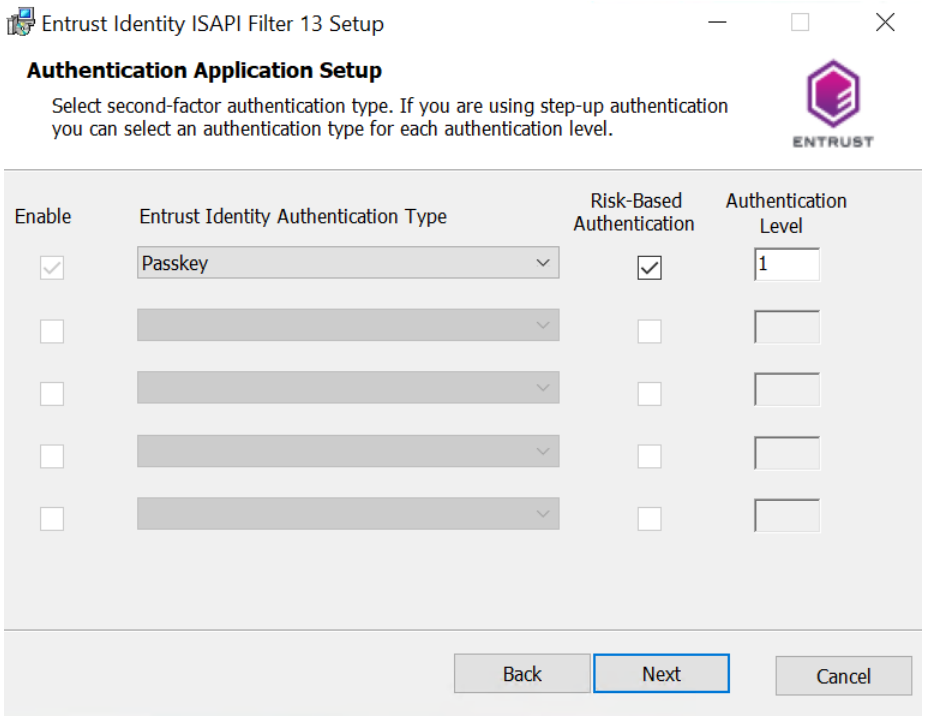
Default port number assignments:

```
8080    non-SSL
8443    SSL
```

c If needed, select **Auth Port Requires SSL**.

Note:
If you select SSL, you must already have imported the appropriate certificates into the local computer store of the computer where you are installing the authentication application. See "Configuring SSL between the authentication Web application and Entrust Identity Enterprise" on page 56.

d Click **Next**. The **Authentication Application Setup** page appears, unless you selected **Generic Forms Based Passwordless Authentication** as your first-factor authenticator or selected **One step** as your authentication mode, in which case, skip to Step 24.



a For each authentication level, select **Enable** and choose an authentication type under **IdentityGuard Authentication Type**.

b Select the **Risk-Based Authentication** check box if you want to enable IP address validation or machine authentication.

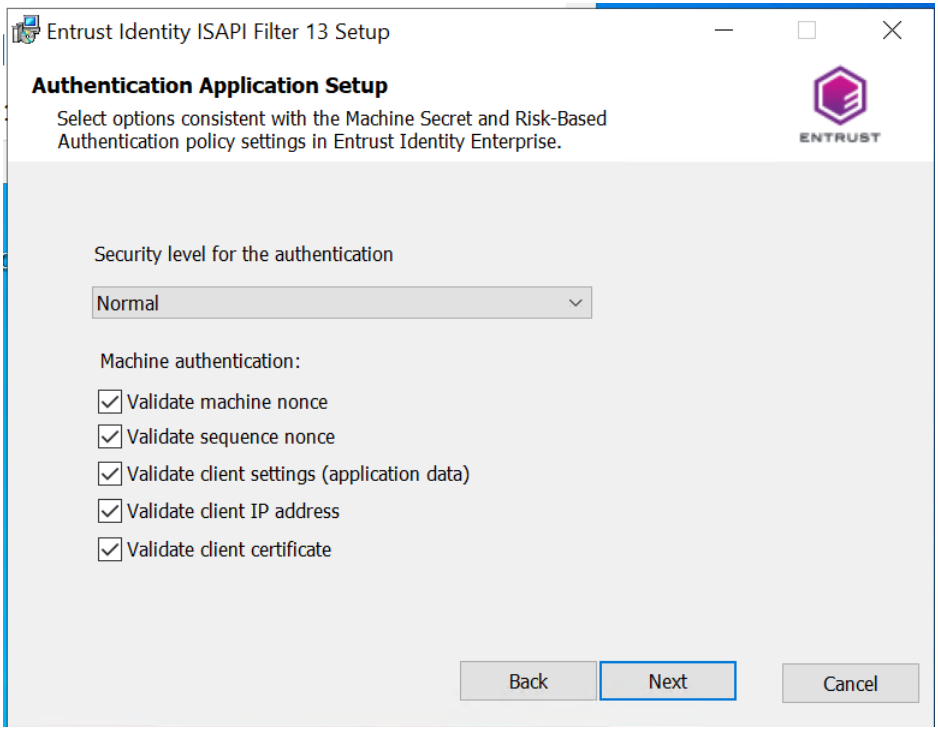
See “Risk-based authentication (RBA)” on page 24 and “Configuring risk-based authentication” on page 185, for more information about risk-based authentication.

c Enter the authentication level corresponding to the authentication type in the **Authentication Level** field.

Note:
The default authentication level is `1`, but you can change this. You must assign authentication types to all the levels you selected in the **Auth Level** field. See also:
- “Second-factor authentication methods” on page 23
- “Policy-based authentication” on page 28
- “Authentication levels” on page 27
- “Step-up authentication” on page 27

d Click **Next**.

22 If you did not select **Risk-based Authentication**, proceed to Step 24. If you selected **Risk-based Authentication**, the **Authentication Application Setup** page for security levels appears.



a Select the **Enhanced** or **Normal** level from the **Security level for the authentication** drop-down list (default **Normal**). The exact meanings of **Enhanced** and **Normal** are defined in your Entrust Identity Enterprise policy settings for Risk-Based Authentication (RBA). They are not defined in this solution.

b Select the appropriate **Machine authentication** check box if you want to use any of the following machine authentication features:

– **Validate machine nonce**

– **Validate sequence nonce**

This option cannot be selected unless you have selected **Validate machine nonce** or **Validate client settings (application data)**.

– **Validate client settings (application data)**

– **Validate client IP address**

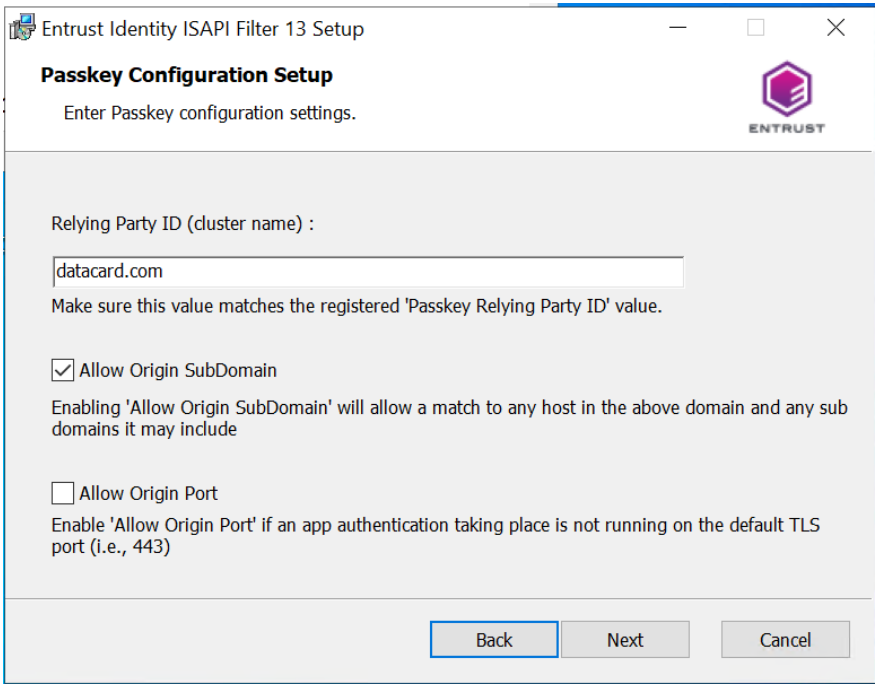
– **Validate client certificate**

This option requires a valid certificate. When a user logs in and successfully authenticates, Entrust Identity Enterprise uses the certificate to authenticate the user in subsequent logins.

By default they are all selected. Your selections here must match your Entrust Identity Enterprise policy settings for machine secrets.

Your RBA settings apply to all your authentication levels. If you want to define a different RBA policy for each level, then you can modify these settings after installation. See “Post-installation configuration” on page 125.

23 Click **Next**. If you selected Policy-based/Passkey authenticator in the previous step, The **Passkey Configuration Setup** page appears.



a In the **Relying Party ID** (cluster name) field, enter the Relying Party ID that users have used to register the Passkey. It can be a qualified domain name (FQDN) or the domain itself. The following provides an example of valid Relying Party ID values:

`<cluster name>.<subdomain>.<domain.com>.`

`<subdomain>.<domain.com>`

`<domain.com>`

b Set for the following, based on your required configuration:

– Enabling passkey authentication to Entrust Identity Self-Service Module.

– Enabling the self-administration action that allows a passkey to be registered.

c Select **Allow Origin Subdomain** to require the origin returned by the passkey to match the allowed origin associated with the Entrust Identity Enterprise relying party.

If the allowed origin references a host name, then successful passkey registration and authentication can only take place using apps running on that host.

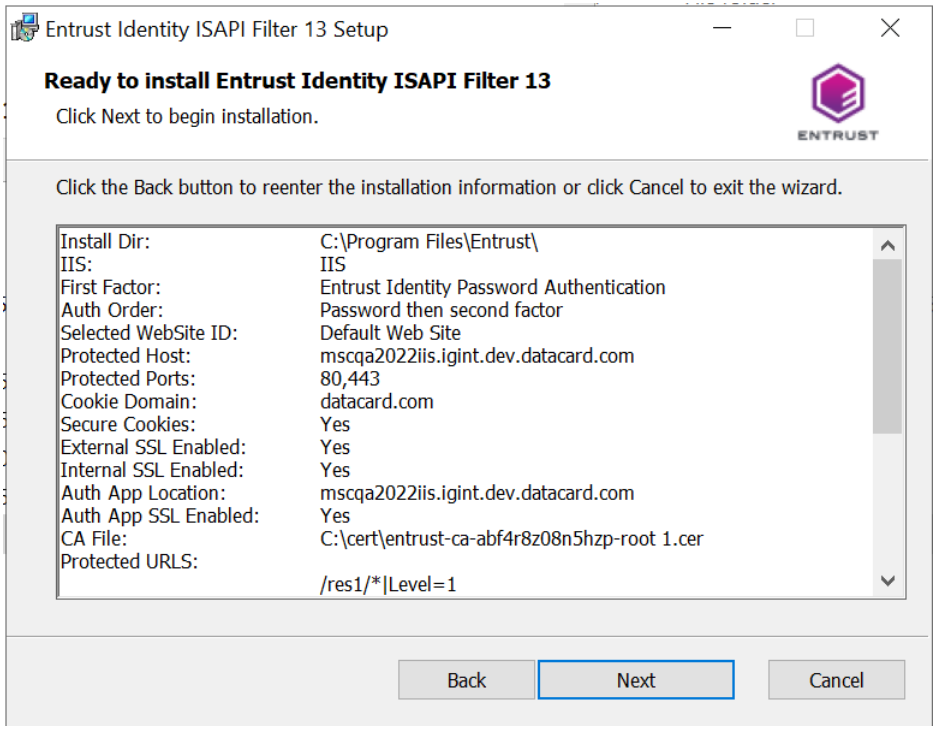
Note:
Entrust recommends that passkey registration takes place from any host where Entrust Identity Self-Service Module is installed, and that passkey authentication takes from any host that is running an app that requires Entrust Identity Enterprise to access protected resources. This setup requires the allowed origin to reference a domain (for example, `mycompany.com`) and the property must be set to **`true`** (the default is `false`) to allow a match to any host in that domain and any subdomains it may include.

d Select **Allow Origin Port** if the app users will use for passkey registration or authentication is not running on the default TLS port (for example, 443). The default is **`true`**.

Note:
You cannot associate a port number with the relying party ID (see https://www.w3.org/TR/webauthn-2/#rp-id), if the client origin returned by a passkey includes a port number. As a result, it is necessary to relax the origin matching rule so that any port is allowed.

e By default, Entrust Identity Self-Service Module is configured to run on port 8445. If you use a reverse proxy to expose your Entrust Identity Self-Service Module instances on port 443, and all applications that require Entrust Identity Enterprise passkey authentication are also accessible through port 443, set this property to **`false`**.

24 The **Ready to Install the Application** page appears.



Check all the settings displayed. To correct any errors, click **Back** to return to a previous screen and change the information.

25 To start the installation, click **Next**.

When the installation completes successfully, the final screen appears.

26 Click **Finish**.

You have now successfully installed the Entrust ISAPI Filter on an IIS server.

27 Start (or restart) the World Wide Web Publishing service.

Configuring the ISAPI Filter for SharePoint

If you are using the ISAPI Filter to protect SharePoint, you need to modify the ISAPI Filter configuration files to take advantage of the SharePoint integration features.

Topics in this section:

If you are using the ISAPI Filter with SharePoint, you must also complete the following post-installation task:

Changing the session state

If you are running Windows Server and you installed the Entrust Identity Enterprise Authentication Web Application component on the Web site where SharePoint is installed, you must add the session state module to the IdentityGuardAuth application.

If you are installing the authentication application on a SharePoint site on Windows, adding the session state module is not required since it is already added by default.

To change the session state

1. Open a Windows command prompt.
2. Navigate to the following directory:

   ```
   C:\Windows\System32\inetsrv
   ```

3. Enter the following command:

   ```
   appcmd add module /name:Session
   /type:System.Web.SessionState.SessionStateModule
   /preCondition:managedHandler
   /app.name:"<SharePoint-site>/IdentityGuardAuth"
   ```

   Where **<SharePoint-site>** is the common Web site where SharePoint and the IdentityGuardAuth application are running.

   For example:

```
appcmd add module /name:Session
/type:System.Web.SessionState.SessionStateModule
/preCondition:managedHandler
/app.name:"sharepoint.example.com/IdentityGuardAuth"
```

Configuring protected and unprotected URLs for SharePoint

When configuring the protected and unprotected URLs for SharePoint, you must edit the `IdentityGuardFilterConfiguration.xml` file to add the configuration settings required.

For more information about configuring protected and unprotected URLs, see "Configuring a protected host" on page 223.

To protect a SharePoint site

1. Open the `IdentityGuardFilterConfiguration.xml` file in a text editor.
2. Add the SharePoint site to the file, using the following example:

```
<ProtectedURLs authlevel="1" firstfactorid="iwa">
    <URL authlevel="1">/sites/*</URL>
    <URL
localaccessonly="true">/IdentityGuardAuth/AuthValidationService.as
mx</URL>
    <URL localaccessonly="true">/_vti_bin/sitedata.asmx</URL>
</ProtectedURLs>
<UnprotectedURLs>
    <URL>/_layouts/*</URL>
    <URL>/ScriptRecource.axd*</URL>
    <URL>/WebResource.axd*</URL>
</UnprotectedURLs>
```

   Replace **/sites/*** in the above example with the URL of the SharePoint site you want to protect. You can add additional URLs by adding additional `<URL>` elements.
3. Save and close the file.

Configuring the logoff URL

A logoff URL is used when you configure ISAPI Filter so that ISAPI Filter terminates the Entrust Identity Enterprise authentication session when the first-factor authentication session ends.

The URLs you configure here are specific to the environment in which the filter is deployed and the applications being protected. You can configure the logoff URL

to be any page or URL in your application that indicates the first-factor authentication session is over; for example, a page stating the user is logged off, a login page, or a logoff link.

Upon detection of the logoff URL, the ISAPI Filter terminates the Entrust Identity Enterprise authentication session.

When there is no logoff URL specified, the Entrust Identity Enterprise session continues until the user closes the browser window, or the session idle time-out period elapses. Configuring a logoff URL ensures that users are shown a second-factor challenge by Entrust Identity Enterprise the next time they log in to your application.

Using the redirect attribute

The optional `LogoffURLs redirect` attribute specifies where the user is directed after logging off. The ISAPI Filter allows the request for the logoff URL through to whatever page your application normally displays. You can override this behavior by providing a redirect attribute. Upon detection of the logoff URL with a redirect attribute, the ISAPI Filter terminates the Entrust Identity Enterprise authentication session and redirects the user to the specified page.

The Entrust Identity Enterprise Authentication Web application includes a default logoff page, `IdentityGuardLogoff.aspx`., that you can use as the redirect target. (See “Customizing user interface strings, including error messages” on page 267 for more information about customizing the appearance of the logoff page.) You can also provide your own page.

Consider using the redirect attribute if:

- You are using Generic Forms Authentication.
- You have a log off link in your application, but there is no associated log off page, or you want to display a custom logoff page.

The following rules govern the use of the redirect attribute. Consider these rules when defining logoff URLs with redirect.

- The `redirect` attribute applies to all URLs within the `LogoffURLs` element.
- If the `redirect` attribute is empty or missing, then the filter does not perform a redirect, and allows the original request to be processed.
- The redirect attribute can also be specified at the `URL` element attribute. If specified here, it overrides the global `LogoffURLs redirect` attribute.

If the redirect attribute is missing on the `URL` element, the default value from `LogoffURLs` is used. If there is no redirect value at the `LogoffURLs`, the request is completed without a redirect.

- When specifying the `redirect` attribute, use only absolute URLs to ensure that users see the logoff page you intend, regardless of their current HTTP request path.

To specify a logoff URL with redirect

1. Open the `IdentityGuardFilterConfiguration.xml` file in a text editor.
2. Find the `ProtectedHost` element.

```
<ProtectedHost host="" port="80,443">
  ...
</ProtectedHost>
```

3. After the close of the `UnprotectedURLs` list, define the logoff URLs with the redirect attribute as shown in the following example. This example uses the default logoff page provided with the Authentication Web application. You can use your own URL instead.

```
<ProtectedHost host="" port="80,443">
    ...
    <ProtectedURLs authlevel="1" firstfactorid="owa">
        <URL/protected</URL>
    </ProtectedURLs>
    <UnprotectedURLs/>
<LogoffURLs
redirect="https://www.example.com/IdentityGuardAuth/IdentityGuardLogoff.aspx">
    <URL>/customapp/logoff.aspx</URL>
    <URL>/signout.aspx</URL>
</LogoffURLs>
</ProtectedHost>
```

4. If desired, you can define the `redirect` attribute at the URL level to override the global redirect definition. For example:

```
<LogoffURLs
redirect="https://www.example.com/IdentityGuardAuth/IdentityGuardLogoff.aspx">
    <URL
redirect="https://www.example.com/yourlogoffpage.aspx">/sites/sampleappln/_layouts/SignOut.aspx</URL>
</LogoffURLs>
```

5 Save and close the file.

Using logoff URLs for various ISAPI Filter configurations

Logoff URLs are configured differently for each first-factor type and system configuration.

Table 4: Using logoff URLs for various ISAPI Filter configurations

ISAPI Filter configuration	Using logoff URLs
Outlook Web Access	Logoff URLs are not used in this mode.
Integrated Windows Authentication (IWA)	In this mode, ISAPI Filter does not use any logoff URLs by default. You may choose to configure logoff URLs with or without use of a `redirect` attribute. In this case, first-factor authentication (IWA) is not logged off. The logoff URLs only log off second-factor authentication.

Generic forms-based authentication	In this mode, ISAPI Filter does not use any logoff URLs by default. You can configure logoff URLs based on the custom applications you are protecting. It is recommended that you define a `redirect` attribute to redirect users to a logoff page such as the `IdentityGuardLogoff.aspx` provided.
	This example assumes there is one application in the root directory, and another under `/customapp`.
	`<LogoffURLs redirect="https://www.company.com/IdentityGuardAuth/IdentityGuardLogoff.aspx">`
	`<URL>/customapp/logout.aspx</URL>`
	`<URL>/signout.aspx</URL>`
	`</LogoffURLs>`

SharePoint in the IIS-only configuration
(* This is a special case of IWA.)
	In this mode, ISAPI Filter does not use any logoff URLs by default, but if you use ISAPI Filter with a SharePoint configuration, you can configure a logoff URL for the SharePoint `SignOut.aspx` page. For example:
	`http://testsite.anycorp.com:27689/_layouts/SignOut.aspx`
	For example:
	`<LogoffURLs>`
	`<URL>/sites/sampleappln/_layouts/SignOut.aspx</URL>`
	`</LogoffURLs>`

Configuring the filter to use persistent cookies with SharePoint

To fully integrate the ISAPI Filter with SharePoint, you need to configure the filter to use persistent cookies. With persistent cookies, your SharePoint users can access, modify and save Microsoft Office seamlessly. By default, the filter uses session cookies. Without persistent cookies present, documents retrieved from SharePoint and edited cannot be saved directly back to SharePoint.

To set persistent cookies, you must modify two configuration files. In an IIS-only installation, they will be in the same folder on the IIS server.

Ensure that the cookie settings for `lifetime` and `domain` (if used) have the same values in both configuration files.

To set cookie persistence in the authentication application

1. Open the `IdentityGuardAuthAppConfiguration.xml` file in a text editor.
2. Find the `Headers` element near the bottom of the file.

   ```
   <Headers>
   ```

3. Find the `Cookies` element just below `Headers` for the domain used for SharePoint access. It looks something like this:

   ```
   <Cookies domain="exch2010.com" secure="true">
   ```

4. Change the `Cookies` element to add the new `lifetime` attribute.

   ```
   <Cookies domain="exch2010.com" secure="true" lifetime=”30”>
   ```

   where `lifetime` equals the number of minutes the cookie should persist.
5. Save and close the file.

For information on `domain` and other attributes, see “Configuring authentication cookies” on page 235.

To set cookie persistence in the ISAPI Filter

1. Open the `IdentityGuardFilterConfiguration.xml` file in a text editor.
2. Find the `ProtectedHost` element.
3. Find the `Cookies` element just below `ProtectedHost`. By default, it looks something like this:

   ```
   <Cookies domain="exch2010.com">
       <UserID>IGUser</UserID>
       <SessionID>IGSession</SessionID>
       <SessionKey>IGSessionKey</SessionKey>
       <AuthValidationServiceUID>IGAuthValidationServiceUID</AuthValid
       ationServiceUID>
   </Cookies>
   ```

4 Change the `Cookies` element to add two new attributes and define the new cookie, IGPersistent.

```
<Cookies domain="exch2010.com" lifetime=”30”
cookiesynclifetime=”10”>
    <UserID>IGUser</UserID>
    <SessionID>IGSession</SessionID>
    <SessionKey>IGSessionKey</SessionKey>
    <AuthValidationServiceUID>IGAuthValidationServiceUID</AuthValid
    ationServiceUID>
    <IGPersistent>IGPersist</IGPersistent>
</Cookies>
```

where

- `lifetime` equals the number of minutes the cookie should persist
- `cookiesynclifetime` gives the time in minutes that will elapsed before the `lifetime` attribute is reset to the expressed value
- `domain` sets the domain

5 Save and close the file.

For information on `domain` and other attributes, see “Configuring authentication cookies” on page 235.

Using the ISAPI Filter with SharePoint on a non-default port

If you install the ISAPI Filter with a SharePoint site located on a port other than a default port (80 or 443), you must modify the `IdentityGuardFilterConfiguration.xml` file after installation to add the port information to the URLs.

After installing ISAPI Filter, open `IdentityGuardFilterConfiguration.xml` for editing, and make the following changes.

To use the ISAPI Filter with SharePoint on a non-default port (IIS)

1 Open the `IdentityGuardFilterConfiguration.xml` file in a text editor.

2 Find the `IdentityGuardLogin` setting just below the `ProtectedHost` element.

3 Modify the URL attribute to indicate the non-default port. For example, if you are using HTTP on port 81 instead of port 80, change:

`http://host.domain.com/IdentityGuardAuth`

to

`http://host.domain.com:81/IdentityGuardAuth`

4 Move down to the `AuthValidationService` setting in the XML file.

5 Modify the URL attribute to indicate the non-default port. For example, if you are using HTTP on port 81 instead of port 80, change:

```
http://host.domain.com/IdentityGuardAuth/AuthValidationService.asmx
```

to

```
http://host.domain.com:81/IdentityGuardAuth/AuthValidationService.asmx
```

6 Save and close the file.

Restarting the Web service after configuring the ISAPI Filter

After editing the ISAPI Filter configuration files, you must restart the applicable Web service for the changes to take effect.

See "Restarting services after changing configuration files" on page 141 for details about restarting the applicable services.

Configuring IIS servers

This section describes various configurations that you can set up on your IIS server.

This section contains the following topics:

Setting up Basic Authentication

You can use the Integrated Windows Authentication (also called Windows authentication) or Basic Authentication provided by IIS as the first-factor authentication, and Entrust Identity Enterprise as the second-factor authenticator. If you want to use Basic Authentication, complete the following steps.

To set up Basic Authentication

1. Basic Authentication was installed when you added the Application Server role. See "IIS 10: Adding the required role services on IIS" on page 42.
2. Set up the ISAPI Filter to protect the Web application that needs protection.
3. Enable Basic Authentication for the IdentityGuardAuth Web application on IIS:
   a. In IIS Manager, select **IdentityGuardAuth**.
   b. In the middle pane, double-click **Authentication**.
   c. Select **Anonymous Authentication** and then click **Disable**.
   d. Select **Basic Authentication** and then click **Enable**.
   e. Select **Windows Authentication** and then click **Disable**.
4. Ensure that `AuthValidationService.asmx` within the IdentityGuardAuth Web application has anonymous access configured:
   a. In IIS Manager, right-click **IdentityGuardAuth** and select **Switch to Content View**).
   b. Select `AuthValidationService.asmx` and then under Actions, click **Switch to Features View** (or right-click `AuthValidationService.asmx` and select **Switch to Features View**).
   c. Double-click **Authentication**.
   d. Select **Anonymous Authentication** and then click **Enable**.
5. Restart IIS.

Implementing your first-factor login in generic forms-based authentication

If you set up the ISAPI Filter solution to use generic forms-based authentication, you must modify the code on the default form to implement your own first-factor login.

Generic forms-based login is implemented in `ApplicationLogin.aspx.cs`. It does not actually perform first-factor user ID and password authentication. It is the default template used to get the user ID from the user without performing first-factor authentication. It performs only second-factor authentication.

You can customize the application to use your own first-factor login. See "To modify the provided sample to implement your own first-factor authentication", below.

When you implement your own first-factor login, you should also specify logoff URLs. Using logoff URLs allows you to ensure that when users end their first-factor login sessions, their Entrust Identity Enterprise second-factor session is automatically terminated. See "Configuring the logoff URL" on page 100 to implement logoff URLs.

The exact steps depend on which application you are integrating with.

To modify the provided sample to implement your own first-factor authentication

1. Back up the `IdentityGuardAuth` Web folder before making these changes. By default this folder is in located at:

   `C:\Program Files\Entrust\Identity\WinIS\webapp\IdentityGuardAuth`

2. In a text editor, open the `ApplicationLogin.aspx` file.
3. Integrate your application's first-factor authentication form into the sample file. You could add a password field, for example.

   Modify the code in `ApplicationLogin.aspx.cs` to add first-factor authentication to your application. This may include, for example:

   - Connecting to a database and verifying a user's credentials.
   - Creating a proxy for your application login form.

   The file contains detailed comments to help you make your modifications.

Passkey/FIDO2 registration and authentication

The following sections describe how to register a Passkey/FIDO2 authenticator and then use Passkey/FIDO2 for authentication.

Topics in this section:

Passkey/FIDO2 registration and authentication with IDaaS

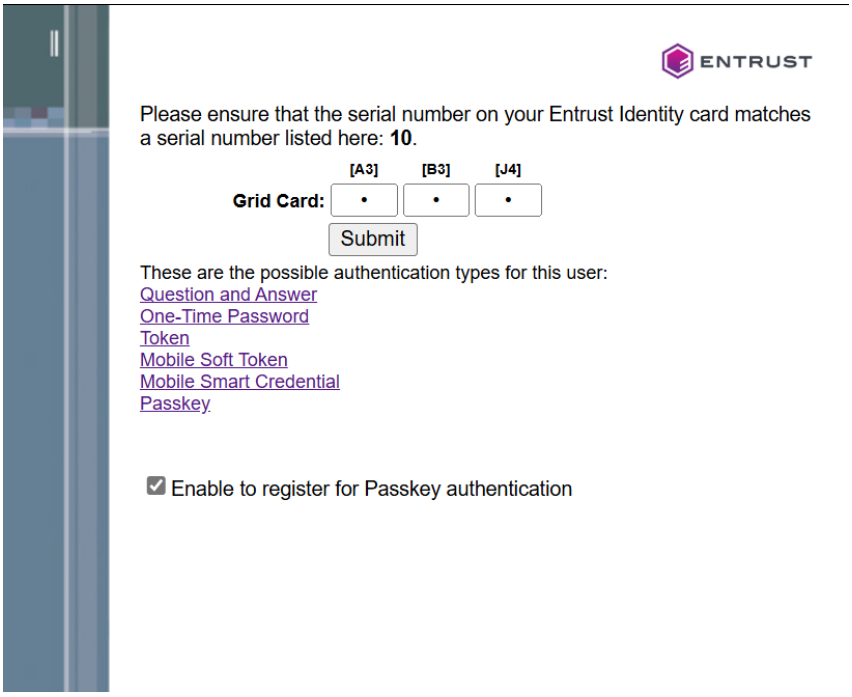
Topics in this section:

Register a Passkey/FIDO2 token with IDaaS

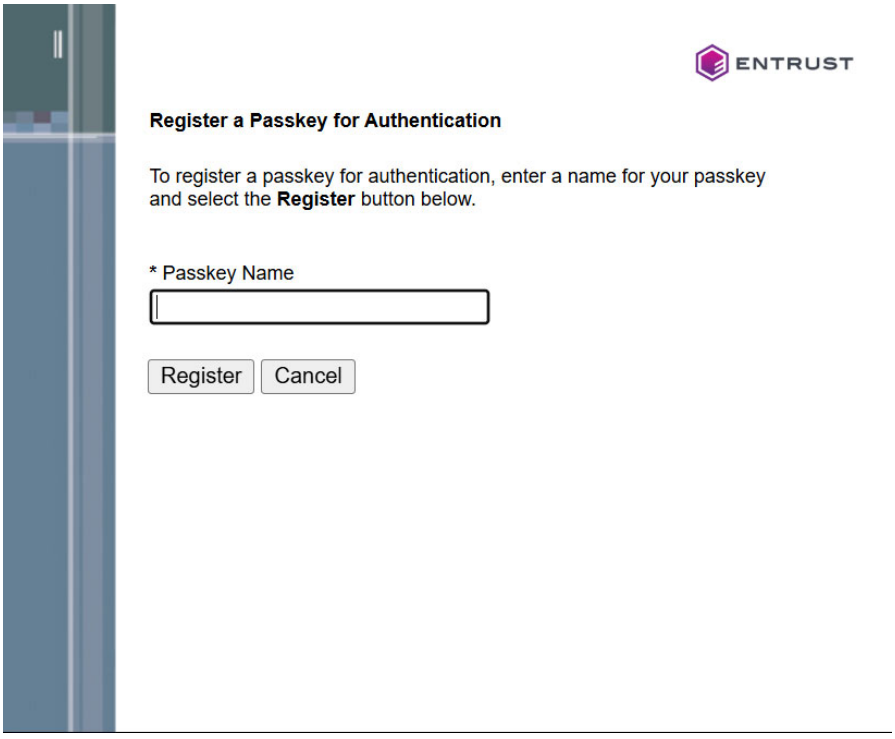
To register a Passkey/FIDO2 token

1. Before you begin, locate your Passkey/FIDO2 token that is compliant with the FIDO2/WebAuthn specification.
2. Access the ISAPI Filter resource and enter your username and password in first-factor page.

3 In second-factor page, answer the challenge and select the **Enable to register a Passkey for authentication** checkbox.



4 Click **Submit**. You are prompted to enter the Passkey Name.



5 Enter the **Passkey Name** and then click **Register**.

6 Follow the screen prompts to complete Passkey/FIDO2 registration.

Authenticate using a Passkey/FIDO2 token with IDaaS

To authenticate using Passkey/FIDO2 token

1 Before you begin, locate your Passkey/FIDO2 token that is compliant with the FIDO2/WebAuthn specification.

2 Clear all browser cookies.

3 Access the ISAPI Filter resource and enter your username and password.

4 In the **Challenge Authentication** page, click **Passkey**.

5 Follow the screen prompts to complete Passkey/FIDO2 authentication.

6 After successful authentication, the user is logged into the protected resource.

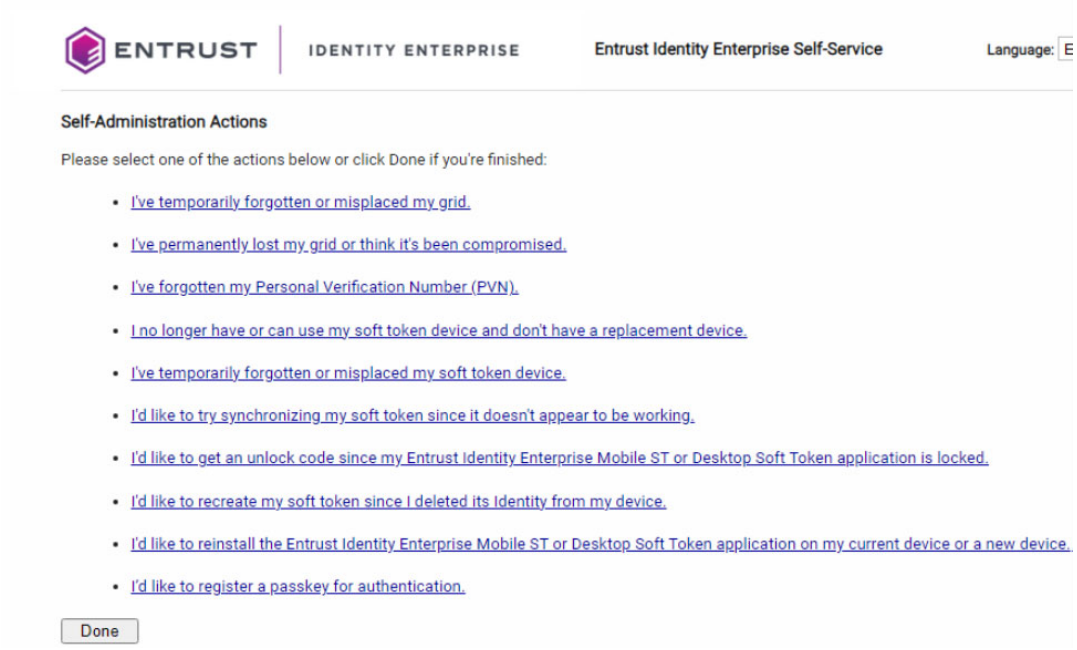
Passkey/FIDO2 registration and authenticaiton with Entrust Identity Self-Service Module

Topics in this section:

Registering Passkey/FIDO2 token with IDE

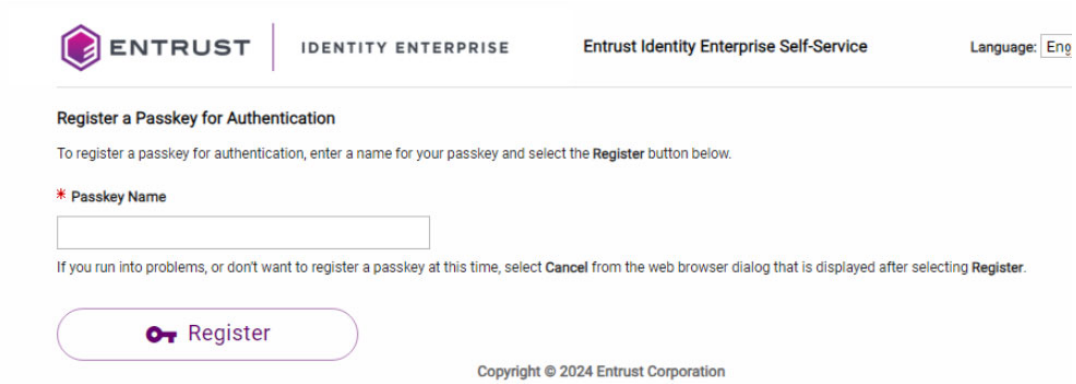
1 Before you begin, locate your Passkey/FIDO2 token that is compliant with the FIDO2/WebAuthn specification.

2 Log in to the Entrust Identity Enterprise Self-Service Module portal. The **Self-Administration Actions** page appears.



3 Click **I'd like to register a passkey for authentication**.

4 Click **Yes** on the confirmation prompt. The **Register a Passkey for Authentication** page appears.



5 Enter a **Passkey Name**.

6 Click **Register**.

7 Follow the screen prompts to complete the Passkey/FIDO2 token registration.

Authenticate using Passkey/FIDO2 token with IDE

To authenticate using Passkey/FIDO2 token

1 Before you begin, locate your Passkey/FIDO2 token that is compliant with the FIDO2/WebAuthn specification.

2 Clear all browser cookies.

3 Access the ISAPI Filter resource and enter your username and password.

4 In the **Challenge Authentication** page, click Passkey.

5 Follow the screen prompts to complete Passkey/FIDO2 authentication.

6 After successful authentication, the user is logged into the protected resource.

4

Testing the solution

After you install the Entrust ISAPI Filter, it is recommended that you test it before performing any customizations.

Topics in this chapter:

Testing your solution

Test your solution by trying to access a protected URL from a browser. Try accessing the protected URL from different parts of your network, such as:

- on the same server as the filter
- on the same subnet as the ISAPI solution

Testing on IIS with Outlook Web Access

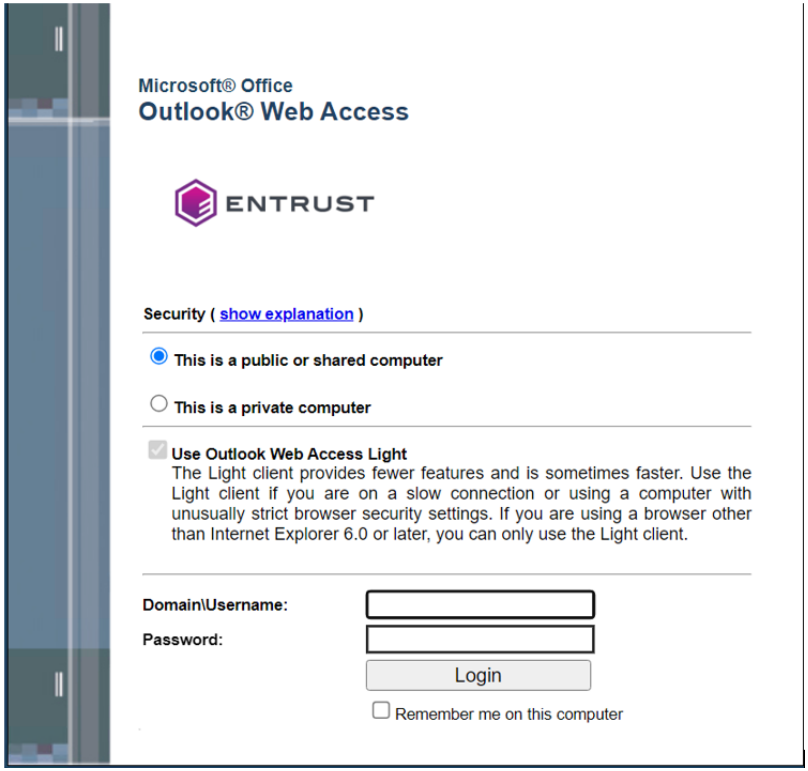
Use the following procedure to test your ISAPI Filter solution on IIS with OWA.

To test your solution on IIS with OWA

1 Open a browser window and enter one of your protected URLs; for example:

`https://host.company.com/owa`

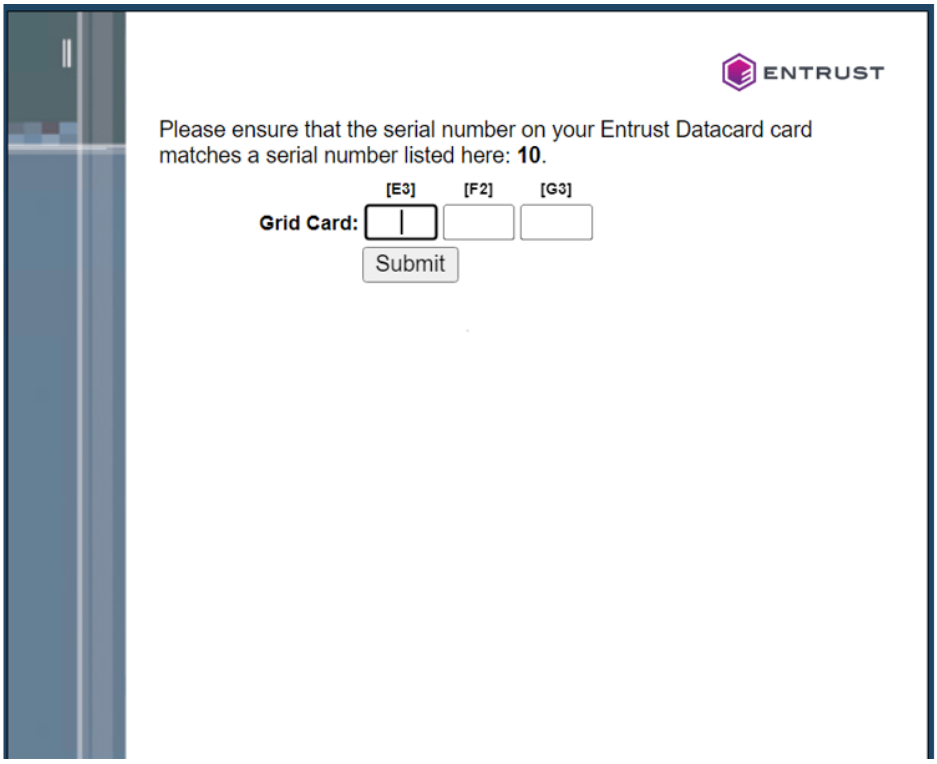
The Outlook Web Access authentication page appears.



2 Enter your OWA user name and password.

3 Click **Login**.

The second-factor challenge appears in your browser.



4 Enter the correct response to the challenge. You are redirected to the user's OWA Inbox.

If you are unable to reach the protected URL, check the log files on the IIS server.

Also check your OWA or Exchange logs. If you do not find sufficient information, increase the log level to get more detailed information, and retry the test. For instructions on changing the logging level, see "Configure ISAPI Filter for Identity as a Service" on page 142.

Testing reverse authentication on IIS

The ISAPI Filter makes it possible to switch the authentication order for installations that use OWA, Entrust Identity Enterprise Password Authentication, Generic Forms Based Authentication, or Remote Desktop Web Access.

Instead of presenting the password to the user first followed by second-factor authentication, you can present second-factor authentication first followed by the password.

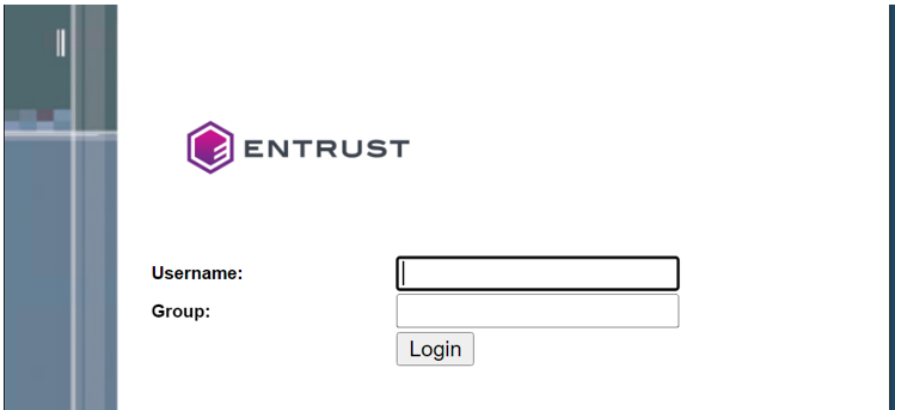
The following example shows how to test your ISAPI Filter on Entrust Identity Enterprise Password Authentication when authentication order is reversed. The sequence is identical for other authentication methods.

To test your solution on IIS when authentication order is reversed

1 Open a browser window and enter one of your protected URLs; for example:

`https://host.domain.com/test1`

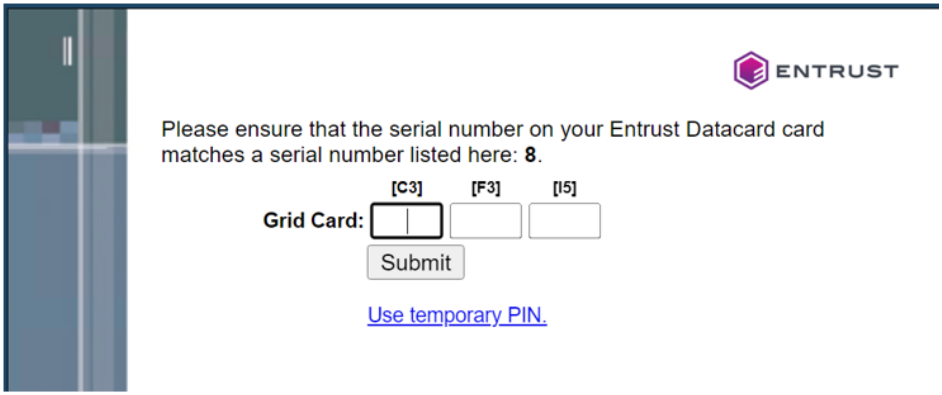
The authentication page appears showing just the user name and group prompt.



2 Enter your user name.

3 Click **Login**.

The second-factor challenge appears in your browser.



4 Enter the correct response to the challenge.

The password prompt appears.

5 Enter your password and click **Submit**.

You are redirected to the user's protected page.

If you are unable to reach the protected URL, check the log files on the IIS server. If you do not find sufficient information, increase the log level to get more detailed information, and retry the test. For instructions on changing the logging level, see "Configure ISAPI Filter for Identity as a Service" on page 142.

Testing on IIS with Integrated Windows Authentication

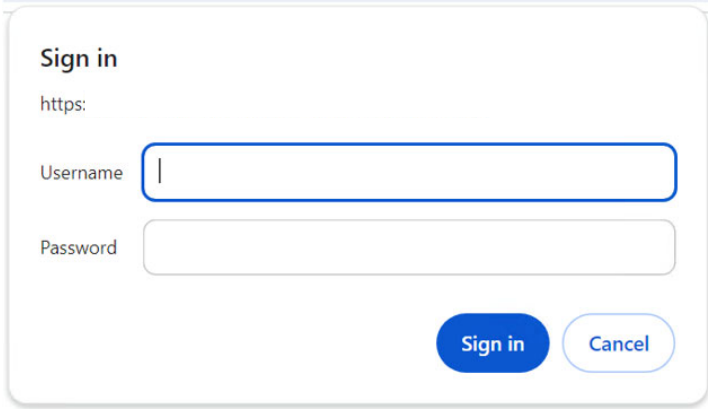
Use the following procedure to test your ISAPI Filter solution on IIS with Integrated Windows Authentication.

To test your solution on IIS with Integrated Windows Authentication

1 Open a browser window and enter one of your protected URLs; for example:

```
https://host.company.com/protected/test.htm
```

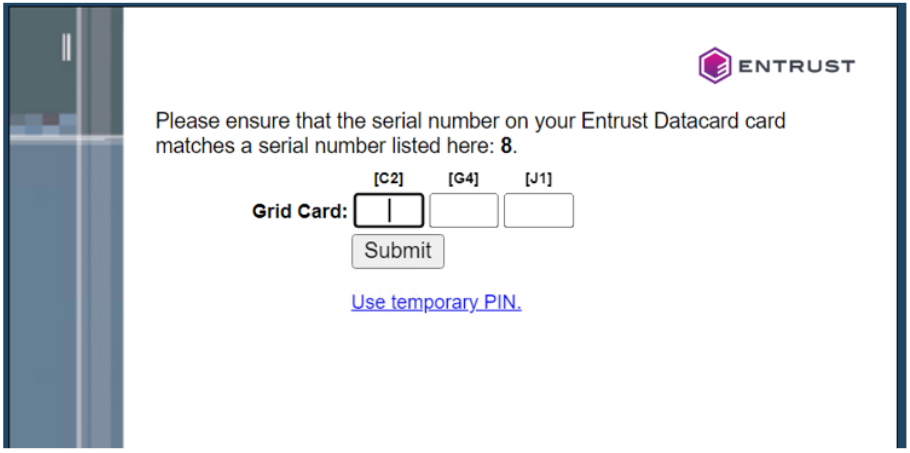
You may see the integrated Windows authentication dialog box as shown below. This dialog box may or may not appear, depending on your browser settings.



2 If you see this dialog box, enter your Windows password and click **OK**.

Note:
If you enabled risk-based authentication, you may temporarily see a redirection page (as shown below) before the second-factor authentication challenge. This page is related to machine authentication, and validates your machine settings. If a challenge is required, it automatically redirects your browser to the second-factor challenge page. You do not need to do anything.

The second-factor challenge appears in your browser.



3 Enter the correct response to the challenge. You are redirected to the protected URL you requested at the start of the test.

Sometimes, an `IdentityGuardUpdateFlash.aspx` page appears after the second-factor challenge response and before the protected URL. It displays **Please Wait** in red in the browser window. This page updates the machine and sequence nonce values that are stored in Flash. There is no need to interact with this page. It is followed by the protected URL. You only see it if the protected URL takes a long time to load.

If there is no second-factor challenge, the Flash update page appears immediately after the redirection page.

If you are unable to reach the protected URL, consult the log files in located at `C:\Program Files\Entrust\Daracard\WinIS\log`.

Also check your IIS logs or Windows Event Viewer. If you do not find sufficient information, increase the log level to get more detailed information, and retry the

test. For instructions on changing the logging level, see “Configure ISAPI Filter for Identity as a Service” on page 142. Consult your IIS Web logs.

Testing on IIS with Generic Forms Based Authentication

Use the following procedure to test your ISAPI Filter solution on IIS with Generic Forms Based Authentication.

To test your solution on IIS with Generic Forms Based Authentication

1. Open a browser window and enter one of your protected URLs; for example:

   `https://host.company.com/protected/test.htm`

   The Entrust authentication page appears.
2. Enter your Entrust Identity Enterprise **Username** and **Group** (optional).
3. Click **Login**.

   The second-factor challenge appears in your browser.
4. Enter the correct response to the challenge. You are redirected to the protected URL you requested at the start of the test.

If you are unable to reach the protected URL, consult the log files located at `C:\Program Files\Entrust\IdentityGuard\WinIS\log`

Also check your IIS logs or Windows Event Viewer. If you do not find sufficient information there, increase the log level to get more detailed information, and retry the test. For instructions on changing the logging level, see “Configure ISAPI Filter for Identity as a Service” on page 142.

Testing one-step authentication on IIS

The ISAPI Filter makes one step logins possible when the authentication method is Token RO.

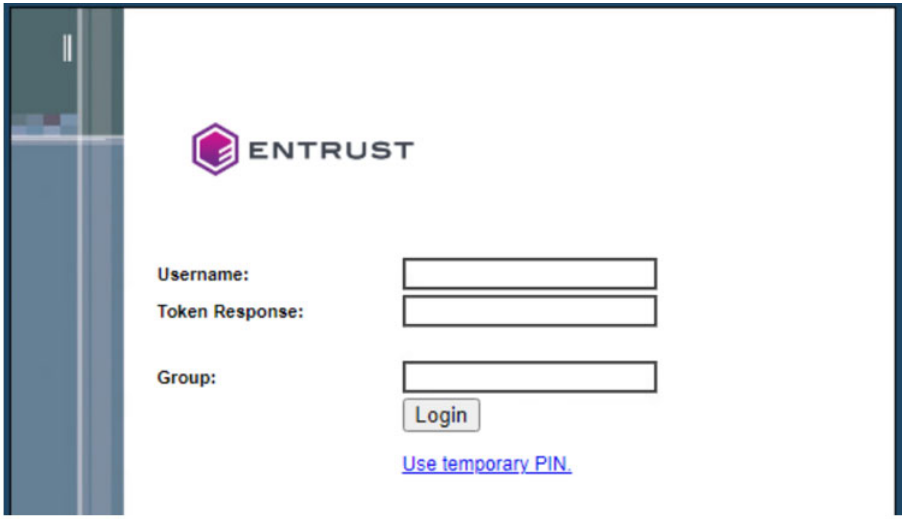
The following example shows a simple test of your ISAPI Filter f in a one-step scenario.

To test your solution on IIS with one-step login

1. Open a browser window and enter one of your protected URLs; for example:

   `https://host.company.com/protected/test.htm`

The authentication page appears .



2 Enter a user name.

3 Enter the token response.

4 Enter the password.

5 Click **Login**.

You are redirected to the resource page.

Testing PCI-DSS solution with generic forms-based authentication

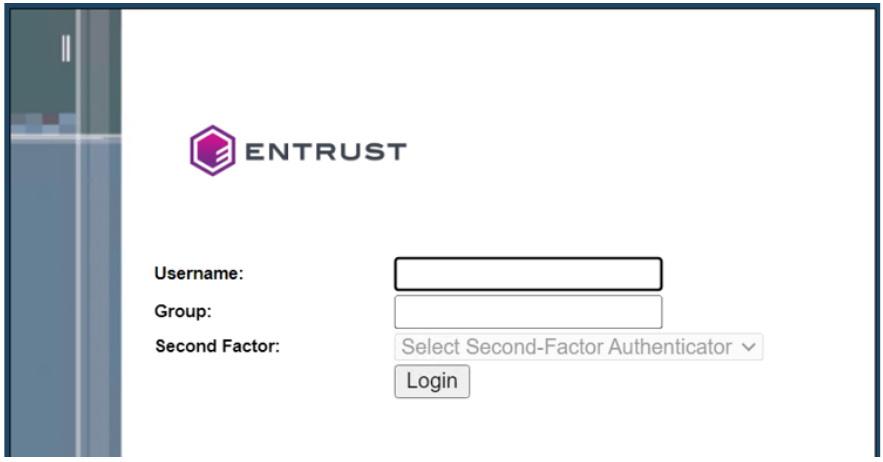
Test your solution by trying to access a protected URL from a browser.

To test your PCI-DSS solution with generic forms-based authentication

1 Open a browser window and enter one of your protected URLs; for example:

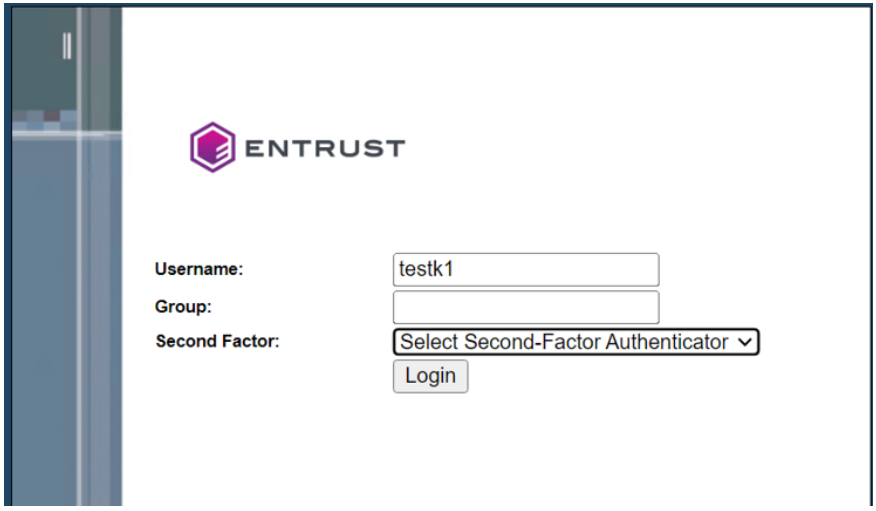
```
https://host.company.com/protected/test.htm
```

The Entrust authentication page appears.

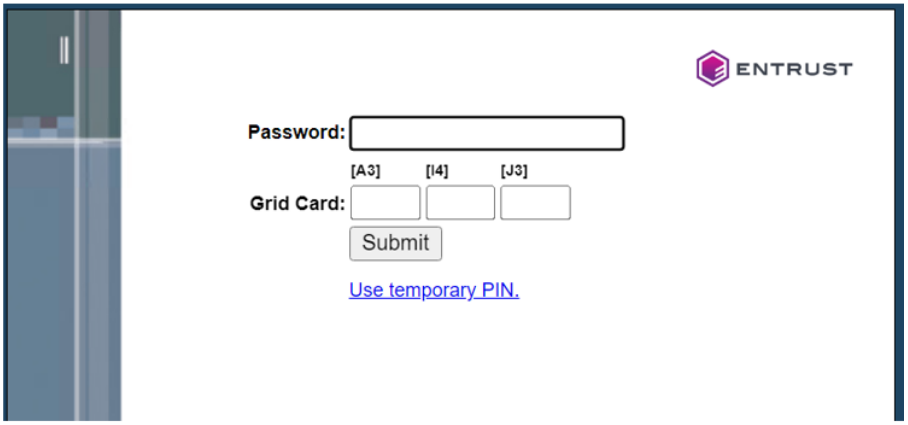


2 Enter your Entrust username and Group (optional).

3 Select the second factor authentication method from the drop-down list.



The second factor challenge page appears on the next screen.



4 Enter the user password, second factor challenge response, personal verification number (optional), and then click **Login**.

You are redirected to the protected URL that you requested at the start of the test.

5

Post-installation configuration

This section describes procedures that you may want to perform after you install the ISAPI Filter and authentication application. Topics in this chapter:

- “Modifying other configurations” on page 239

Locating configuration files

There are three configuration files for the ISAPI Filter solution, one for the filter component, one for global logging of the filter settings, and one for the authentication application.

The configuration file for:

- the filter component: `IdentityGuardFilterConfiguration.xml`
- the global filter logging: `IdentityGuardFilterLoggerConfig.xml`
- the authentication application: `IdentityGuardAuthAppConfiguration.xml`

In an IIS-only installation, these three files are located on the same computer.

The default location for these configuration files on the computer where the filter component is installed is

`C:\Program Files\Entrust\Identity\WinIS\config`

Note:
After making changes to the configuration files, you must restart the applicable Web or firewall service. See "Restarting services after changing configuration files" on page 141.

Configuring ISAPI Filter manually

To configure the ISAPI Filter manually, you open the required configuration file in a text editor and edit and then save the file with the required changes.

Configuring ISAPI Filter using the Configuration Console

The Entrust ISAPI Filter includes a Configuration Console that allows you to make post-installation changes to your integration using a simple tool. In order to use the configuration console, you must first install Java SE Development Kit (JDK 7 or 8).

You can use the configuration console to edit the following files:

- `IdentityGuardFilterLoggerConfig.xml`
- `IdentityGuardFilterConfiguration.xml`
- `IdentityGuardAuthAppConfiguration.xml`

To configure and open the configuration console

1 Install the Java SE Development kit (JDK 7 or 8) available at http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html.

2 Run **`ConfigConsole.jar`** to open the Configuration Console.

Editing the configuration files using the Configuration Console

After you configure the Configuration Console you can run it to edit the filter and AuthApp configuration files.

Topics in this section:

Opening a configuration file for editing

After you run the `ConfigConsole.jar`, the Entrust Identity Enterprise Configuration Console window opens to enable you to edit the configuration files.

To open a configuration file

1 On the Configuration Console menu bar, click **File>Browse**.

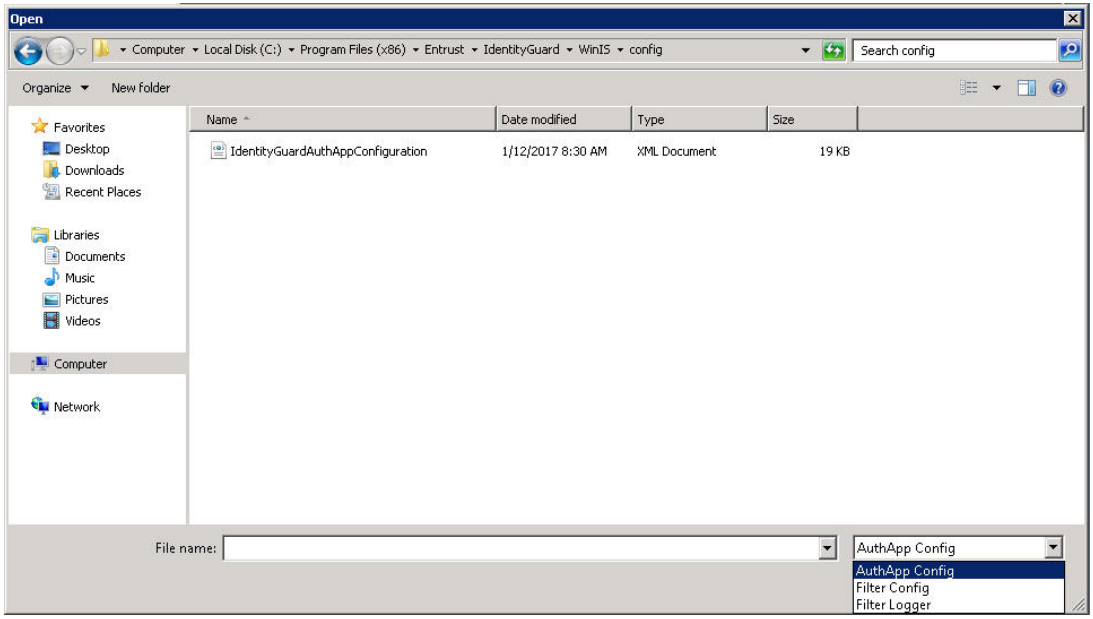
2 Browse to the ISAPI Filter config folder:

The default location is

`C:\Program Files\Entrust\Identity\WinIS\config`

3 Select the configuration file type that you want to edit from the drop-down list in the right-hand bottom corner of the window. For example, if you want to edit

the authentication application configuration file, select **AuthApp Config** from the drop-down list.

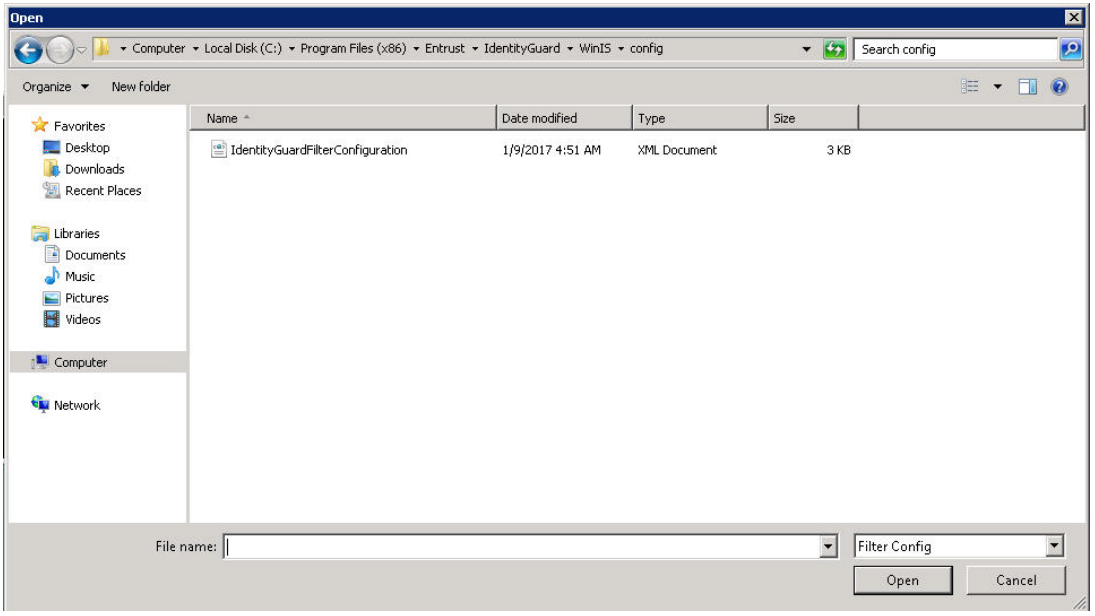


Editing the filter configuration file

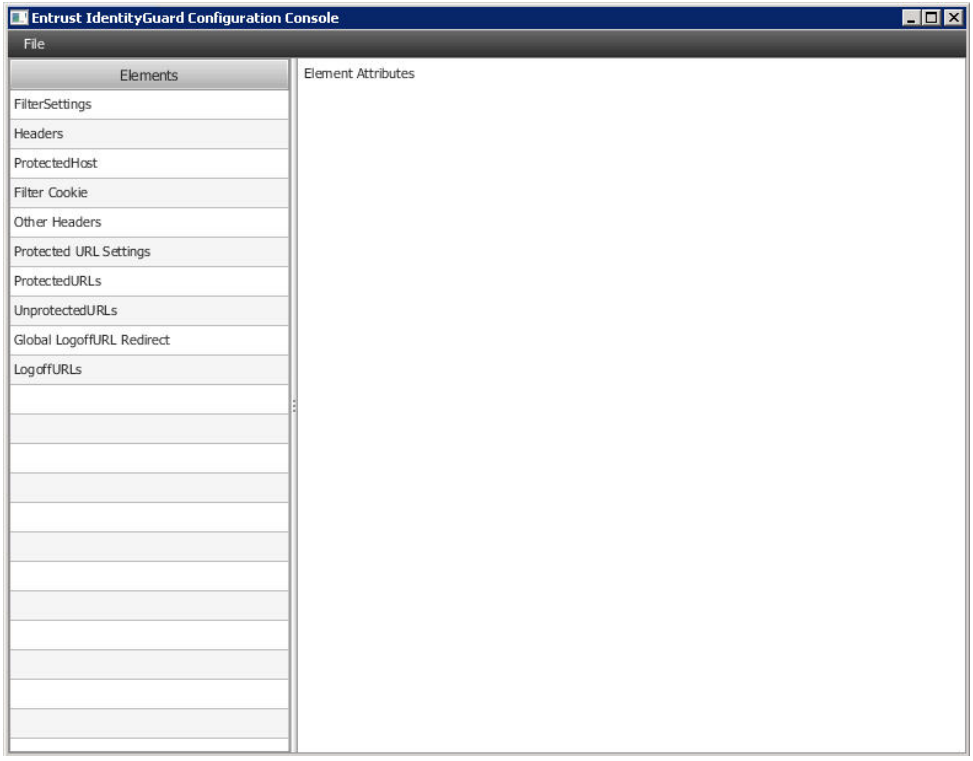
To edit the filter configuration file

1 Browse to ISAPI config folder and select the **Filter Config** configuration file type from the drop-down menu (see "Opening a configuration file for editing" on page 128).

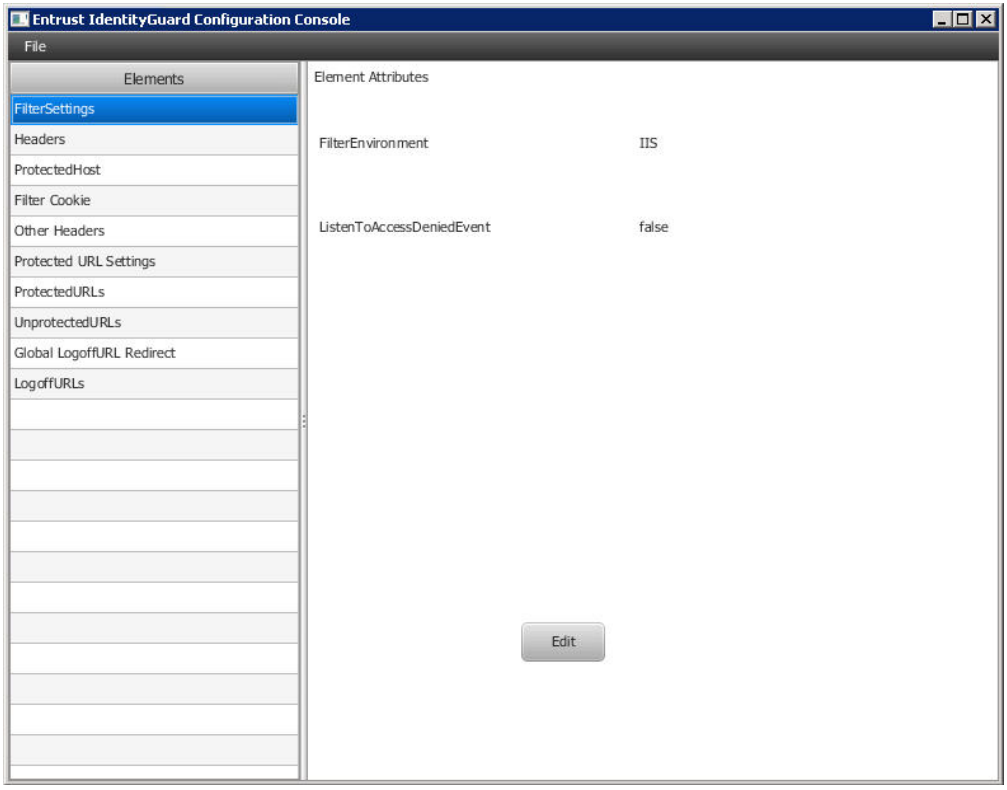
2 Select the file `IdentityGuardFilterConfiguration` and then double-click or click **Open** to open the file.



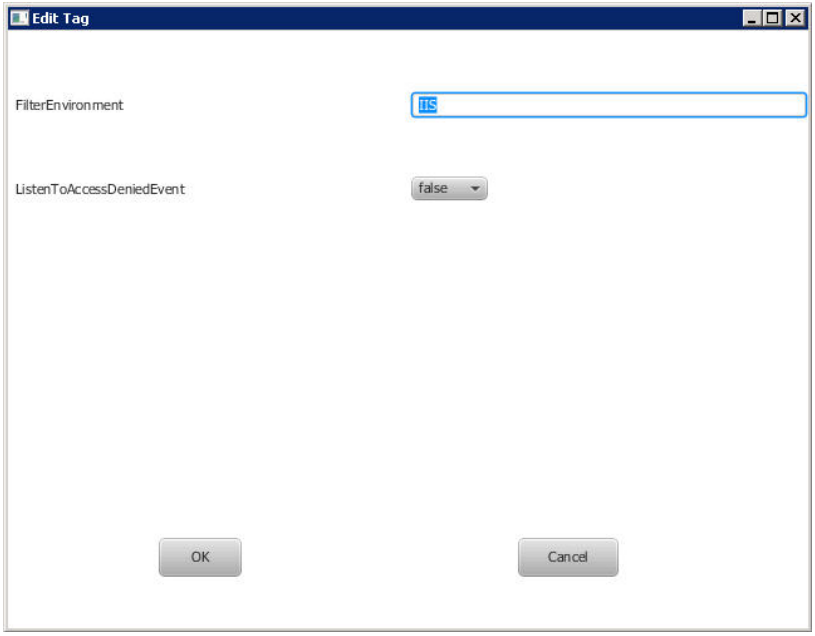
The Configuration Console opens the filter configuration file and lists the filter configuration file Elements available for editing.



3 To edit any of the elements, select the element in the **Elements** list, for example, **FilterSettings** as shown below, to display the **Element Attributes** page and then click **Edit**.



The **Edit Tag** page appears.



4 Make the required changes and then click **OK** to apply them.

5 To close the configuration file, select **File > Close** in the in Configuration Console.

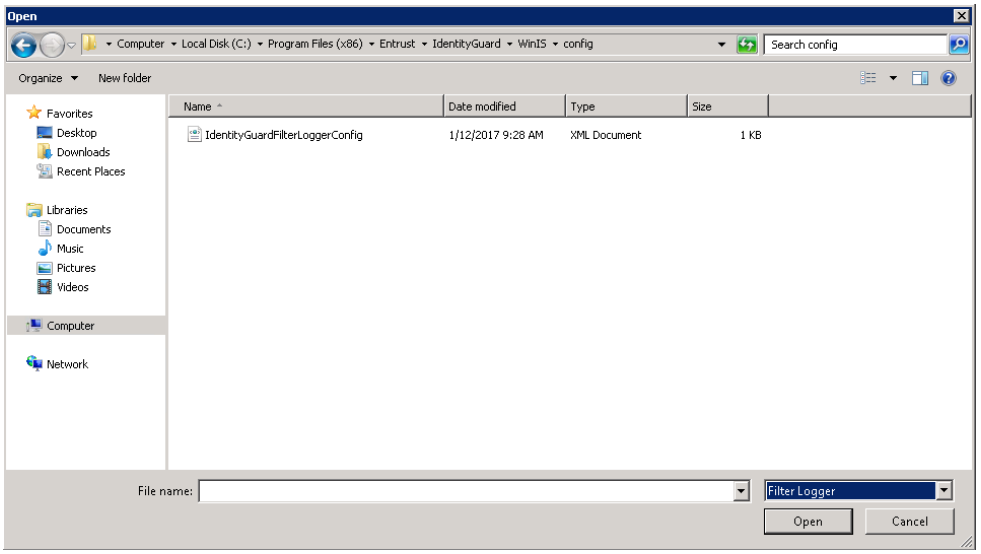
6 To close the Configuration Console, click **Close** on the title bar.

Editing the filter logger configuration file

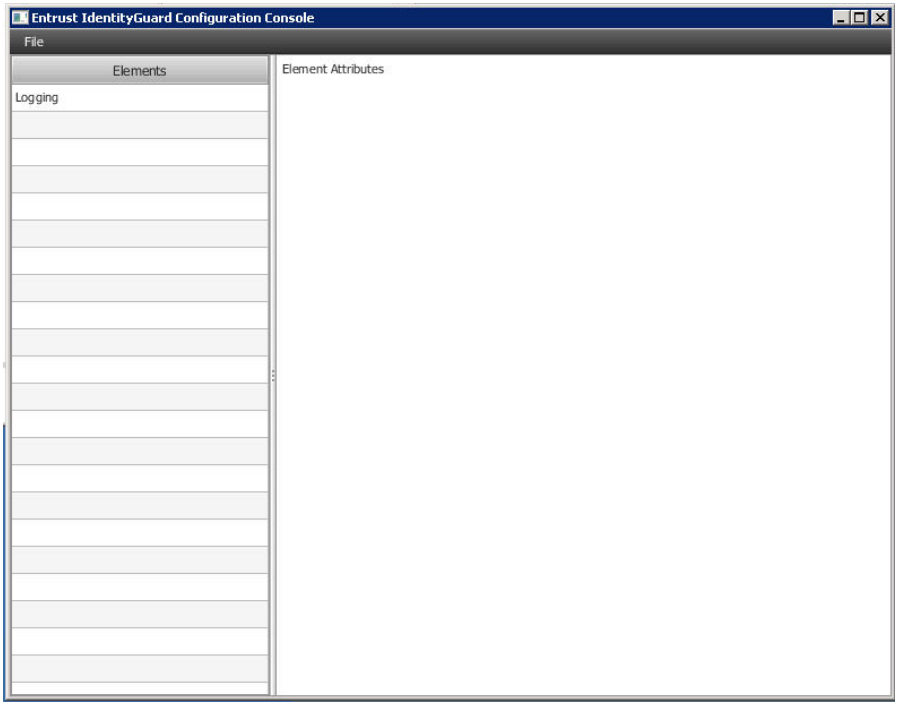
To edit the filter logger configuration file

1 Browse to ISAPI config folder and select the **Filter Logger** configuration file type from the drop-down menu (see "Opening a configuration file for editing" on page 128).

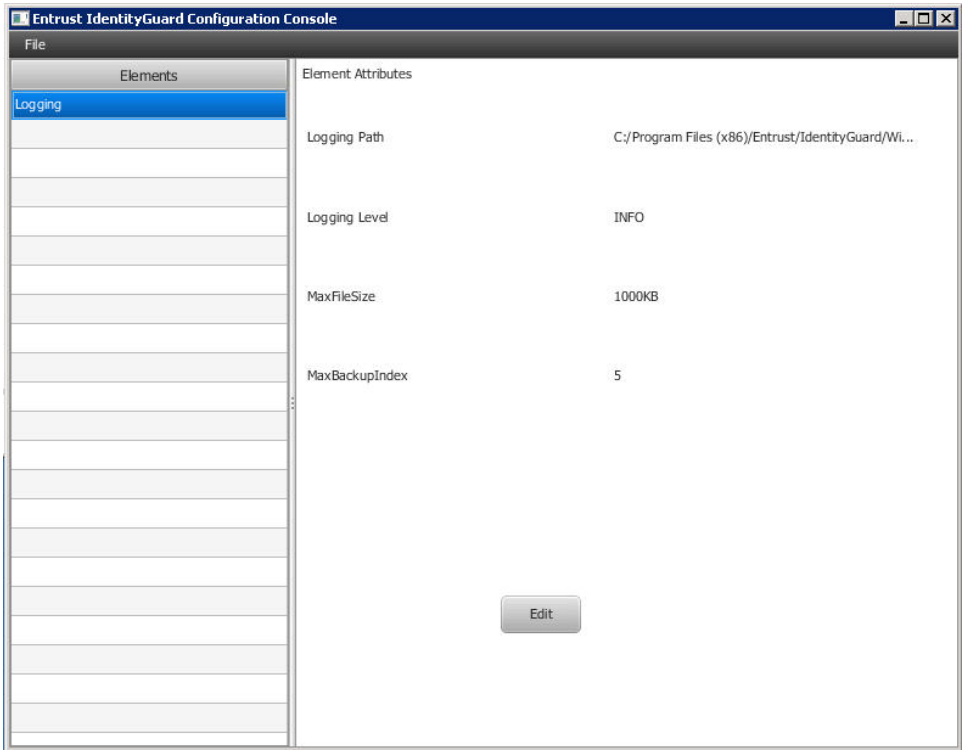
2 Select the file `IdentityGuardFilterLoggerConfig` and then double-click or click **Open** to open the file.



The Configuration Console opens the configuration file and lists the logger configuration file Elements available for editing.

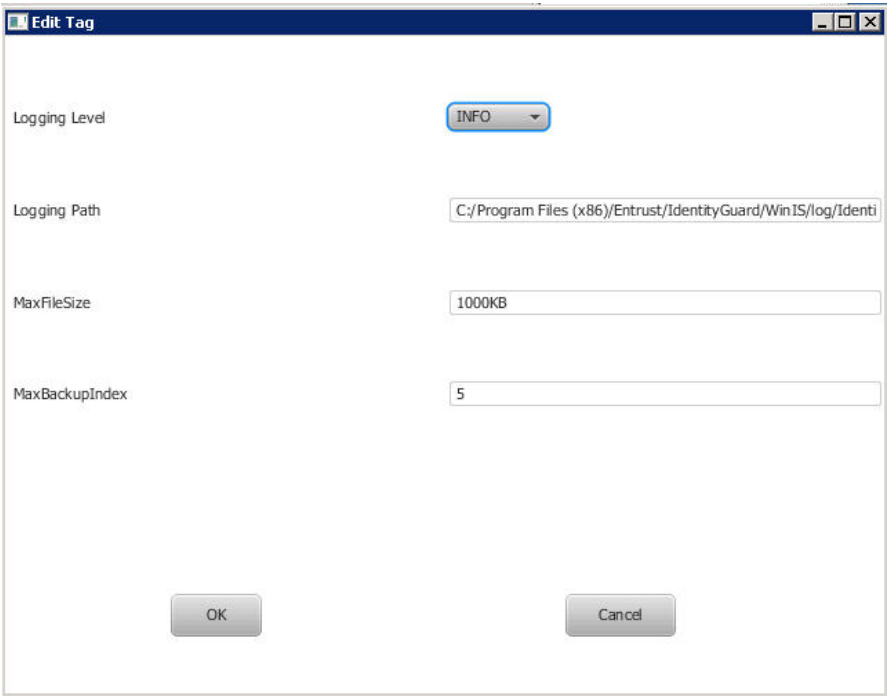


3 Select the **Logging** element in the **Elements** list to display the **Elements Attribute** page.



4 Click **Edit**. The **Edit Tag** page appears.

The **Edit Tag** page appears.



5 Make the required changes and then click **OK** to apply them.

6 To close the configuration file, select **File > Close** in the in Configuration Console.

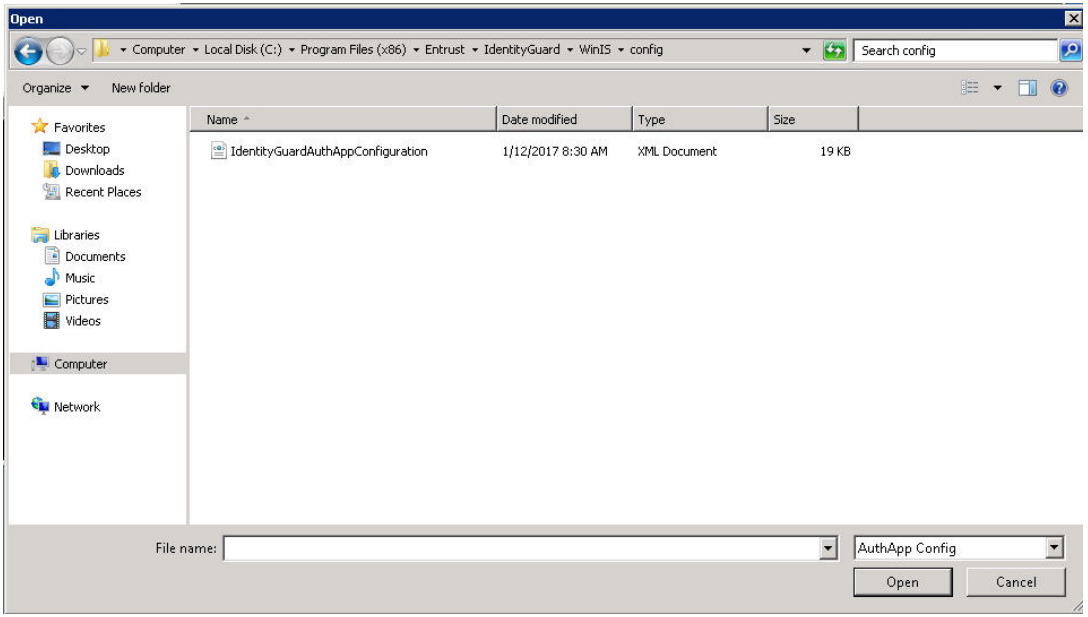
7 To close the Configuration Console, click **Close** on the title bar.

Editing the AuthApp configuration file

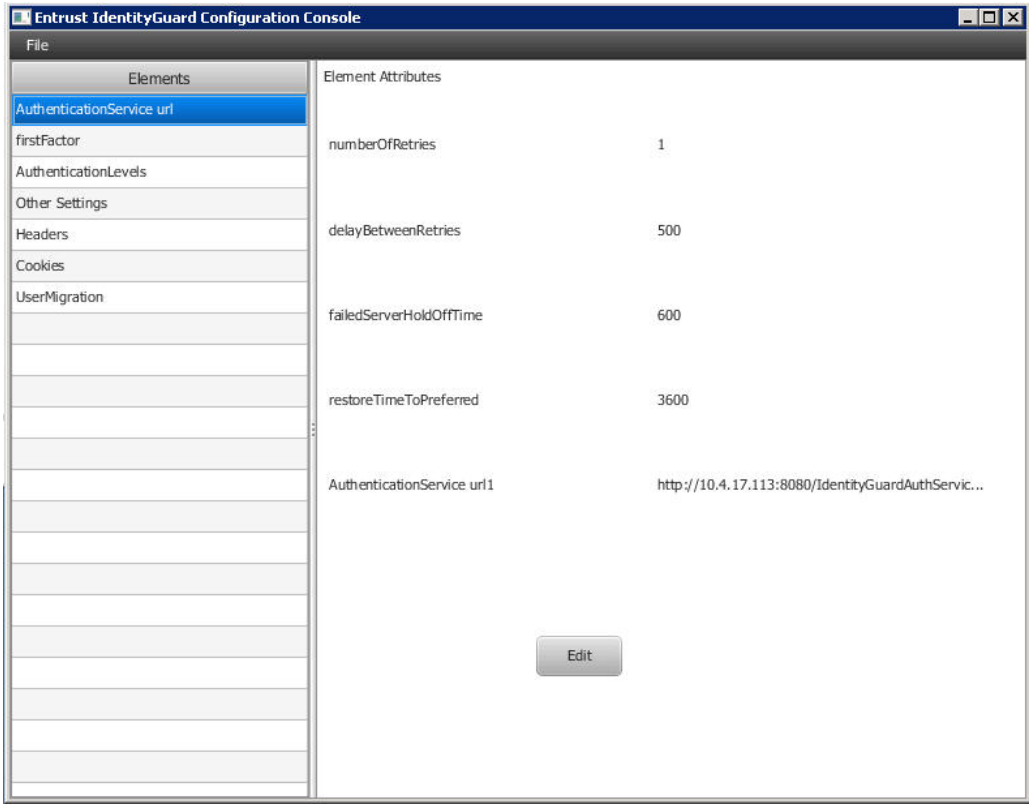
To edit the AuthApp configuration file

1 Browse to ISAPI config folder and select the **Auth App** configuration file type from the drop-down menu (see "Opening a configuration file for editing" on page 128).

2 Select the file `IdentityGuardAuthAppConfiguration` and then double-click or click **Open** to open the file.

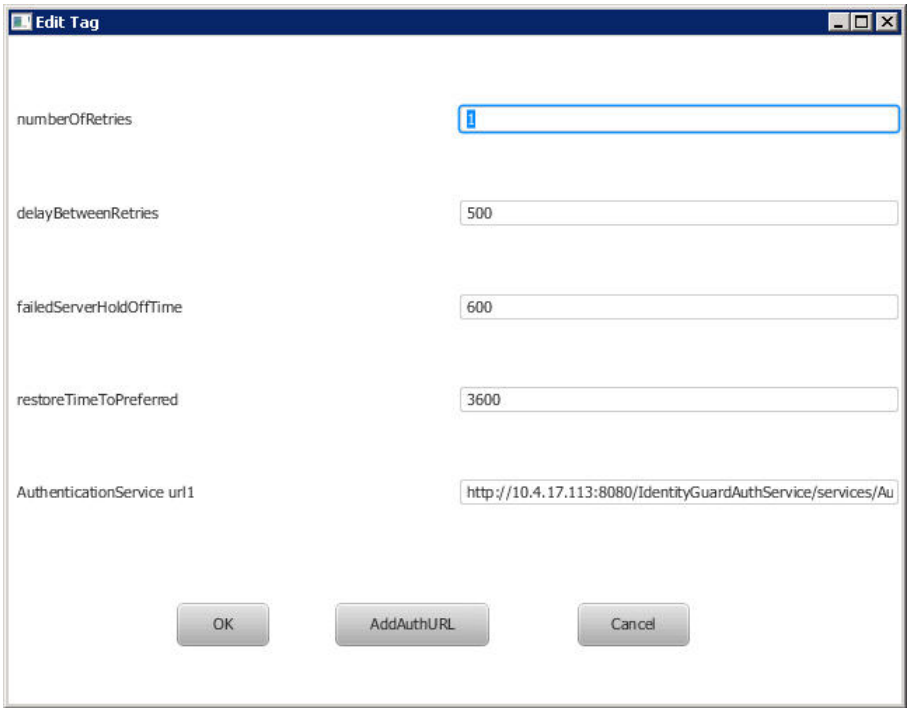


The Configuration Console opens the configuration file and lists the AuthApp configuration file Elements available for editing.



3 To edit any of the elements, select the element in the **Elements** list, for example, **AuthenticationService url** to display the **Element Attributes** page and then click **Edit**.

The **Edit Tag** page appears.



4 Make the required changes and then click **OK** to apply them.

5 To close the configuration file, select **File > Close** in the in Configuration Console.

6 To close the Configuration Console, click **Close** on the title bar.

Note:
If you add additional Authentication Service urls in **AuthApp Configuration file > AuthenticationService url** Element and if you want to remove the additional url later, you can remove the url content from the text box and then click **OK** to apply the changes.

Restarting services after changing configuration files

When you make changes to configuration files, restart the applicable services.

Restarting the World Wide Web Publishing Service

After changing configuration files on an IIS server, you must restart the World Wide Web Publishing Service.

To restart the World Wide Web Publishing Service

1. Open the Windows Services utility.
2. Scroll down to locate the **World Wide Web Publishing Service**.
3. Click **Restart the service**.

   The **Service Control** window appears. It informs you that it is attempting to stop and start the service. Wait for the service to restart.

Configure ISAPI Filter for Identity as a Service

This section includes instructions to configure an Authentication API and how to set up soft token push mutual authentication.

Topics in this section:

Configure an Authentication API

You need to configure the Authentication API to allow authentication using Identity as a Service.

To configure for Identity as a Service authentication

1. Stop World Wide Web Publishing Service.
2. Go to `<ISAPIFilter_install>\WinIS\config` and open the `IdentityGuardAuthAppConfiguration.xml` file.
3. Set the **AuthenticationAPI** element to **`IntelliTrust`**.

   This setting is used to make REST calls to Identity as a Service and to validate the second factor authenticators against Identity as a Service.

   a. To configure the AuthenticationAPI, locate the following:

   ```
   <AuthenticationAPI></AuthenticationAPI>
   ```

   b. Change the Authentication API as follows:

   ```
   <AuthenticationAPI>IntelliTrust</AuthenticationAPI>
   ```

4. Configure the Application ID.

   The Application ID is the unique identifier of the Identity as a Service authentication API application. This is ID is created in Identity as a Service when you add ISAPI Filter as an authentication API to Identity as a Service.

   a. To configure the ApplicationID, locate the following:

   ```
   <ApplicationID></ApplicationID>
   ```

   b. Add the ApplicationID created in Identity as a Service. For example:

```
<ApplicationID>1d123f45-d5c8-45f5-a614-e3f123c45be6</Applicatio
nID>
```

5. Save and close `IdentityGuardAuthAppConfiguration.xml`.
6. Restart World Wide Web Publishing Service.

Configure ISAPI filter for soft token push mutual challenge

Mutual authentication challenge requires users to respond to a mutual push authentication challenge. When enabled, users must match the challenge that appears on the second-factor page of ISAPI filter with the mutual challenge shown in their Entrust Identity soft token app.

This procedure assumes that you have already integrated ISAPI Filter with Identity as a Service. See the Identity as a Service Technical Integration Guides online help for assistance.

Configure soft token push mutual challenge

Follow these steps to configure soft token push mutual challenges. For further assistance, see the Identity as a Service Administrator Help .

To configure soft token push mutual challenge

1. Log in to Identity as a Service as an administrator.
2. Go to **Home > Policies > Authenticators > Entrust Soft Token**. The **Entrust Soft Token Authenticator** page appears.
3. Select **Enable Mutual Challenge** for Entrust Soft Token authenticator.

   See “Modify Entrust Soft Token authenticator” in the I**dentity as a Service Administrator Online Help** for more information.
4. Create a custom authentication flow with **External/Password Authentication** for first-factor and Entrust soft token push for second-factor authentication.

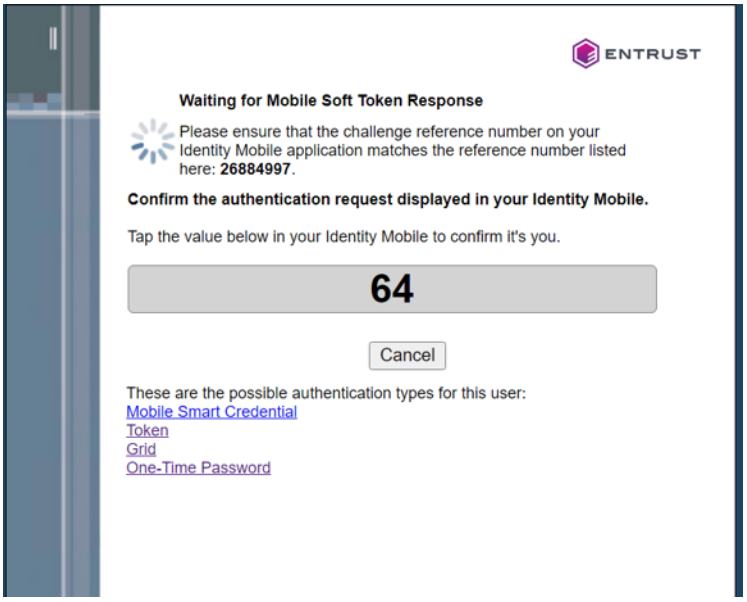
   See “Create authentication flows” in the **Identity as a Service Administrator Online Help**.
5. Using the authentication flow you created, create a resource rule to protect ISAPI Filter for mutual challenge authentication with IDaaS.

   See “Create IDaaS applications resource rules” in the *Identity as a Service Administrator Guide*.

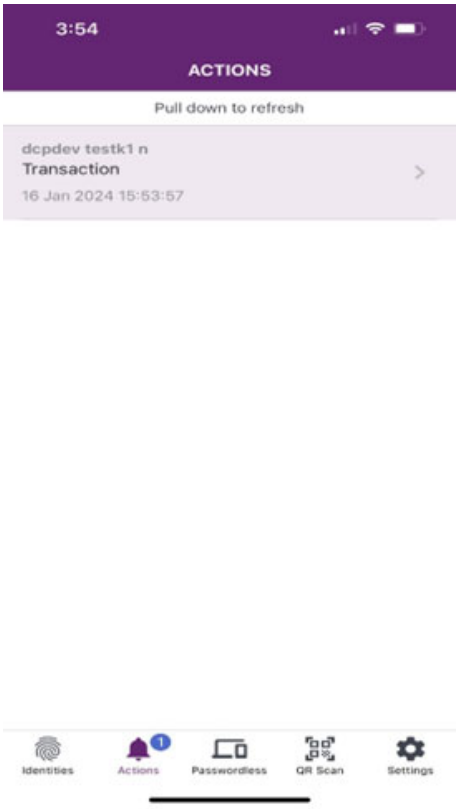
How soft token with mutual challenge works

1. The user accesses the ISAPI resource and enters their username and password in the first-factor page.

2 On the second-factor page, the mutual authentication push token challenge number appears.

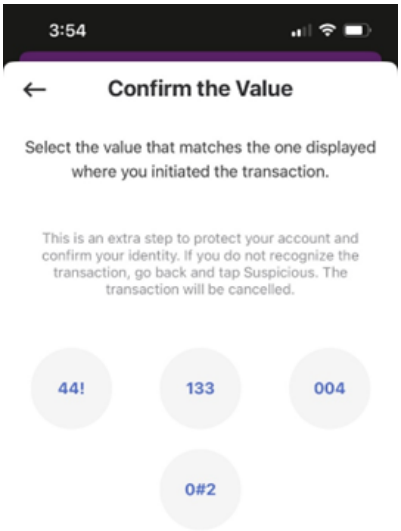


3 The user receives the notification in their mobile device in an Entrust Identity app.



4 The user clicks the **Actions** button to see more about the request and then clicks **Confirm**.

5 The user clicks the value that appears on the second-factor authentication page.



6 After successful authentication, the user is automatically logged into Entrust Desktop for Windows.

7 The user must return to the first-factor authentication page if mutual authentication is unsuccessful.

Configuring ISAPI Filter for Entrust Identity Enterprise Server

You need to configure the Authentication API to allow authentication using Entrust Identity Enterprise.

To configure for Identity as a Service authentication

1. Stop World Wide Web Publishing Service.
2. Go to `<ISAPIFilter_install>\WinIS\config` and open the `IdentityGuardAuthAppConfiguration.xml` file.
3. Set the **AuthenticationAPI** element to **`IdentityGuard`**.

   This setting is used to make SOAP calls to Entrust Identity Enterprise and to validate the second factor authenticators against Entrust Identity Enterprise.

   a. To configure the AuthenticationAPI, locate the following:

   ```
   <AuthenticationAPI></AuthenticationAPI>
   ```

   b. Change the Authentication API as follows:

   ```
   <AuthenticationAPI>IdentityGuard</AuthenticationAPI>
   ```
4. Provide the Entrust Identity Enterprise server and configuration elements. See "Configuring second-factor authentication" on page 168
5. Save and close `IdentityGuardAuthAppConfiguration.xml`.
6. Restart World Wide Web Publishing Service.

Configuring logging

The solution allows you to configure logging for both the authentication application and the ISAPI Filter independently. The solution uses Apache logging packages to implement logging.

The authentication application uses Apache log4net 2.0.12. For more detailed information read the Apache documentation at:

http://logging.apache.org/log4net/release/sdk/log4net.Appender.RollingFileAppenderMembers.html

The filter uses Apache log4cxx 1.1.0. For more detailed information read the Apache documentation at:

http://logging.apache.org/log4cxx/apidocs/classlog4cxx_1_1_rolling_file_appender-members.html

Topics in this section:

Location of log files

The log files are located at

`C:\Program Files\Entrust\Identity\WinIS\log`

The log file for:

- the ISAPI Filter component is `IdentityGuardFilterSystem.log`
- the authentication application is `IdentityGuardISAPIAuthApp.log`

Changing the logging level

You can configure the default logging level attribute for the authentication application and the filter.

Changing the Authentication application logging level

The default logging level for the authentication application is `INFO`. The possible values are:

- OFF
- FATAL
- ERROR
- WARN
- INFO
- DEBUG
- ALL

These levels show increasing amounts of information.

To change the authentication application logging level

1. Open the `IdentityGuardAuthAppConfiguration.xml` file either manually using a text editor or using the Configuration Console (see"Configuring ISAPI Filter using the Configuration Console" on page 127).
2. Find the `Logging` element, and the `level` child element.
3. Change the `value` attribute to the level you want. The default is INFO.

   ```
   <level value="DEBUG" />
   ```
4. Save and close `IdentityGuardAuthAppConfiguration.xml`.

Note:
The DEBUG and TRACE log levels generate a lot of logs. When you have finished troubleshooting, set the logging level back to INFO to avoid slowing down your system.

Changing the ISAPI Filter logging level

You can change the ISAPI Filter logging in two places, global filter logging and logging for each protected host. You can configure the log level for each separately. If you are increasing the logging level for troubleshooting purposes, it is best to increase both the global level and the level for each protected host, to get the maximum amount of information.

The default logging level for the ISAPI Filter is `INFO`. The possible values are:

- OFF
- FATAL
- ERROR
- WARN
- INFO
- DEBUG
- TRACE
- ALL

These levels show increasing amounts of information.

To change the global filter logging level

1 Open the `IdentityGuardFilterLoggerConfig.xml` file either manually using a text editor or using the Configuration Console (see"Configuring ISAPI Filter using the Configuration Console" on page 127).

Note:
This is the logging configuration file for the filter. It is not the filter configuration file.

2 Find the `level` element.

3 Change the `value` attribute to the level you want; for example:

```
<level value="DEBUG" />
```

4 Save and close `IdentityGuardFilterLoggerConfig.xml`.

Note:
The DEBUG and TRACE log levels generate a lot of logs. When you have finished troubleshooting, set the logging level back to INFO to avoid slowing down your system.

To change the protected host filter logging level

1 Open the `IdentityGuardFilterConfiguration.xml` file.

2 Find the `LogLevel` element under the `Logging` element.

3 Change the setting to the level you want; for example:

```
<LogLevel>DEBUG</LogLevel>
```

4 Save and close `IdentityGuardFilterConfiguration.xml`.

Note:
The DEBUG and TRACE log levels generate a lot of logs. When you have finished troubleshooting, set the logging level back to INFO to avoid slowing down your system.

Configuring the log file settings

You can configure the settings affecting the log files, such as the name of the log files, how many backups to keep, and so on.

To configure the log file settings for the authentication application

1 Open the configuration file, `IdentityGuardAuthAppConfiguration.xml` either manually using a text editor or using the Configuration Console (see"Configuring ISAPI Filter using the Configuration Console" on page 127).

2 Locate the section that begins with:

```
<!-- Logging settings for auth app -->
```

3 Modify the settings described below, depending on how you want to configure the log files.

- `file`

  This setting specifies the name and location of the log file. For example:

  ```
  <file value="C:\\Program
  Files\\Entrust\\IdentityGuard\\WinIS\\log\\IdentityGuardISAPIAu
  thApp.log"/>
  ```

- `appendToFile`

  This setting contains a Boolean value. If `true`, then new logging information is appended at the bottom of the log file. If `false`, then new logging information is written to a new log file, after renaming the previous log file by adding the suffix `.#` where `#` is an integer. For example, a log file named `authapp.log` is renamed to `authapp.log.1` and a new `authapp.log` is created. For example:

  ```
  <appendToFile value="true" />
  ```

- `maximumFileSize`

  This setting specifies the maximum size the log file can reach, before a new log file is created. When the log file reaches this size, it is renamed and a new log file is created. For example:

```
<maximumFileSize value="1000KB" />
```

- `maxSizeRollBackups`

  This setting specifies the number of backups of the log file to keep. Every time a new log file is created, all previous log files are renamed by adding the suffix `.#` where `#` is an integer. The value in this setting determines how many renamed files are kept before deleting. If `10` is specified, then 10 renamed files are kept as well as the active log file. Every time a new log file is created the oldest renamed file (with a `.10` suffix) is deleted. For example:

  ```
  <maxSizeRollBackups value="10" />
  ```

4 Save and close `IdentityGuardAuthAppConfiguration.xml`.

To configure the log file settings for the filter

1 Open the file `IdentityGuardFilterLoggerConfig.xml` either manually using a text editor or using the Configuration Console (see"Configuring ISAPI Filter using the Configuration Console" on page 127).

2 Locate the section that begins with:

```
<appender name="RollingFileAppender"
```

3 Modify the parameters described below, depending on how you want to configure the log files.

- `file`

  This setting specifies the name and location of the log file. For example:

  ```
  <param name="file" value="C:/Program
  Files/Entrust/IdentityGuard/WinIS/log/IdentityGuardFilterSystem
  .log" />
  ```

- `Append`

  This setting contains a Boolean value. If `true`, then new logging information is appended at the bottom of the log file. If `false`, then new logging information is written to a new log file, after renaming the previous log file by adding the suffix `.#` where `#` is an integer. For example a log file named `filter.log` is renamed to `filter.log.1` and a new `filter.log` is created. For example:

  ```
  <param name="Append" value="true" />
  ```

- `MaxFileSize`

  This setting specifies the maximum size the log file can reach, before a new log file is created. When the log file reaches this size, it is renamed and a new log file is created. For example:

  ```
  <param name="MaxFileSize" value="1000KB" />
  ```

- `MaxBackupIndex`

This setting specifies the number of backups of the log file to keep. Every time a new log file is created, all previous log files are renamed by adding the suffix `.#` where `#` is an integer. The value in this setting determines how many renamed files are kept before deleting. If `10` is specified, then 10 renamed files are kept as well as the active log file. Every time a new log file is created the oldest renamed file (with a `.10` suffix) is deleted. For example:

```
<param name="MaxBackupIndex" value="10" />
```

4 Save and close `IdentityGuardFilterLoggerConfig.xml`.

Mapping authentication application users to Entrust Identity Enterprise users

For Integrated Windows Authentication (IWA) and Outlook Web Access (OWA) authentication, the user IDs that your users enter are mapped to an Entrust Identity Enterprise user ID. The ISAPI Filter solution, by default, supports several different formats for the IWA and OWA user IDs, however, you can customize these mappings to meet your requirements.

Note:
In both cases, the Entrust Identity Enterprise group used for the mapping depends on whether the `<Group>` option is specified in the corresponding `AuthMethod` configuration. If it is, then that Entrust Identity Enterprise group is used to look up the user. If not, then an Entrust Identity Enterprise group is not specified, and by default, all Entrust Identity Enterprise groups are searched for that user name.

Note:
The `<Group>` sub-element is used only during initial authentication to determine the user's group—it is not used during step-up authentication.

Back up to a different folder any files that you are going to modify. You do not need to restart the ISAPI Filter after making changes. The changes are picked up on the next login attempt.

Topics in this section:

Customizing user mapping for Outlook Web Access (OWA)

The `OWAUseridMapping.cs` file contains the class `OWAUseridMapping`. This class is used to map an OWA user to an Entrust Identity Enterprise user.

It contains a single static method, `MapOWAUser2IdentityGuardUser()`, which defines the user ID mapping.

By default, this method supports three formats for the OWA user ID:

- Windows domain format: `domain\username`; for example `entrust\test1`.
- Email ID format: `username@domain`; for example `test1@entrust.com`.
- User name only, if you are using a single domain; for example `test1`.

In every case the user ID is extracted and mapped to an Entrust Identity Enterprise username.

You can modify this code if you want to create your own custom user ID mapping. The file contains detailed comments to help you in performing your customization.

To map a Microsoft Outlook Web Access (OWA) user ID to an Entrust Identity Enterprise user ID, follow the instructions below.

To customize user mapping for Outlook Web Access (OWA)

1. Open the file `C:\Program Files\Entrust\Identity\WinIS\webapp\IdentityGuardAuth\App_Code\OWAUseridMapping.cs`
2. Locate the `MapOWAUser2IdentityGuardUser()` method, which defines the Microsoft user ID mapping.
3. Modify the code to correctly map the OWA users to the Entrust Identity Enterprise users. The solution supports, as a standard feature, entry of the user name in these formats:
   - `domain\username`
   - `username@domain`
   - `username`

Customizing user mapping for Integrated Windows Authentication (IWA)

The `IntegratedAuthUseridMapping.cs` file contains the class `IntegratedAuthUseridMapping`. This class is used to map a Windows user to an Entrust Identity Enterprise user.

It contains a single static method, `MapUser2IdentityGuardUser()`, which defines the user ID mapping. You can modify this code if you want to perform your own custom user ID mapping.

The default implementation of mapping removes the Windows domain from the user ID, and attempts to use the Windows user name as the Entrust Identity Enterprise user name. For example, if the Windows user ID is `mycorp.com\user1`, then it maps the user name to an Entrust Identity Enterprise user called `user1`.

To map an IWA user ID to an Entrust Identity Enterprise user ID, follow the instructions below.

To modify the authentication application for IWA

1. Open the file

   `C:\Program Files\Entrust\Identity\WinIS\webapp\IdentityGuardAuth\App_Code\IntegratedAuthUseridMapping.cs`

2. Locate the `IntegratedAuthUseridMapping()` method, which defines the Microsoft user ID mapping.
3. Modify the code to correctly map the IWA users to the Entrust Identity Enterprise users.

The file contains detailed comments to help you in performing your customization.

Configuring PCI DSS authentication

PCI DSS works with OWA, RDWEB and igFormsBased and is valid for grid, OTP, token, and MobileST (token push) second factor authentications only.

You can configure the PCI DSS after installation either manually editing configuration files or using the Configuration Console.

To configure PCI-DSS authentication

1. Open `IdentityGuardAuthAppConfiguration.xml` for editing either manually using a text editor or using the Configuration Console (see"Configuring ISAPI Filter using the Configuration Console" on page 127).
2. Change the authentication order as follows:
   a. Near the top of the file under the <Audit> element, modify the following: Enable_PCI-DSS_Support>true</Enable_PCI-DSS_Support>

   b Set the attribute value to true.
3. Add the PCI-DSS supported authenticators, as required.
   a. Scroll to the <AuthenticationMethods> element.
   b. Modify the following to add supported authenticators:

```
<Authenticator>
<Grid/>
<Token/>
<OTP/>
<MobileST/>
</Authenticator>
```

4. Save and close `IdentityGuardFilterConfiguration.xml`.
5. Restart the ISAPI Filter and the Authentication Application. For instructions, see "Restarting services after changing configuration files" on page 141.

Configuring first-factor authentication

You can change the first-factor authentication method after installation.

To reconfigure first-factor authentication

1. Open `IdentityGuardAuthAppConfiguration.xml` for editing either manually using a text editor or using the Configuration Console (see"Configuring ISAPI Filter using the Configuration Console" on page 127).
2. Scroll to the `<FirstFactors>` section.
3. Add your first-factor methods. For example, to use Entrust Identity Enterprise password, edit the `<FirstFactors>` section to look like similar to this:

   ```
   <FirstFactors>
      <FirstFactor id="igFormsBased">
         <FormsBased page="IdentityGuardLogin.aspx"/>
      </FirstFactor>
   </FirstFactors>
   ```

   where `igFormsBased` is any name you choose for this first-factor authentication method.
4. Comment out the `FirstFactor` lines you were using before the change. Add the following characters at the beginning and end of each line to comment them out:

   `<!--` `FirstFactor id="iwa"` **`-->`**
5. Save and close `IdentityGuardAuthAppConfiguration.xml`.
6. Open `IdentityGuardFilterConfiguration.xml` for editing.
7. Scroll to the `ProtectedURLs` definition statement.
8. Edit the `ProtectedURLs` definition by including the first-factor ID value you added in Step 3, above:

   ```
   <ProtectedURLs authlevel="1" firstfactorid="igFormsBased">
      <URL authlevel="1">/Protect/*</URL>
   </ProtectedURLs>
   ```
9. Save and close `IdentityGuardFilterConfiguration.xml`.
10. Restart the ISAPI Filter and the Authentication Application. For instructions, see "Restarting services after changing configuration files" on page 141.

    ```
    <UserIdHeader name="EntrustIdentityGuardUserId"
    encoding="base64"/>
    <SSLOnlyCertAuth>false<SSLOnlyCertAuth/>
    ```

Configuring shared secret

You can configure the shared secret to allow users to pass the user's shared secrets to a protected application in a configured secure header.

To configure the shared secret

1. Open `IdentityGuardFilterConfiguration.xml` for editing either manually using a text editor or using the Configuration Console (see"Configuring ISAPI Filter using the Configuration Console" on page 127).
2. Add the shared secret header just after the `CertSubjectHeader` section. If `CertSubjectHeader` is not present, add it after `UserIdHeader`. The attribute `name` is a configurable header name, the attribute `encoding` specifies the encoding type and the attribute `value` is the name of the shared secret configured in Entrust Identity Enterprise for the user.

   ```
   <UserIdHeader name="EntrustIdentityGuardUserId"
   encoding="base64"/>

   <SSLOnlyCertAuth>false<SSLOnlyCertAuth/>

   <CertSubjectHeader name="EntrustIdentityGuardCertSubject"
   encoding="base64"/>

   <UserSharedSecretHeader name="TruePassUser" encoding="base64"
   value="useralias"/>
   ```

   Where `useralias` is the name of the shared secret in Entrust Identity Enterprise server.
3. Save and close `IdentityGuardFilterConfiguration.xml`.
4. Restart the ISAPI Filter and the Authentication Application. For instructions, see "Restarting services after changing configuration files" on page 141.

Changing authentication features after installation

You can change several authentication features after an IIS-only installation.

- You can change the authentication order; that is, present the password to the user first followed by second-factor authentication (the default), or present second-factor authentication first followed by the password. This applies to installations that use OWA, Entrust Identity Enterprise Password Authentication, Generic Forms Based Authentication, or Remote Desktop Web Access.
- You can switch between forms-based authentication and forms-based passwordless authentication for first-factor authentication. This option is “passwordless” because it authenticates a user using a token and a PVN, without a password required.
- You can switch between one-step and two-step token authentication. (One-step is valid for token authentication only.)

To reconfigure authentication features

1 Open `IdentityGuardAuthAppConfiguration.xml` for editing either manually using a text editor or using the Configuration Console (see“Configuring ISAPI Filter using the Configuration Console” on page 127).

2 To change the authentication order:

   a Near the top of the file under the `<Audit>` element, either modify or add the following:

```
<ChangeAuthenticationOrder>true</ChangeAuthenticationOrder>
```

   b Set the attribute value to `true` or `false`.

3 To switch between password and passwordless authentication on generic forms:

   a Near the top of the file under the `<Audit>` element, either modify or add the following:

```
<Password-lessAuthentication>false</Password-lessAuthentication>
```

   b Set the attribute value to `true` or `false`.

4 To change between one-step and two-step token authentication:

   a Scroll to the `<AuthenticationMethods>` element.

   b Find the authenticator that defines <Token>. It will look something like this:

```
<AuthMethod id="Token">
    <Authenticator>
    <Token>
```

```
    <OneStep OverrideShowPVNUI="true"/>
    </Token>
    </Authenticator>
</AuthMethod>
```

c Modify the `OverrideShowPVNUI` attribute value to `true` or `false`.

5 Save and close the `IdentityGuardAuthAppConfiguration.xml` file.

6 Restart the applicable Web or firewall service. For instructions, see "Restarting services after changing configuration files" on page 141.

Defining second-factor authentication levels and methods

When you installed the authentication application, you selected a second-factor authentication method (for example, grid authentication) as well as an authentication level (for example, 1). To change the second-factor authentication method after installation, follow the procedures below.

Each authentication method is defined in the authentication application configuration file, and referenced in the filter configuration file. See also "Authentication levels" on page 27.

Topics in this section include:

Defining an authentication level

Protected URLs are assigned levels in the filter configuration file, and the meaning of a level is defined in the authentication application configuration file. You can modify an existing authentication level, or define your own levels, by modifying the authentication application configuration file. The ISAPI Filter solution treats Level 1 as the standard level, Level 2 as stricter, and so on.

To define a new authentication level

1. Open the file `IdentityGuardAuthAppConfiguration.xm` either manually using a text editor or using the Configuration Console (see"Configuring ISAPI Filter using the Configuration Console" on page 127).
2. Find the `AuthenticationLevels` element. For example:

```
<AuthenticationLevels>
  ...
</AuthenticationLevels>
```

3. Define a `Level` element for each level used in the filter configuration file (`IdentityGuardFilterConfiguration.xml`).
   a. Assign a level number to the `Level` element. For example:

```
<AuthenticationLevels>
  <Level number="1">
  </Level>
</AuthenticationLevels>
```

   b. Define a child element `AuthMethod`. For example:

```
<AuthenticationLevels>
  <Level number="1">
    <AuthMethod ref="policyRBA" />
  </Level>
</AuthenticationLevels>
```

The `ref` attribute references the `AuthMethod` element which appears later in the same file. You can give it any value you want, as long as it matches the value you assign to the `id` attribute of the `AuthMethod` element. See "Defining an authentication method" on page 164.

4 Repeat Step 3 for each level that you want to define.

The authentication application configuration file must contain at least one `Level` and one `AuthMethod` definition.

5 Save and close `IdentityGuardAuthAppConfiguration.xml`.

6 Restart the applicable Web or firewall service. For instructions, see "Restarting services after changing configuration files" on page 141.

Defining an authentication method

Each authentication level defined in the authentication application configuration file must have a matching authentication method definition.

To define an authentication method

1 Open the file `IdentityGuardAuthAppConfiguration.xml` either manually using a text editor or using the Configuration Console (see"Configuring ISAPI Filter using the Configuration Console" on page 127).

2 Find the `AuthenticationMethods` element. For example:

```
<AuthenticationMethods>
  ...
</AuthenticationMethods>
```

3 Define an `AuthMethod` element for each level defined in the `AuthenticationLevels` section of the same file.

See "Defining an authentication level" on page 163.

a Define an `AuthMethod` element and assign a value to its `id` attribute. For example:

```
<AuthenticationMethods>
  <AuthMethod id="policyRBA">
  </AuthMethod>
</AuthenticationMethods>
```

The `id` value should be the same as the `ref` value in the authentication level definition that you want to match, and must be otherwise unique.

b Define an `Authenticator` element.

This determines what type of authentication is used for second-factor authentication. For example:

```
<AuthMethod id="policyRBA">
  <Authenticator>
    <Policy/>
  </Authenticator>
</AuthMethod>
```

The example determines that the second-factor authentication method used by `policyRBA` is governed by the authentication policies configured in Entrust Identity Enterprise.

Other possible values for the method are:

– `<Grid/>` for grid authentication
– `<Token/>` for token authentication
– `<MobileSC/>` for mobile smart credential authentication
– `<MobileST/>` for mobile soft token (TVS) authentication
– `<KB/>` for knowledge-based authentication
– `<OTP/>` for out-of-band one-time password authentication

c If you want to use risk-based authentication for this `AuthMethod`, add an `RBA` element after the `Authenticator` element.

```
<AuthMethod id="policyRBA">
  <Authenticator>
    <Policy/>
  </Authenticator>
  <RBA>
    ...
  </RBA>
</AuthMethod>
```

For detailed instructions on configuring risk-based authentication, see “Configuring risk-based authentication” on page 185.

d If you want to specify an Entrust Identity Enterprise group to use for all users, when this `AuthMethod` is used, enter a `Group` element. For example:

```
<AuthMethod id="policyRBA">
  <Authenticator>
```

```
    <Policy/>
  </Authenticator>
  <RBA>
    ...
  </RBA>
  <Group>IGGroupName</Group>
</AuthMethod>
```

The example specifies an Entrust Identity Enterprise group named `IGGroupName`.

4 Repeat Step 3 for each authentication method that you want to define.

The authentication application configuration file must contain at least one `Level` and one `AuthMethod` definition.

5 Save and close `IdentityGuardAuthAppConfiguration.xml`.

6 Restart the applicable Web or firewall service. For instructions, see "Restarting services after changing configuration files" on page 141.

Note:
If you do not specify a group in `AuthMethod`, then the group is set to null. This instructs Entrust Identity Enterprise to search all groups when attempting to find this user.
If your user-search requirements are not met by searching all Entrust Identity Enterprise groups, or only using one statically configured Entrust Identity Enterprise group for that `AuthMethod`, you can modify this behavior by supplying your own business logic to dynamically determine which IdentityGuard group this user must belong to.
For example, you could dynamically map the Windows domain to an Entrust Identity Enterprise group. See also "Mapping authentication application users to Entrust Identity Enterprise users" on page 155.

Note:
The `<Group>` sub-element is used only during initial authentication to determine the user's group—it is not used during step-up authentication.

If you want to modify your second-factor authentication, you must edit the authentication application configuration file. The following sections describe how to configure your solution for different authentication types.

The default configuration file contains sample definitions for different authentication types, such as grid, token, and so on, with and without risk-based authentication. These are commented out by default. You can remove the commenting characters, and modify the examples to suit your purposes. See “Configuring second-factor authentication” on page 168.

Configuring second-factor authentication

This section describes how to configure your solution for different authentication types. See also “Defining an authentication level” on page 163 and “Defining an authentication method” on page 164.

Topics in this section:

Configuring second factor authentication to none

You can configure the second factor authentication type to none. When this feature is configured, the user does not need to provide second factor authentication.

In order to set second factor authentication to none, the Entrust Identity Enterprise policy needs to include the following:

- Normal Security Authentication Types should have `None`
- Allow `None` Authentication Type to `Yes`

To set second factor to none

1. Open the `IdentityGuardAuthAppConfiguration.xml` file either manually using a text editor or using the Configuration Console (see“Configuring ISAPI Filter using the Configuration Console” on page 127).
2. Locate the Authenticator section and update it as follows:

```
<Authenticator>
<None/>
</Authenticator>
```

3. Save and close `IdentityGuardAuthAppConfiguration.xml`.

Configuring grid authentication

You can use grid authentication as your second-factor authentication type by editing the authentication application configuration file.

To configure grid authentication

1. Open the `IdentityGuardAuthAppConfiguration.xml` file either manually using a text editor or using the Configuration Console (see"Configuring ISAPI Filter using the Configuration Console" on page 127).
2. Locate the `AuthenticationMethods` element.

```
<AuthenticationMethods>
  ...
</AuthenticationMethods>
```

3. Define an `AuthMethod` element as shown in the example below.

```
<AuthenticationMethods>
  <AuthMethod id="gridAuth">
    <Authenticator>
      <Grid />
    </Authenticator>
  </AuthMethod>
  ...
</AuthenticationMethods>
```

4. In the `AuthenticationLevels` element, enter the name "`gridAuth`" in the `ref` attribute of the `Level` element to which you want to assign this authentication type.

   For example, if you want to assign grid authentication to your Level 1 authentication, find the corresponding `Level` element under the `AuthenticationLevels` element, and enter the name "`gridAuth`" in the `ref` attribute.

```
<Level number="1">
  <AuthMethod ref="gridAuth" />
</Level>
```

   The name "`gridAuth`" is just an example. You can create any name you like, but it must be the same in the `Level` element and the `AuthMethod` element, including the same case, and must be otherwise unique.
5. Save and close `IdentityGuardAuthAppConfiguration.xml`.
6. To allow users to use their grid as an alternative authenticator, see "Configuring alternate authenticators" on page 211.

7 Restart the applicable Web or firewall service. For instructions, see “Restarting services after changing configuration files” on page 141.

Note:
The name you use for the `id` and `ref` attributes must be in matching case for the solution to work properly.
For example these two match:
`<AuthMethod id="gridAuth">` and `<AuthMethod ref="gridAuth" />`
These two do not match:
`<AuthMethod id="GridAuth">` and `<AuthMethod ref="gridauth" />`

Configuring token authentication

The ISAPI solution supports use of dynamic passwords generated by hardware tokens. It supports both response-only, and challenge-response tokens. You can use token authentication as your second-factor authentication type by editing the authentication application configuration file.

To configure token authentication

1 Open the `IdentityGuardAuthAppConfiguration.xml` file either manually using a text editor or using the Configuration Console (see“Configuring ISAPI Filter using the Configuration Console” on page 127).

2 Locate the `AuthenticationMethods` element.

```
<AuthenticationMethods>
  ...
</AuthenticationMethods>
```

3 Define an `AuthMethod` element as shown in the example below.

```
<AuthenticationMethods>
  <AuthMethod id=”tokenAuth”>
    <Authenticator>
      <Token />
    </Authenticator>
  </AuthMethod>
  ...
</AuthenticationMethods>
```

4 In the `AuthenticationLevels` element, enter the name "`tokenAuth`" in the `ref` attribute of the `Level` element to which you want to assign this authentication type.

For example, if you want to assign token authentication to your Level 1 authentication, find the corresponding `Level` element under the `AuthenticationLevels` element, and enter the name "`tokenAuth`" in its `ref` attribute.

```
<Level number="1">
  <AuthMethod ref="tokenAuth" />
</Level>
```

The name "`tokenAuth`" is just an example. You can create any name you like, but it must be the same in the `Level` element and the `AuthMethod` element, including the same case, and must be otherwise unique.

5 Save and close `IdentityGuardAuthAppConfiguration.xml`.

Note:
The name you use for the `id` and `ref` attributes must be in matching case for the solution to work properly.
For example these two match:
`<AuthMethod id="tokenAuth">` and `<AuthMethod ref="tokenAuth" />`
These two do not match:
`<AuthMethod id="TokenAuth">` and `<AuthMethod ref="tokenauth" />`

The configuration file does not differentiate between response-only and challenge-response tokens. Entrust Identity Enterprise determines the token mode of operation, when a challenge is requested.

6 To change token authentication to one-step authentication, see "Changing authentication features after installation" on page 161.

7 To allow users to use their token as an alternative authenticator, see "Configuring alternate authenticators" on page 211.

8 Restart the applicable Web or firewall service. For instructions, see "Restarting services after changing configuration files" on page 141.

Configuring mobile smart credential authentication

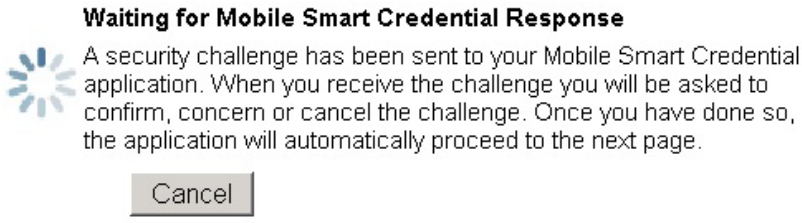
The ISAPI solution supports the use of mobile smart credential for authentication. See the following sections:

- "The mobile smart credential authentication process" on page 172
- "To configure mobile smart credential authentication" on page 172

The mobile smart credential authentication process

The authentication unfolds as follows:

1. Before attempting to log in through the ISAPI filter,
   a. the user downloads the Entrust Identity Mobile Smart Credential app to their Android, BlackBerry, or iOS device.
   b. the user activates the mobile smart credential within the app. The mobile credential contains a digital ID (an X.509 certificate), and other identifying information. For more information about creating, downloading, and activating mobile smart credentials, see the *Entrust Identity Enterprise Server Administration Guide*.
2. The user accesses a URL protected by the ISAPI Filter solution.
3. The user completes a first-factor authentication.
4. The user sees a progress screen, telling them to switch to their mobile device.



5. On their mobile device, the user opens the Entrust Identity Enterprise Mobile Smart Credential app.The app displays a message asking whether they want to authenticate.
6. The user can select **Confirm** to allow the authentication to proceed, **Cancel** to abort the authentication, or **Concern**, to report the authentication as suspicious (possibly because they did not initiate the authentication).
7. If the user selects **Confirm**, they are authenticated, and they see the protected resource they were trying to access.

To configure mobile smart credential authentication

1. Ensure that you have completed other configuration procedures as needed.

a Configure your Entrust Identity Enterprise Server for mobile smart credentials. See the *Entrust Identity Enterprise Administration Guide* for details.

b Configure your Entrust Identity Enterprise Self-Service Module for mobile smart credentials. See the *Entrust Identity Enterprise Self-Service Module Installation and Configuration Guide* for details.

c Have users download, activate, and begin using their mobile smart credential. See the Entrust Identity Enterprise Mobile Smart Credential online help, available from within the app.

2 Open the `IdentityGuardAuthAppConfiguration.xml` file either manually using a text editor or using the Configuration Console (see"Configuring ISAPI Filter using the Configuration Console" on page 127).

3 Locate the `AuthenticationMethods` element.

```
<AuthenticationMethods>
  ...
</AuthenticationMethods>
```

4 Define an `AuthMethod` element as shown in the example below.

```
<AuthenticationMethods>
  <AuthMethod id="MobileSC">
    <Authenticator>
      <MobileSC/>
    </Authenticator>
  </AuthMethod>
  ...
</AuthenticationMethods>
```

5 In the `AuthenticationLevels` element, enter the name "`MobileSC`" in the `ref` attribute of the `Level` element to which you want to assign this authentication type.

For example, if you want to assign mobile smart credential authentication to your Level 1 authentication, find the corresponding `Level` element under the `AuthenticationLevels` element, and enter the name "`MobileSC`" in its `ref` attribute.

```
<Level number="1">
  <AuthMethod ref="MobileSC" />
</Level>
```

The name "`MobileSC`" is just an example. You can create any name you like, but it must be the same in the `Level` element and the `AuthMethod` element, including the same case, and must be otherwise unique.

6 Save and close `IdentityGuardAuthAppConfiguration.xml`.

Note:
The name you use for the `id` and `ref` attributes must be in matching case for the solution to work properly.
For example these two match:
`<AuthMethod id="MobileSC">` and `<AuthMethod ref="MobileSC" />`
These two do not match:
`<AuthMethod id="mobilesc">` and `<AuthMethod ref="Mobilesc" />`

7 Optional. Specify a polling interval, in seconds, by adding the text in bold.

```
<AuthenticationMethods>
  <AuthMethod id=”MobileSC”>
    <Authenticator>
      <MobileSC pollingInterval=”3”/>
    </Authenticator>
  </AuthMethod>
</AuthenticationMethods>
```

The polling interval indicates how long the authentication application waits, in seconds, before calling Entrust Identity Enterprise Server after receiving a request from the user's browser to try and authenticate the mobile smart credential authentication challenge.

8 To allow users to use mobile smart credentials as an alternative authenticator, see “Configuring alternate authenticators” on page 211.

9 Restart the applicable Web or firewall service. For instructions, see “Restarting services after changing configuration files” on page 141.

Configuring mobile soft token (TVS) authentication

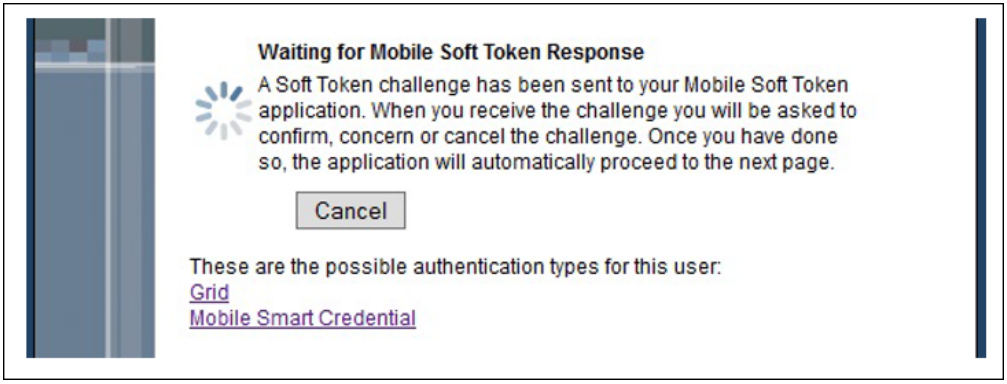
The ISAPI solution supports the use of mobile soft token (TVS) for authentication. See the following sections:

- “The mobile soft token (TVS) authentication process” on page 174
- “To configure mobile soft token (TVS) authentication” on page 175

The mobile soft token (TVS) authentication process

The authentication unfolds as follows:

1. Before attempting to log in through the ISAPI filter, the user downloads the Entrust Identity Mobile Soft Token app to their Android, BlackBerry, or iOS device.
2. The user activates SoftToken app and adds user identities. For more information about creating, downloading, and activating mobile soft token (TVS), see the *Entrust Identity Enterprise Server Administration Guide*.
3. The user accesses a URL protected by the ISAPI Filter solution.
4. The user completes a first-factor authentication.
5. The user sees a progress screen, telling them to switch to their mobile device.



6. On their mobile device, the user opens the Entrust Identity Mobile Soft Token (TVS) app.The app displays a message asking whether they want to authenticate.
7. The user can select **Confirm** to allow the authentication to proceed, **Cancel** to abort the authentication, or **Concern**, to report the authentication as suspicious (possibly because they did not initiate the authentication).
8. If the user selects **Confirm**, they are authenticated, and they see the protected resource they were trying to access.

To configure mobile soft token (TVS) authentication

1. Ensure that you have completed other configuration procedures as needed.
   a. Configure your Entrust Identity Enterprise Server for mobile soft token (TVS). See the *Entrust Identity Enterprise Administration Guide* for details.
   b. Configure your Entrust Identity Enterprise Self-Service Module for mobile soft token (TVS). See the *Entrust Identity Enterprise Self-Service Module Installation and Configuration Guide* for details.

c Have users download, activate, and begin using their mobile smart credential. See the Entrust Identity Mobile Soft Token (TVS) online help, available from within the app.

2 Open the `IdentityGuardAuthAppConfiguration.xml` file either manually using a text editor or using the Configuration Console (see"Configuring ISAPI Filter using the Configuration Console" on page 127).

3 Locate the `AuthenticationMethods` element.

```
<AuthenticationMethods>
  ...
</AuthenticationMethods>
```

4 Define an `AuthMethod` element as shown in the example below.

```
<AuthenticationMethods>
  <AuthMethod id="MobileST">
    <Authenticator>
      <MobileST/>
    </Authenticator>
  </AuthMethod>
  ...
</AuthenticationMethods>
```

5 In the `AuthenticationLevels` element, enter the name "`MobileST`" in the `ref` attribute of the `Level` element to which you want to assign this authentication type.

For example, if you want to assign mobile soft token (TVS) authentication to your Level 1 authentication, find the corresponding `Level` element under the `AuthenticationLevels` element, and enter the name "`MobileST`" in its `ref` attribute.

```
<Level number="1">
  <AuthMethod ref="MobileST" />
</Level>
```

The name "`MobileST`" is just an example. You can create any name you like, but it must be the same in the `Level` element and the `AuthMethod` element, including the same case, and must be otherwise unique.

6 Save and close `IdentityGuardAuthAppConfiguration.xml`.

Note:
The name you use for the `id` and `ref` attributes must be in matching case for the solution to work properly.
For example these two match:
`<AuthMethod id="MobileST">` and `<AuthMethod ref="MobileST" />`
These two do not match:
`<AuthMethod id="mobilest">` and `<AuthMethod ref="Mobilest" />`

7 Optional. Specify a polling interval, in seconds, by adding the text in bold.

```
<AuthenticationMethods>
  <AuthMethod id="MobileST">
    <Authenticator>
      <MobileSC pollingInterval="3"/>
    </Authenticator>
  </AuthMethod>
</AuthenticationMethods>
```

The polling interval indicates how long the authentication application waits, in seconds, before calling Entrust Identity Enterprise Server after receiving a request from the user's browser to try and authenticate the mobile soft token (TVS) authentication challenge.

8 Save and close the `IdentityGuardAuthAppConfiguration.xml` file.

9 To allow users to use mobile smart credentials as an alternative authenticator, see "Configuring alternate authenticators" on page 211.

10 Restart the applicable Web or firewall service. For instructions, see "Restarting services after changing configuration files" on page 141.

Configuring out-of-band OTP authentication

The ISAPI solution supports the use of an out-of-band one-time password (OTP) that is sent to the user when needed, or when a threshold is reached, depending on Entrust Identity Enterprise policy.

Entrust Identity Enterprise allows users to have multiple OTPs. Since OTPs can be used only once, the user's supply of OTPs is reduced with each authentication. When the user's supply of OTPs falls below a threshold, Entrust Identity Enterprise automatically generates and sends a new supply of OTPs. The operation and refresh threshold are defined in Entrust Identity Enterprise policy. See the *Entrust Identity Enterprise Administration Guide* for more information.

Users can select one or more delivery destinations when authenticating, and they are offered the option of having the OTPs re-sent, if necessary.

You can use OTP authentication as your second-factor authentication type by editing the authentication application configuration file.

To configure OTP authentication

1 Open the `IdentityGuardAuthAppConfiguration.xml` file either manually using a text editor or using the Configuration Console (see"Configuring ISAPI Filter using the Configuration Console" on page 127).

2 Find the `AuthenticationMethods` element.

```
<AuthenticationMethods>
  ...
</AuthenticationMethods>
```

3 Define an `AuthMethod` element as shown in the example below.

```
<AuthenticationMethods>
  <AuthMethod id="OTPAuth">
    <Authenticator>
      <OTP>
        <AllowManualDelivery>true</AllowManualDelivery>
      </OTP>
    </Authenticator>
  </AuthMethod>
  ...
</AuthenticationMethods>
```

`<AllowManualDelivery>` is an optional element under <OTP>. It is used to allow the administrator to decide the delivery mode.

By default, it is set to `true`, and the OTP can be delivered manually to a user who has no delivery method configured. If it is set to `false`, users without a delivery method see an error page informing them that they do not have a delivery method configured and cannot respond to the challenge.

4 In the `AuthenticationLevels` element, enter the name "`OTPAuth`" in the `ref` attribute of the `Level` element to which you want to assign this authentication type.

For example, if you want to assign OTP authentication to your Level 1 authentication, find the corresponding `Level` element under the `AuthenticationLevels` element, and enter the name "`OTPAuth`" in its `ref` attribute.

```
<Level number="1">
```

```
    <AuthMethod ref="OTPAuth" />
  </Level>
```

The name "OTPAuth" is just an example. You can create any name you like, but it must be the same in the Level element and the AuthMethod element, including matching case, and must be otherwise unique.

Note:
The name you use for the id and ref attributes must be in matching case for the solution to work properly.
For example these two match:
<AuthMethod id="OTPAuth"> and <AuthMethod ref="OTPAuth" />
These two do not match:
<AuthMethod id="OTPAuth"> and <AuthMethod ref="otpauth" />

5 Enable optional display and masking of information values in OTP challenges, as shown below:

```
<IdentityGuardV11ExSupportRequired>true</IdentityGuardV11ExSupportRequired>
```

By default, this element is set to false.

Note:
Identity as a Service is not supported if this setting is set to true.

When users initiate the sending of one-time passwords (OTP) to be used for authentication, they choose the email or phone number to which the OTPs should be sent. This feature shows the contact information values (with masking) in addition to generic labels such as Work Email and Work Phone. For details, see "Set policies for out-of-band OTPs" in the *Entrust Identity Enterprise Server Administration Guide*.

6 Save and close IdentityGuardAuthAppConfiguration.xml.

7 To allow users to use an OTP as an alternative authenticator, see "Configuring alternate authenticators" on page 211.

8 Restart the applicable Web or firewall service. For instructions, see "Restarting services after changing configuration files" on page 141.

9 Ensure that you have completed other configuration procedures as needed.

   a Configure your Entrust Identity Enterprise one-time password polices. See the *Entrust Identity Enterprise Administration Guide* for more information.

b Configure an Entrust Identity Enterprise out-of-band delivery method. See the *Entrust Identity Enterprise Administration Guide* for more information about out-of-band email delivery configuration and out-of-band voice delivery configuration.

If you want to use the Authentify voice delivery mechanism for delivering the one-time-password (OTP), you must configure a personal verification number (PVN) for each user—Authentify requires PVN.

Configuring knowledge-based authentication

The ISAPI solution supports the use of user-configured questions and answers in second-factor authentication.

You can use knowledge-based authentication as your second-factor authentication type by editing the authentication application configuration file.

To configure knowledge-based authentication

1. Open the `IdentityGuardAuthAppConfiguration.xml` file either manually using a text editor or using the Configuration Console (see"Configuring ISAPI Filter using the Configuration Console" on page 127).
2. Find the `AuthenticationMethods` element.

```
<AuthenticationMethods>
  ...
</AuthenticationMethods>
```

3. Define an `AuthMethod` element as shown in the example below:

```
<AuthenticationMethods>
  <AuthMethod id="kbAuth">
    <Authenticator>
      <KB>
        <OverrideKBChallengeSize>2</OverrideKBChallengeSize>
        <MaskAnswers>false</MaskAnswers>
      </KB>
    </Authenticator>
  </AuthMethod>
  ...
</AuthenticationMethods>
```

The `KB` element has an optional child element `OverrideKBChallengeSize`. This contains an integer value that overrides the default number of questions posed to the user when performing a Q&A challenge. The default value is specified by the Entrust Identity Enterprise policy setting. In the example

above, the value has been set to `2`. For information on the `MaskAnswers` element, see "Masking the answers to the questions in knowledge-based authentication" on page 182.

The value must conform to the minimum and maximum `Q&A Challenge Size` policy settings in Entrust Identity Enterprise. See the *Entrust Identity Enterprise Administration Guide* for more information about configuring knowledge-based authentication policies.

4 In the `AuthenticationLevels` element, enter the name "`kbAuth`" in the `ref` attribute of the `Level` element to which you want to assign this authentication type.

For example, if you want to assign knowledge-based authentication to your Level 1 authentication, find the corresponding `Level` element under the `AuthenticationLevels` element, and enter the name "`kbAuth`" in its `ref` attribute.

```
<Level number="1">
  <AuthMethod ref="kbAuth" />
</Level>
```

The name "`kbAuth`" is just an example. You can create any name you like, but it must be the same in the `Level` element and the `AuthMethod` element, including the same case, and must be otherwise unique.

5 Save and close `IdentityGuardAuthAppConfiguration.xml`.

Note:
The name you use for the `id` and `ref` attributes must be in matching case for the solution to work properly.
For example these two match:
`<AuthMethod id="kbAuth">` and `<AuthMethod ref="kbAuth" />`
These two do not match:
`<AuthMethod id="KBAuth">` and `<AuthMethod ref="kbauth" />`

6 To allow users to use Q&A as an alternative authenticator, see "Configuring alternate authenticators" on page 211.

7 Restart the applicable Web or firewall service. For instructions, see "Restarting services after changing configuration files" on page 141.

Masking the answers to the questions in knowledge-based authentication

Some users may be in an environment where it is preferable for their answers not to be displayed on the screen while they are entering them. You can configure knowledge-based authentication to mask these answers, just like a password.

To mask the answers to the questions in knowledge-based authentication

1. Open the `IdentityGuardAuthAppConfiguration.xml` file either manually using a text editor or using the Configuration Console (see"Configuring ISAPI Filter using the Configuration Console" on page 127).
2. Find the `<AuthMethod id="kbAuth">` element under the `<AuthenticationMethods>` element.

   ```
   <AuthMethod id="kbAuth">
   ```

3. Change the value for `MaskAnswers` from the default of `false` to `true`.

   ```
   <MaskAnswers>true</MaskAnswers>
   ```

4. Save and close `IdentityGuardAuthAppConfiguration.xml`.
5. Restart the applicable Web or firewall service. For instructions, see "Restarting services after changing configuration files" on page 141.

Configuring Passkey/FIDO2 authentication

You can use passkey authentication as your second-factor authentication type by editing the authentication application configuration file.

To configure passkey authentication

1. Open the `IdentityGuardAuthAppConfiguration.xml` file either manually using a text editor or using the Configuration Console (see"Configuring ISAPI Filter using the Configuration Console" on page 127).
2. Locate the `AuthenticationMethods` element.

   ```
   <AuthenticationMethods>
   ...
   </AuthenticationMethods>
   ```

3. Define an AuthMethod element as shown in the example below.

   ```
   <AuthenticationMethods>
   <AuthMethod id="passkeyAuth">
   <Authenticator>
   <Passkey />
   ```

```
</Authenticator>
</AuthMethod>
...
</AuthenticationMethods>
```

4 In the `AuthenticationLevels` element, enter the name `passkeyAuth` in the ref attribute of the Level element to which you want to assign this authentication type.

For example, if you want to assign passkey authentication to your Level 1 authentication, find the corresponding Level element under the AuthenticationLevels element, and enter the name `passkeyAuth` in the ref attribute.

```
<Level number="1">
<AuthMethod ref="passkeyAuth" />
</Level>
```

The name `passkeyAuth` is just an example. You can create any name you like, but it must be the same in the `Level` element and the `AuthMethod` element, including the same case and must be otherwise unique.

5 Save and close `IdentityGuardAuthAppConfiguration.xml`.

6 To allow users to use their Passkey as an alternative authenticator, see "Configuring alternate authenticators" on page 211.

7 Restart the applicable Web or firewall service. For instructions, see "Restarting services after changing configuration files" on page 141.

Configuring policy-based authentication

When you configure the ISAPI solution to use policy-based authentication, the second-factor authentication method used is determined by the Entrust Identity Enterprise **Authentication Types** policy. Two users might see different authenticators when they log in; for example knowledge-based for one, and token for the other. Entrust Identity Enterprise determines which second-factor authentication type to present to a user, based on:

- which challenge types are currently supported and valid for that user
- the order in which the second-factor authentication types are specified in the Entrust Identity Enterprise **Authentication Types** policy

See the *Entrust Identity Enterprise Administration Guide* for more information about **Authentication Types** policies.

You can configure your solution to use policy-based second-factor authentication by editing the authentication application configuration file.

To configure policy-based authentication

1 Open the `IdentityGuardAuthAppConfiguration.xml` file either manually using a text editor or using the Configuration Console (see"Configuring ISAPI Filter using the Configuration Console" on page 127).

2 Find the `AuthenticationMethods` element.

```
<AuthenticationMethods>
  ...
</AuthenticationMethods>
```

3 Define an `AuthMethod` element as shown in the example below.

```
<AuthenticationMethods>
  <AuthMethod id="policyAuth">
    <Authenticator>
      <Policy>
        <OverrideKBChallengeSize>2</OverrideKBChallengeSize>
        <MaskAnswers>false</MaskAnswers>
        <AllowManualDelivery>true</AllowManualDelivery>
      </Policy>
    </Authenticator>
  </AuthMethod>
  ...
</AuthenticationMethods>
```

The `Policy` element has two optional child elements, `OverrideKBChallengeSize` and `AllowManualDelivery`.

When Entrust Identity Enterprise policy determines that Q&A authentication should be used for a particular user, this optional setting determines the number of questions to show, if you do not want to use the default value specified in the user's policy.

`OverrideKBChallengeSize` contains an integer value that overrides the number of questions posed to the user when performing a Q&A challenge. In the example above, the value has been set to `2`.

The value must conform to the minimum and maximum `Q&A Challenge Size` policy settings in Entrust Identity Enterprise. See the *Entrust Identity Enterprise Administration Guide* for more information about configuring knowledge-based authentication policies.

`AllowManualDelivery` is used to allow the administrator to decide the delivery mode.

By default, it is set to `true`, and the OTP can be delivered manually to a user who has no delivery method configured. If it is set to `false`, users without a

delivery method see an error page informing them that they do not have a delivery method configured and cannot respond to the challenge.

4 In the `AuthenticationLevels` element, enter the name "`policyAuth`" in the `ref` attribute of the `Level` element to which you want to assign this authentication type.

For example, if you want to assign policy-based authentication to your Level 1 authentication, find the corresponding `Level` element under the `AuthenticationLevels` element, and enter the name "`policyAuth`" in its `ref` attribute.

```
<Level number="1">
  <AuthMethod ref="policyAuth" />
</Level>
```

The name "`policyAuth`" is just an example. You can create any name you like, but it must be the same in the `Level` element and the `AuthMethod` element, including the same case, and must be otherwise unique.

5 Save and close `IdentityGuardAuthAppConfiguration.xml`.

Note:
The name you use for the `id` and `ref` attributes must be in matching case for the solution to work properly.
For example these two match:
`<AuthMethod id="policyAuth">` and `<AuthMethod ref="policyAuth" />`
These two do not match:
`<AuthMethod id="PolicyAuth">` and `<AuthMethod ref="policyauth" />`

Configuring risk-based authentication

The ISAPI Filter solution allows you to use Entrust IdentityGuard and Entrust Identity as a Service risk-based authentication (RBA). You can define the RBA settings you want during installation of the solution. However, editing the authentication application configuration file allows you to define separate risk-based settings for each second-factor authentication method. See the *Entrust Identity Enterprise Administration Guide* or the Identity as a Service Administrator Help for more information about risk-based authentication policies.

To configure risk-based authentication for Identity as a Service

1. Open the `IdentityGuardAuthAppConfiguration.xml` file either manually using a text editor or using the Configuration Console (see"Configuring ISAPI Filter using the Configuration Console" on page 127).
2. Find the `AuthMethod` element for which you want to define risk-based authentication settings.

   For example:

   ```
   <AuthMethod id="policyRBA">
     <Authenticator>
       <Policy/>
     </Authenticator>
   </AuthMethod>
   ```

3. Add an `EnableRBAWithIntelliTrust` element with either true or false value immediately after the closing Authenticator tag in the AuthMethod definition to which you want to add risk-based authentication.

   ```
   <AuthMethod id="policyRBA">
     <Authenticator>
       <Policy/>
     </Authenticator>

   <EnableRBAWithIntelliTrust>true/false</EnableRBAWithIntelliTrust>
   </AuthMethod>
   ```

4. Note:

Note:
If `<EnableRBAWithIntelliTrust>` element is not present, the default value is `set` to false. The RBA **RememberMe** checkbox appears based on the value of this setting.

Your `<EnableRBAWithIntelliTrust>` settings must be consistent with the Machine Authenticator settings in your Entrust Identity as a Service. For example, if Machine Authenticator is enabled in IDaaS but you configure `<EnableRBAWithIntelliTrust>` to `false` in ISAPI Filter, then the **RememberMe** checkbox does not appear, RBA authentication does not pass, and the user is always challenged.

If Machine Authenticator is disabled in IDaaS but you configure `<EnableRBAWithIntelliTrust>` to `true` in ISAPI Filter, then the **RememberMe** checkbox always appears and if user checks it, the machine will not get registered in Entrust Identity as a Service and the user is always challenged.

5 Save and close `IdentityGuardAuthAppConfiguration.xml` either manually using a text editor or using the Configuration Console (see"Configuring ISAPI Filter using the Configuration Console" on page 127).

You have completed the configuration of risk-based authentication. When you enable machine authentication, Entrust Identity as a Service stores machine information so that it can verify the machine later on by matching the information stored. See the *Entrust Identity as a Service Online Help* for more information about machine authentication.

To configure risk-based authentication with Entrust Identity Enterprise

1 Open the `IdentityGuardAuthAppConfiguration.xml` file either manually using a text editor or using the Configuration Console (see"Configuring ISAPI Filter using the Configuration Console" on page 127).

2 Find the `AuthMethod` element for which you want to define risk-based authentication settings.

For example:

```
<AuthMethod id="policyRBA">
  <Authenticator>
    <Policy/>
  </Authenticator>
</AuthMethod>
```

3 Add an `RBA` element immediately after the closing `Authenticator` tag in the `AuthMethod` definition to which you want to add risk-based authentication.

```
<AuthMethod id="policyRBA">
  <Authenticator>
    <Policy/>
  </Authenticator>
  <RBA>
    ...
  </RBA>
</AuthMethod>
```

4 Add child elements as needed to the `RBA` element, as described in the following sub-steps.

a Add a `SecurityLevel` element.

```
<RBA>
  <SecurityLevel>Normal</SecurityLevel>
  ...
</RBA>
```

`SecurityLevel` is a required element. You can set it to `Normal` or `Enhanced`. The exact meanings of `Normal` and `Enhanced` are defined in your Entrust Identity Enterprise RBA policy settings. See the *Entrust Identity Enterprise Administration Guide* for details.

b Add the `UseIP` element (optional) and set it to `true` if you want to pass the client IP address to Entrust Identity Enterprise for IP Geolocation analysis.

```
<RBA>
  <SecurityLevel>Normal</SecurityLevel>
  <UseIP>true</UseIP>
  ...
</RBA>
```

Setting the `UseIP` element to `false` is the same as leaving it out. You can configure the ISAPI Filter component for forwarding the client IP address to the authentication application. See "Configuring the ISAPI Filter for IP address validation" on page 194 for more information.

c Add an `RegisterMachine` element (optional), if you want to use machine authentication.

```
<RBA>
  <SecurityLevel>Normal</SecurityLevel>
  <UseIP>true</UseIP>
```

```
    <RegisterMachine>
      ...
    </RegisterMachine>
  </RBA>
```

d Add the following child elements to `RegisterMachine` as needed, and set their attributes.

- `UseMachineNonce` stores a random number (nonce) in Entrust Identity Enterprise, and in Flash or a cookie on the client machine. See "Client-side storage of nonces" on page 192.

  This element contains four attributes:

  – `enabled` is a Boolean value that determines if this element is used.
  – `cookieName` is a string value that equals the name of the cookie stored in the client browser.
  – `cookieDomain` is a string value that indicates the domain the cookie must be set against.
  – `cookieLifetime` is an integer value that determines the number of days for which the cookie must be stored in the client browser. Setting it to a smaller number means that the user is forced to perform second-factor authentication more frequently.
- `UseSequenceNonce` stores a sequential number (nonce) in Entrust Identity Enterprise, and in Flash or a cookie on the client machine. See "Client-side storage of nonces" on page 192. This number is updated with each subsequent login and is therefore constantly changing.

Note:
`UseSequenceNonce` cannot be used unless either `UseMachineNonce` or `UseAppData` is also enabled.

This element contains four attributes:

– `enabled` is a Boolean value that determines if this element is used.
– `cookieName` is a string value that equals the name of the cookie stored in the client browser.
– `cookieDomain` is a string value that indicates the domain the cookie must be set against.
– `cookieLifetime` is an integer value that determines the number of days for which the cookie must be stored in the client browser. Setting it

to a smaller number means that the user is forced to log in more frequently.

- `UseAppData` reads information from the user's browser to identify the client machine. See also "Application data collected" on page 193.

  The example below includes all the `RegisterMachine` child elements described above. The only required element is `SecurityLevel`; all of the other elements are optional.

```
<RBA>
  <SecurityLevel>Normal</SecurityLevel>
  <UseIP>true</UseIP>
  <RegisterMachine>
    <UseMachineNonce enabled="true" cookieName="machineNonce"
cookieDomain="mydomain.com" cookieLifetime="365" />
    <UseSequenceNonce enabled="true" cookieName="sequenceNonce"
cookieDomain="mydomain.com" cookieLifetime="365" />
    <UseAppData>true</UseAppData>
  </RegisterMachine>
</RBA>
```

- `DeviceFingerprintData` is a value calculated from a set of device-specific attributes that are collected by your client application. (To collect these attributes, design your client application using the Entrust Identity Enterprise Device Fingerprint SDK introduced in Entrust IdentityGuard release 12.0.) In the **Policies** pages of the Administration interface, you can edit the list of attributes used in the calculation, and assign them relative weights based on your perception of the amount of risk they represent. For more information, see "Configure the device fingerprint policies" in the *Entrust Identity Enterprise Administration Guide*.

  The example below includes all the `DeviceFingerprintData` element described above. The only required element is `SecurityLevel`; all of the other elements are optional.

```
<RBA>
  <SecurityLevel>Normal</SecurityLevel>
  <UseIP>true</UseIP>
  <RegisterMachine>
    <UseMachineNonce enabled="true"
cookieName="machineNonce" cookieDomain="mydomain.com"
cookieLifetime="365" />
    <UseSequenceNonce enabled="true"
cookieName="sequenceNonce" cookieDomain="mydomain.com"
cookieLifetime="365" />
```

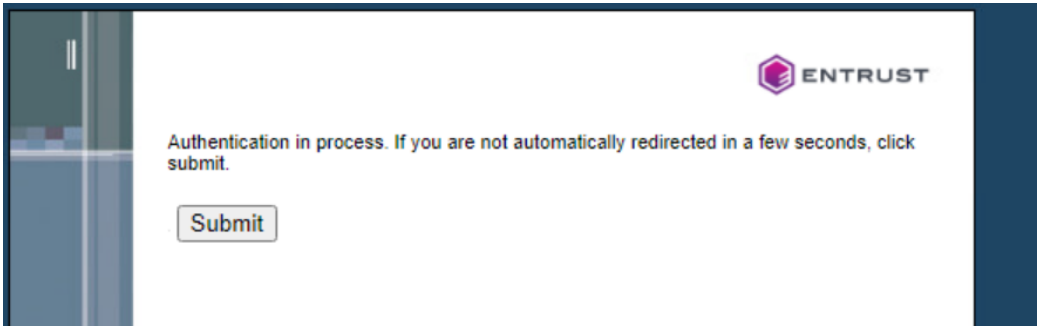
```
        <UseAppData>true</UseAppData>
        <UseDeviceFingerprintData>true</UseDeviceFingerprintData>
      </RegisterMachine>
    </RBA>
```

Note:
Your `<RBA>` settings should be consistent with the RBA settings in your Entrust Identity Enterprise policy; otherwise, risk-based authentication does not work properly. For example, if policy requires sequence nonce but you configure the ISAPI Filter to not require it, then RBA authentication does not pass, and the user is always challenged.

5 Save and close `IdentityGuardAuthAppConfiguration.xml` either manually using a text editor or using the Configuration Console (see"Configuring ISAPI Filter using the Configuration Console" on page 127).

Redirection page

When risk-based authentication is enabled, you may temporarily see a redirection page (as shown below) before the second-factor authentication challenge appears.



This page is displayed while your machine settings are validated. If a challenge is required, it automatically redirects the browser to the second-factor challenge page.

Sometimes the redirection does not take place, and the user must click **Submit** to proceed to the second-factor challenge page. This can happen if JavaScript and meta refresh are both disabled.

To enable the redirection, ensure that JavaScript and meta refresh are both enabled in your browser. If you have JavaScript enabled, then redirection works even if meta refresh is disabled.

To enable JavaScript and meta refresh in Microsoft Internet Explorer

1. Select **Tools** > **Internet Options**.
2. In the **Internet Options** dialog box, click the **Security** tab.
3. Click **Custom level** to open the **Security Settings** dialog box.
4. Scroll down to locate **Allow META REFRESH**. Ensure that **Enable** is selected.
5. Scroll down to locate **Active scripting**. Ensure that **Enable** is selected.
6. Click **OK** twice to save the settings and exit.

To enable JavaScript and meta refresh in Mozilla Firefox

1. Select **Tools** > **Options**.
2. Select the **Content** tab. Ensure that **Enable JavaScript** is selected.
3. Click **OK** to save the settings and exit.

See also "User not automatically redirected from machine authentication page" on page 296.

Client-side storage of nonces

There are two ways to store the machine nonce and the sequence nonce on the user's computer: cookies or an Adobe Flash object. Whether to store as cookies or flash is decided using the following tests:

- If Flash is enabled on the browser, the nonces are stored in a Flash object.
- If cookies are enabled on the browser, the nonces are stored as cookies.
- If none of this can be checked (because JavaScript is disabled), then cookies are used.
- If both cookies and Flash are disabled, the user is always challenged.

Server-side storage of machine secrets

If you are using application data, you must increase the Entrust Identity Enterprise policy settings for **Maximum Number of Machine Secrets** and **Total Maximum Size in Kilobytes**.

The approximate size of a machine secret, when application data, machine nonce, and sequence nonce are all enabled, is 700-900 bytes (depending on the user's browser). This means that you must allocate this amount of storage for each machine secret (for each user), and adjust the Entrust Identity Enterprise machine secret policy settings as necessary.

For example, if users need to be able to register up to 5 machines each, then their **Maximum Number of Machine Secrets** policy must be set to 5, and the **Total**

Maximum Size in Kilobytes policy should be set to 5 KB (5 * 900 bytes, rounded up to the nearest KB).

Application data collected

When `UseAppData` is set to `true`, the solution reads information from the user's browser to identify the client machine. The following list shows some of the data collected.

- User language
- Browser
- Operating system
- Screen width
- Screen height
- Screen color depth
- Plug-ins installed; for example: QuickTime, RealPlayer, Windows Media Player, Acrobat Reader
- Flash version
- Java enabled
- Number of plug-ins
- MIME types supported
- Number of MIME types

Note:
The RBA settings in `IdentityGuardAuthAppConfiguration.xml` must match the Entrust Identity Enterprise risk-based authentication policies. For instance, if the Entrust Identity Enterprise policy is set to require machine nonces, then `useMachineNonce` must be set to `true` in `IdentityGuardAuthAppConfiguration.xml`.

Note:
Generally, 25 or more application data elements are collected. If you set the Entrust Identity Enterprise policy setting for **Application Data Elements Required** to a number other than zero, you must set it to a number lower than 25 (approximately). Otherwise machine authentication may fail because not enough application data elements are provided.

Configuring the ISAPI Filter for IP address validation

You can configure the remote IP header if you enabled IP address validation for risk-based authentication (RBA) for any of your authentication methods, by one of the following methods:

- Select **Validate client IP address** in the **Authentication Application Setup** dialog box during installation.
- Set the `UseIP` element to `true` in the configuration file for the authentication application, after the installation is complete.

  See "Configuring risk-based authentication" on page 185 for details.

If you do not have RBA enabled for any of your authentication methods, you do not need to configure the remote IP header.

How IP address validation works

IP address validation can be enabled, as part of risk-based authentication. In this situation, the IP address of the user's machine must be available to the ISAPI Filter in order for IP address and geolocation validation to occur. By default, the ISAPI Filter uses the source IP address of the incoming HTTP request. If your network has any intermediary devices (such as a gateway or proxy), which effectively change the source IP address of the incoming HTTP request when forwarding the request to the ISAPI Filter, then the solution uses the IP address of the intermediary device and not the IP address of the client. If your gateway or proxy server supports passing the end-user IP address through as a secure header, then you can configure the ISAPI Filter to use this header instead.

The ISAPI Filter provides a `<RemoteIPHeader>` setting in the filter configuration file that allows you to specify how the ISAPI Filter should determine the IP address of the client. By default, the `<RemoteIPHeader>` setting is not specified and the filter uses the source IP address of the incoming HTTP request. Alternatively, you can specify an HTTP header value for this setting to instruct the ISAPI Filter to look for the remote IP address of the client in this particular header.

You must ensure that the HTTP header sent by your gateway or proxy server to the ISAPI Filter is secure. This is to prevent IP address spoofing, which would allow a malicious third party to enter your secure network. Ensuring that this header is trusted and secure is critical to reliable IP address and geolocation validation. To ensure that your gateway or proxy server passes a trusted header to the filter, it must first remove the header (if it is present) in the incoming request. It must then add its own header, containing the end-user IP address, to the forwarded request sent to the filter.

Attention:
You must be sure you understand the risks associated with this header before making the configuration changes described in the following section. The IP address specified in this header is used for risk-based authentication, so a misconfiguration of this header could allow malicious users to bypass IP address verification such as IP Blacklist, Expected Locations, User Location History, and Velocity checks.

Configuring the remote IP header

You can configure the filter to read the incoming header and pass on the correct IP address to the authentication application by editing the filter configuration file.

To configure the remote IP header

1 Open the `IdentityGuardFilterConfiguration.xml` file either manually using a text editor or using the Configuration Console (see"Configuring ISAPI Filter using the Configuration Console" on page 127).

2 Find the `Headers` element. For example:

```
<Headers>
  ...
</Headers>
```

3 Add a `RemoteIPHeader` element within the `Headers` element. For example:

```
<Headers>
  <RemoteIPHeader />
  ...
</Headers>
```

4 Give it a `name` attribute and value. For example:

```
<Headers>
  <RemoteIPHeader name="SECURE_REMOTE_IP" />
  ...
</Headers>
```

The "`SECURE_REMOTE_IP`" value for the `name` attribute is an example. You must set it to the header being used by your gateway or proxy server.

5 Save and close `IdentityGuardFilterConfiguration.xml`.

Forcing login

The ISAPI Filter installation automatically configures forced login behavior correctly when it is installed on an IIS server and you select OWA as the first-factor authentication type.

You need to configure forced login only if you:

- change to OWA first-factor login after installation
- modify your OWA configuration after installation

The purpose of the forced login feature is to force the user to log in to the correct ISAPI OWA first-factor login page.

When an authenticated user logs out, the user is presented with the default OWA logoff URL. This URL redirects them to the default OWA login page, instead of the ISAPI OWA first-factor login page. Having users go to the OWA login page is undesirable because they can log in without passing through the ISAPI OWA login process. To prevent this behavior, and force user to login at the ISAPI OWA first-factor login page, follow the instructions below in "To configure forced login".

The user can be forced to log in to the correct ISAPI OWA first-factor login page by using the `forcelogin` attribute of the `URL` element within `ProtectedURLs`. This is only supported for the Outlook Web Access (OWA) logoff URL on an IIS server. Setting the value of `forcelogin` to "`true`" ensures that the correct ISAPI OWA first-factor login page is shown when trying to access the URL, even if the user has previously logged in.

This attribute is automatically set by the installer if you select OWA as the first-factor authentication type. The default behavior (when this attribute is missing) is to not show the login page after the user logs out from OWA.

To configure forced login

1. Open the `IdentityGuardFilterConfiguration.xml` file either manually using a text editor or using the Configuration Console (see"Configuring ISAPI Filter using the Configuration Console" on page 127).
2. Find the `ProtectedURLs` element under the `ProtectedHost` element.

   For example:

```
<ProtectedHost host="" port="80,443">
...
  <ProtectedURLs authlevel="1" firstfactorid="owa">
    <URL authlevel="1">/privatefolder/*</URL>
  </ProtectedURLs>
  <UnprotectedURLs/>
</ProtectedHost>
```

3 Under ProtectedURL, add a `URL` element with the OWA logoff URL.

Example:

```
<URL authlevel="1">/privatefolder/*</URL>
<URL>/owa/auth/logon.aspx</URL>
```

4 Add the `forcelogin` attribute and set it to `true`.

Example:

```
<URL authlevel="1">/privatefolder/*</URL>
<URL forcelogin="true">/owa/auth/logon.aspx</URL>
```

5 Save and close `IdentityGuardFilterConfiguration.xml`.

This ensures that when the logoff page, `/owa/auth/logon.aspx`, is accessed, the login page is always shown, regardless of whether a login was previously performed.

Configuring step-up authentication

The ISAPI Filter allows you to provide step-up authentication. Using the step-up authentication mechanism, you can configure different levels of authentication for different protected resources. See “Step-up” on page 24 for more information about step-up authentication.

You can configure step-up authentication by editing the filter configuration file.

To configure step-up authentication

1. Open the `IdentityGuardFilterConfiguration.xml` file either manually using a text editor or using the Configuration Console (see“Configuring ISAPI Filter using the Configuration Console” on page 127).
2. Find the `ProtectedURLs` element under the `ProtectedHost` element. For example:

```
<ProtectedHost host="" port="80,443">
  ...
  <ProtectedURLs authlevel="1" firstfactorid="iwa">
    ...
  </ProtectedURLs>
</ProtectedHost>
```

3. Enter a `URL` element for your basic protected resource, under the `ProtectedURLs` element, and optionally, assign an `authlevel` to it. If the `authlevel` attribute is not specified, then it is inherited from the `authlevel` in `<ProtectedURLs>`. For example:

```
<ProtectedHost host="" port="80,443">
  ...
  <ProtectedURLs authlevel="1" firstfactorid="iwa">
    <URL authlevel="1">/PersonalBanking/*</URL>
  </ProtectedURLs>
  <UnprotectedURLs/>
</ProtectedHost>
```

   In the example above, `/PersonalBanking/*` has been assigned an authentication level of 1.
4. Enter another `URL` element for your step-up level protected resource and, if you want it to be different than the authentication level you have already specified, assign an `authlevel` to it.

Note:
If you do not specify an `authlevel` attribute, the level is inherited from the `authlevel` in `<ProtectedURLs>`.

For example:

```
<ProtectedHost host="" port="80,443">
  ...
  <ProtectedURLs authlevel="1" firstfactorid="iwa">
    <URL authlevel="1">/PersonalBanking/*</URL>
    <URL authlevel="2">/StockTrading/*</URL>
  </ProtectedURLs>
  <UnprotectedURLs/>
</ProtectedHost>
```

5 Add an `UnprotectedURLs` element after the closing tag of the `ProtectedURLs` element, and define any URLs that you want to exclude from protection. See "Adding or removing protected and unprotected URLs" on page 226 for more information about excluding URLs from protection.

Even if you do not want to unprotect any URLs, you must still include an empty `UnprotectedURLs` element. For example:

```
<ProtectedHost host="" port="80,443">
  ...
  <ProtectedURLs authlevel="1" firstfactorid="iwa">
    <URL authlevel="1">/PersonalBanking/*</URL>
    <URL authlevel="2">/StockTrading/*</URL>
  </ProtectedURLs>
  <UnprotectedURLs/>
</ProtectedHost>
```

6 Save and close `IdentityGuardFilterConfiguration.xml`.

In the example above, `/StockTrading/*` has been assigned a stricter `authlevel` of `2`. A user who has successfully logged in to `/PersonalBanking/*`, and then tries to access `/StockTrading/*`, is presented with a Level 2 authentication challenge.

Note:
For each authentication level you specify in the filter configuration, you must have a matching `<Level>` setting defined in the authentication application. See “Defining an authentication method” on page 164.

Configuring anonymous challenge authentication with Identity Enterprise

The ISAPI Filter solution supports anonymous challenge authentication. With this feature enabled, the user can either login using forms-based authentication by supplying a username and group name, or choose the **Use Anonymous Login** option. When the anonymous login option is selected, the user is presented with an anonymous challenge.

The following is required to use this feature:

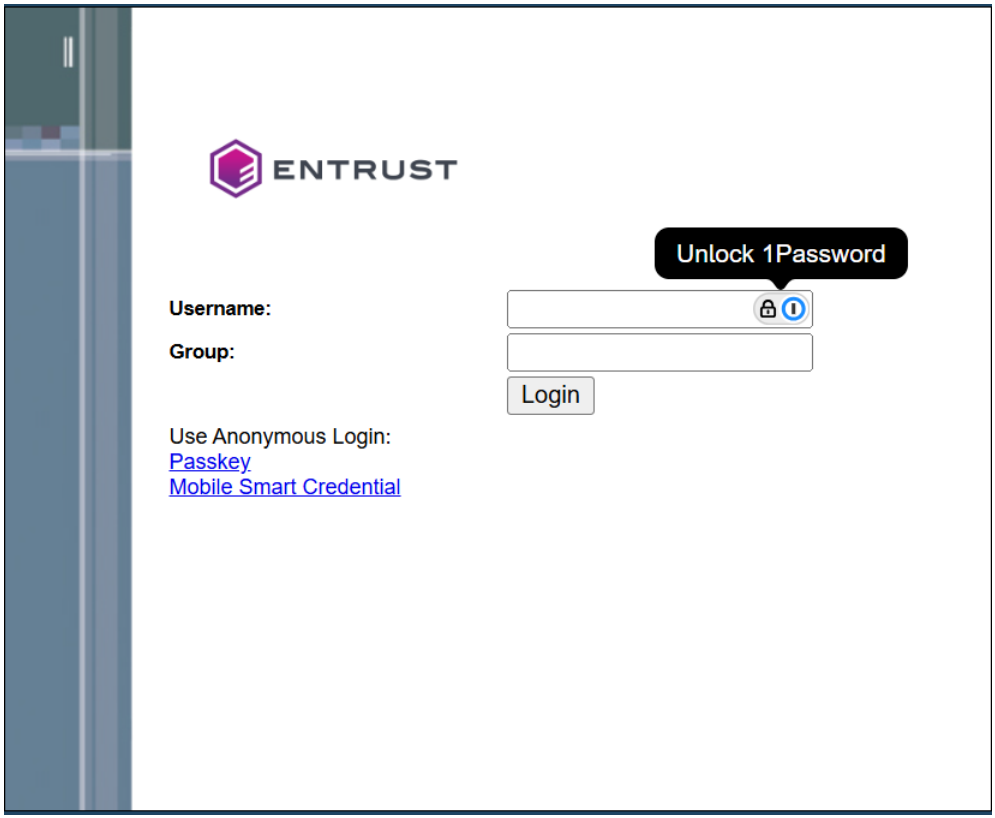
- `firstfactorid` must be set to `formsBased` (see "Configuring first-factor authentication" on page 159).
- MobileSC must be configured in the ISAPI Filter solution (see "To configure mobile smart credential authentication" on page 172).
- Users must have the Entrust Identity Mobile Smart Credential app installed on their mobile device.

Note:
Second factor authentication and RBA are not supported with anonymous challenge.

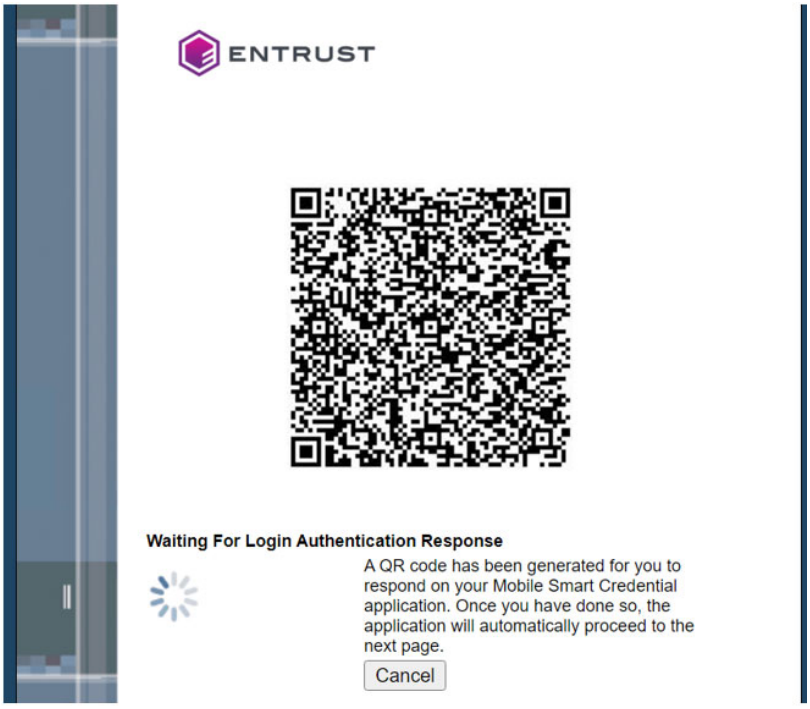
Anonymous challenge authentication process

To login using anonymous authentication with mobile smart credential

1 Access a URL protected by the ISAPI Filter solution. The Entrust authentication page appears.

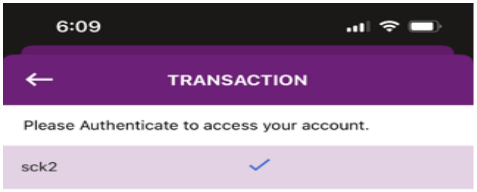


2 Click **Use Anonymous Login**. An anonymous challenge (QR code) page appears.



3 On your mobile device, open the Entrust Identity Mobile Smart Credential app and select **Scan QR Code** in the app.

4 Using the mobile device, scan the QR code. You are presented with the **Authentication Challenge**.



5 Select the user from the **Select Identity** drop-down list and then click **Confirm**.

You are prompted to reconfirm.

6 Click **Yes** on the prompt.

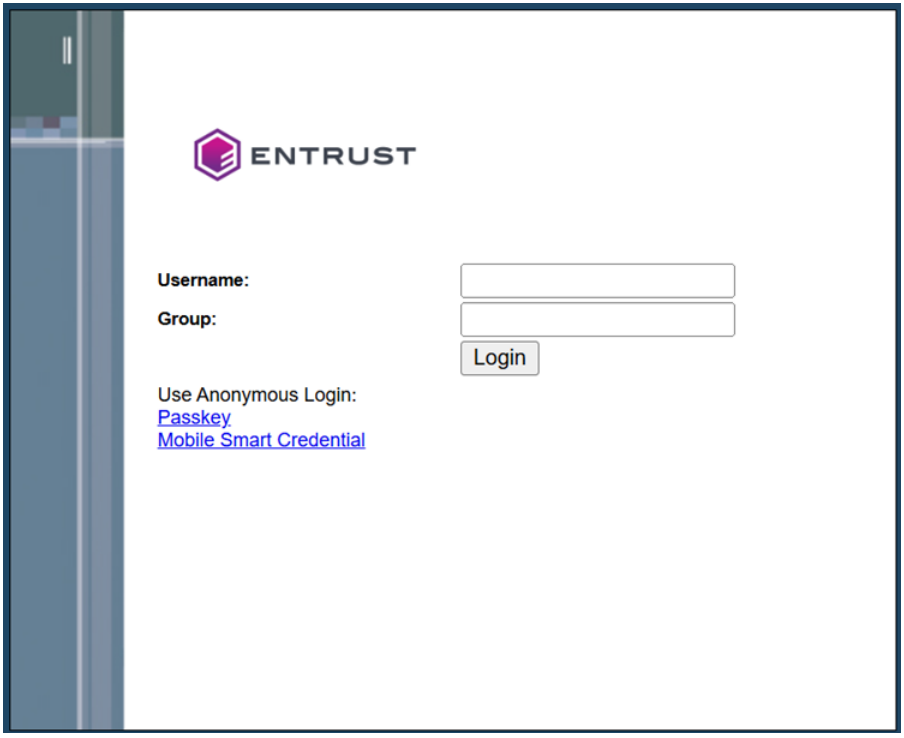
7 Enter the MobileSC **PIN** and then click **OK** to allow the authentication to proceed.

The protected resource is now available.

Log in using anonymous challenge with passkey

To log in using anonymous challenge with passkey

1 Access a URL protected by the ISAPI Filter solution. The **Entrust authentication** page appears.



2 Click **Passkey**. An anonymous Passkey challenge page appears.

3 Follow the screen prompts to complete Passkey/FIDO2 authentication.

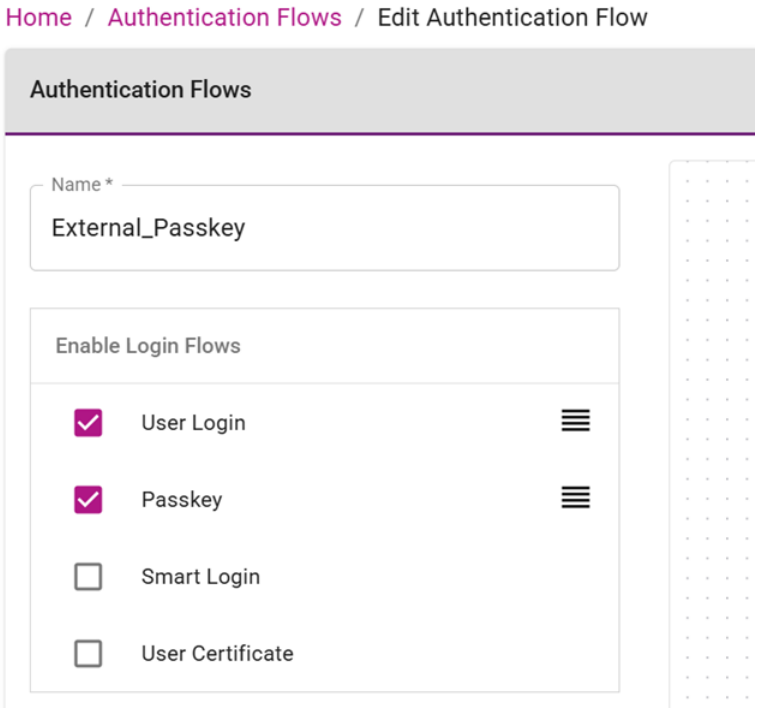
4 After successful authentication, the user is logged into the protected resource.

Configuring anonymous challenge authentication with Identity as a Service

The ISAPI Filter solution supports anonymous challenge authentication. With this feature enabled, the user can either login using forms-based authentication by providing a username and group name or choose the **Use Anonymous Login** option. When the anonymous login option is selected, the user is presented with an anonymous challenge.

The following is required to use this feature:

- `Firstfactorid` must be set to formsBased (see “Configuring first-factor authentication” on page 159).
- Create a custom user login Authentication Flow to enable Passkey/FIDO2 under Enable Login Flow for anonymous authentication. See Create authentication flows in the *IDaaS Administrator Help*.

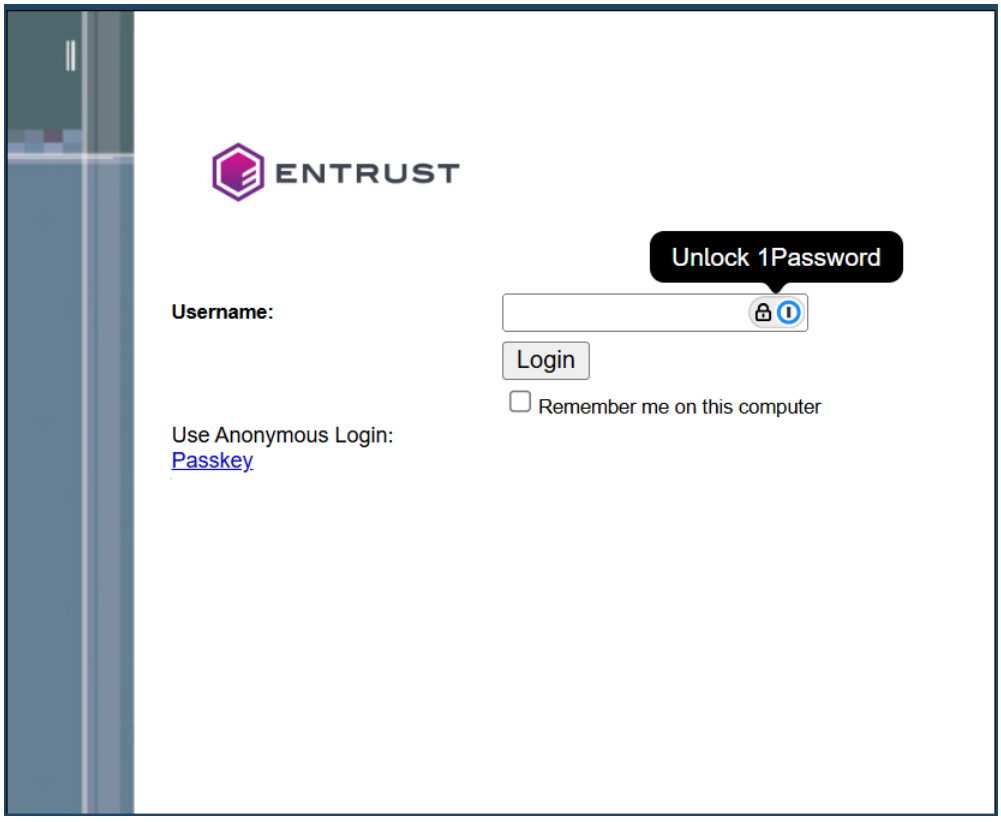


Anonymous challenge authentication process

To login using anonymous authentication

1 Access a URL protected by the ISAPI Filter solution. The **Entrust authentication** page appears.

2 Click **Passkey**. An anonymous Passkey challenge page appears.



3 Follow the screen prompts to complete Passkey/FIDO2 authentication.

4 After successful authentication, the user is logged into the protected resource.

Configuring user access group authorization

The ISAPI Filter solution supports user access group authorization. This features enables you to associate user access groups with protected URLs assigned to Entrust Identity Enterprise access groups.

The ISAPI filter matches the protected URLs `accessgroup` with the ones returned by the Authentication Application. If the `accessgroup` matches Entrust Identity Enterprise access groups, the user is redirected to the requested URL.

This section contains the following topics:

Configuring user access group

After installing the ISAPI filter, you must set the `accessgroup` elements in the `IdentityGuardFilterConfiguration.xml file` to use this feature.

To configure the User Access Group

1. Open `IdentityGuardFilterConfiguration.xml` for editing file either manually using a text editor or using the Configuration Console (see"Configuring ISAPI Filter using the Configuration Console" on page 127).
2. Scroll to the `<ProtectedURLs>` definition statement.
3. Edit the `ProtectedURLs` definition by including the `authlevel` value you added and add new attribute, `accessgroup` which you required. The section should like look like similar to the following example:

```
<ProtectedURLs>.

<ProtectedURLs authlevel="1" accessgroup="group_2"
firstfactorid="formsBased" >

<URL authlevel="1"
accessgroup="group_1">/protected1/resource1.aspx</URL>

<URL authlevel="1">/protected2/resource2.aspx</URL>

</ProtectedURLs>
```

4. Save and close `IdentityGuardFilterConfiguration.xml`.
5. Restart the applicable Web service for the changes to take effect. For instructions, see "Restarting services after changing configuration files" on page 141.

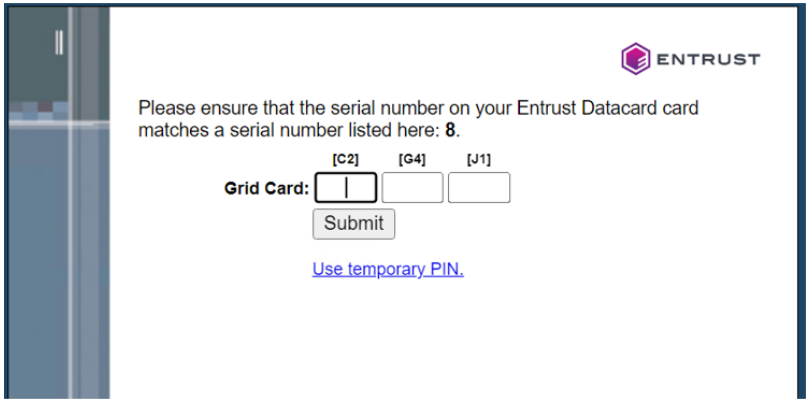
Note:
You can include multiple access groups separated by a comma. For example, `<ProtectedURLs authlevel="1" firstfactorid="formsBased" accessgroup="group_1,group_2,group_3">`

If you include multiple access groups, if the `accessgroup` is identified at the `<URL>` level, then it will override the `accessgroup` identified at `<ProtectedURLs>`. In the above example, for `/protected1/resource1.aspx` the `accessgroup` is `group_1` and for `/protected2/resource2.aspx`, `accessgroup` is `group_2`.

User access group authorization process

To login using user access group authorization

1. Access a URL protected by the ISAPI Filter solution. The Entrust authentication page appears.
2. Enter your Entrust Identity Enterprise **Username** and **Group** (optional).
3. Click **Login**. The Entrust Identity Enterprise second-factor challenge appears. This example shows grid authentication.



4. Enter the correct response to the challenge. You are redirected to the protected URL you requested at the start of the test.

Note:
If the ISAPI Filter `accessgroup` does not match the Entrust Identity Enterprise Access Group(s) and validation fails, a remote access errorappears and the web page is blocked.

Note:
If the Administrator does not specify the access groups at the `<ProtectedURLs>` level and sets the access group names in the Entrust Identity Enterprise policy, the ISAPI filter triggers the group access validation and users are allowed to authenticate, but they are denied access to the resources.

Configuring alternate authenticators

If users do not have their primary authenticator available, you can configure this integration to offer them an alternate authentication method.

To have a link for an alternate authenticator appear on the login screen for a given user, that authenticator must:

- be configured for use in the policy for the Entrust Identity Enterprise group to which the user belongs
- be an authenticator that the user possesses (for example a grid card, knowledge of the answers to questions, or a mobile smart credential)
- be configured as an alternate authentication method for a given `<AuthenticationMethod>` in the `IdentityGuardAuthAppConfiguration.xml` file

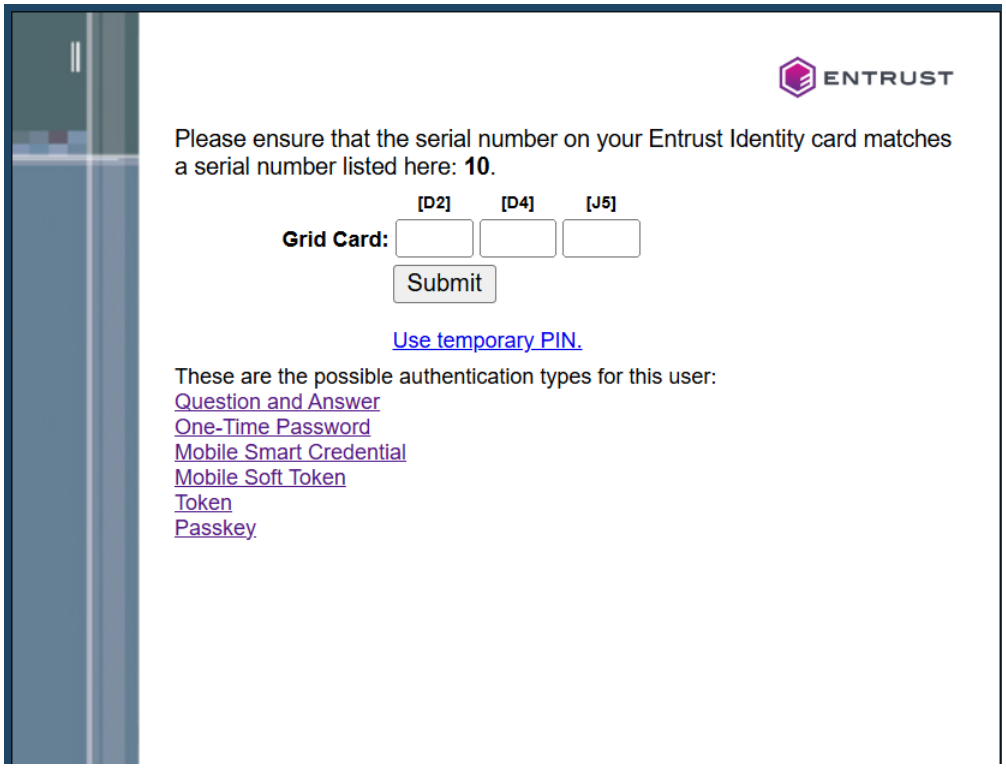
You can configure the ISAPI Filter solution to display alternative second-factor authenticators on the second-factor authentication page (see Figure 5 on page 212). Users can select an alternative if they do not have their primary authenticator.

The authenticators that are supported as alternatives are:

- grid
- knowledge-based Q&A
- one-time password (OTP)
- MobileSC
- MobileST
- token
- Passkey/FIDO2 token

For example, Q&A will be visible as an alternative even if the user has not created Q&A answers yet, if you allowed Q&A in your policy and it is configured in the configuration file.

Figure 5: Alternative authenticators



To enable alternative authenticators

1. Open the `IdentityGuardAuthAppConfiguration.xml` file either manually using a text editor or using the Configuration Console (see"Configuring ISAPI Filter using the Configuration Console" on page 127).
2. Locate the authenticator that will be an alternative authenticator. For example, locate this XML block:

```
<AuthMethod id="gridAuth">
  <Authenticator>
    <Grid />
  </Authenticator>
</AuthMethod>
```

3. Add the following text, in bold:

```
<AuthMethod id="gridAuth">
  <Authenticator>
    <Grid Alternate="true"/>
```

```
<Token Alternate="true"/>
<OTP Alternate="true" />
  <AllowManualDelivery>false</AllowManualDelivery>
</OTP>
<KB Alternate="false">
  <OverrideKBChallengeSize size="4" />
  <MaskAnswers>false</MaskAnswers>
</KB>
<MobileSC Alternate="true"/>
<Passkey Alternate="true"/>
</Authenticator>
</AuthMethod>
```

where **`Alternate="true"`** indicates that the authenticator must be listed as a link below the primary authenticator, if it is not already displayed as the primary authenticator.

4 Save and close `IdentityGuardFilterConfiguration.xml`.

Configuring alternate authenticators with Mobile soft token (TVS)

The following procedures provide examples of configuring alternate authenticators with mobile soft token (TVS) authentication.

Example implementation with fallbacktoClassic set to true

To enable MobileST and alternative authenticators with fallbacktoClassic set to true

1 Open `IdentityGuardAuthAppConfiguration.xml` for editing either manually using a text editor or using the Configuration Console (see"Configuring ISAPI Filter using the Configuration Console" on page 127).

2 Locate this XML block:

```
<AuthMethod id="MobileST">
  <Authenticator>
    <MobileST/>
  </Authenticator>
</AuthMethod>
```

Note:
See “Configuring mobile soft token (TVS) authentication” on page 174 if you have not defined mobile soft token authentication.

3 Add the following text, in bold:

```
    <AuthMethod id="MobileST">
      <Authenticator>
        <MobileST polling interval="15"
fallbacktoClassic="true"/>
        <Token Alternate="true"/>
         <Grid Alternate="true"/>
      </Authenticator>
    </AuthMethod>
```

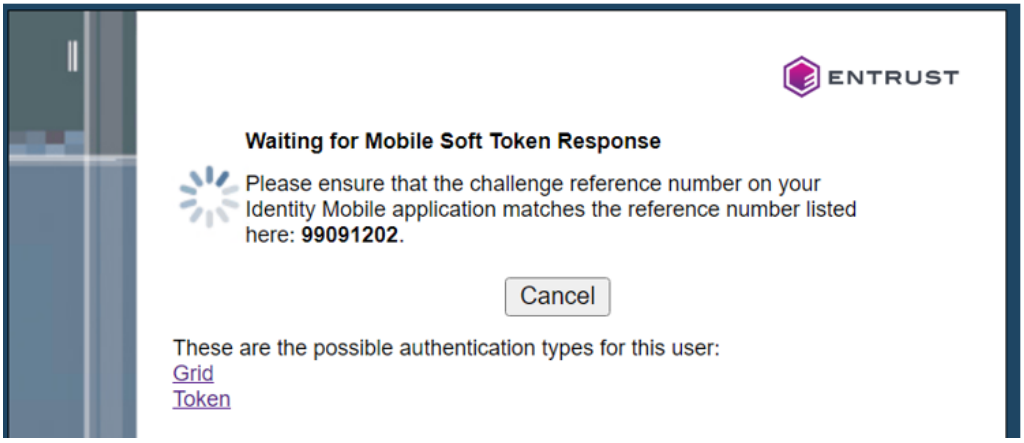
The `polling interval` indicates how long the authentication application waits, in seconds, before calling Entrust Identity Enterprise Server after receiving a request from the user’s browser to try and authenticate the mobile soft token (TVS) authentication challenge.

When `fallbacktoClassic="true"` users can manually enter the OTP if they do not have an Internet connection (this is known as classic mode or offline activated Token).

`Alternate="true"` indicates that the authenticator must be listed as a link below the primary authenticator. If it is not, it is displayed as the primary authenticator.

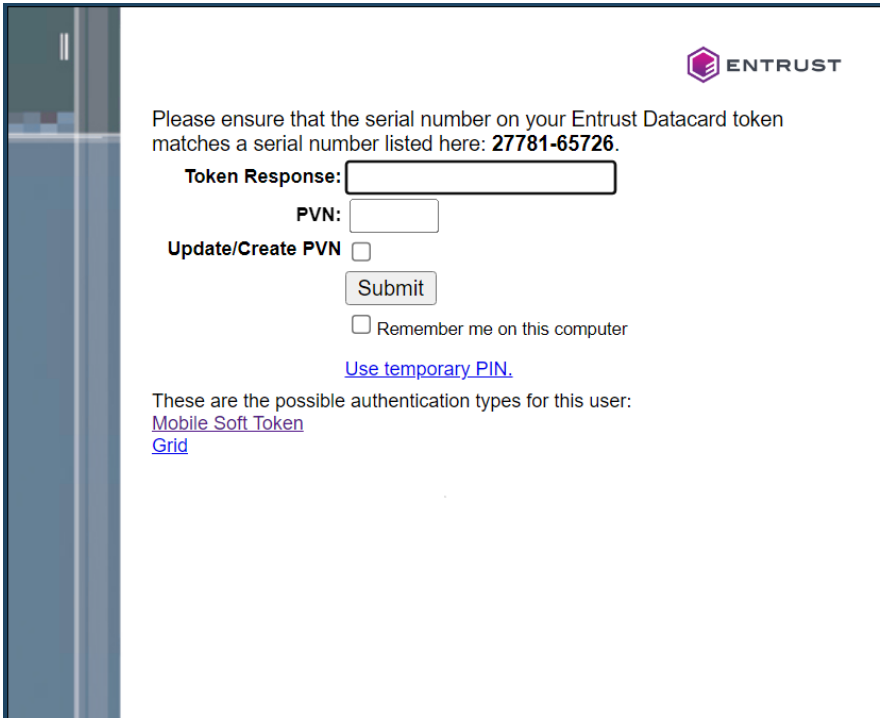
4 Save and close `IdentityGuardAuthAppConfiguration.xml`.

With this configuration, an online activated user sees the following screen:

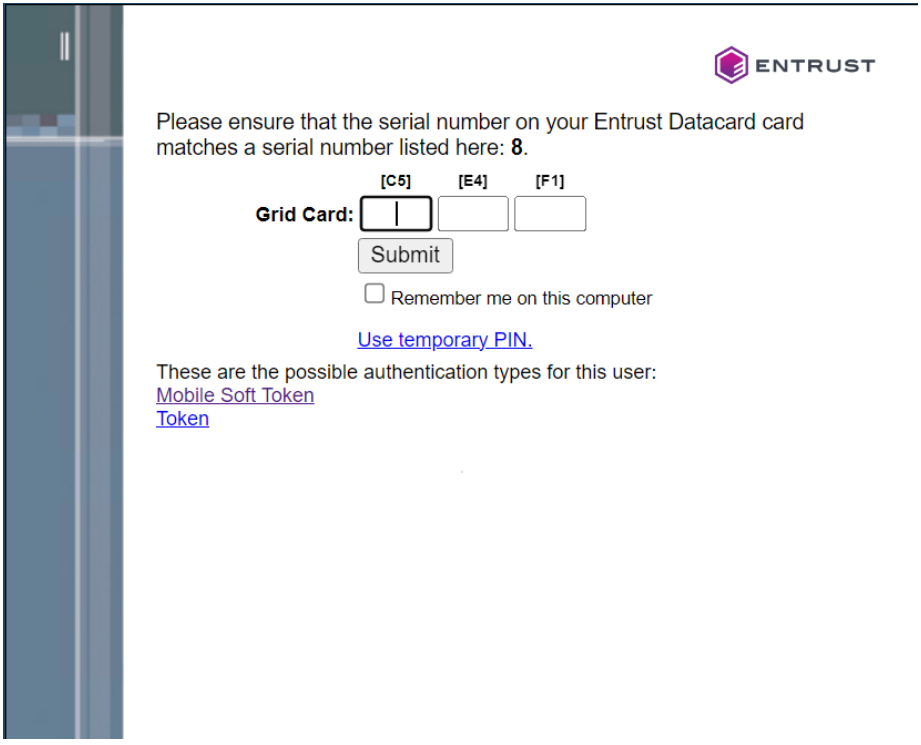


The user then clicks the **Token** link and is redirected to the TOKENRO page.

If the activated user is offline, the user sees the following screen:



If a user does not have a soft token, a grid challenge is presented as shown in the following screen:



If the user clicks the **Token** link, the following message appears: “Your account has not been activated.”

Example implementation with fallbacktoClassic set to false

To enable MobileST and alternative authenticators with fallbacktoClassic set to false

1 Open `IdentityGuardAuthAppConfiguration.xml` for editing either manually using a text editor or using the Configuration Console (see“Configuring ISAPI Filter using the Configuration Console” on page 127).

2 Locate this XML block:

```
<AuthMethod id=”MobileST”>
  <Authenticator>
    <MobileST/>
  </Authenticator>
```

```
</AuthMethod>
```

Note:
See “Configuring mobile soft token (TVS) authentication” on page 174 if you have not defined mobile soft token authentication.

3 Add the following text, in bold:

```
<AuthMethod id=”MobileST”>
  <Authenticator>
    <MobileST pollinginterval=”15”
fallbacktoClassic=”false”/>
      <Token Alternate=”true”/>
       <Grid Alternate=”true”/>
  </Authenticator>
</AuthMethod>
```

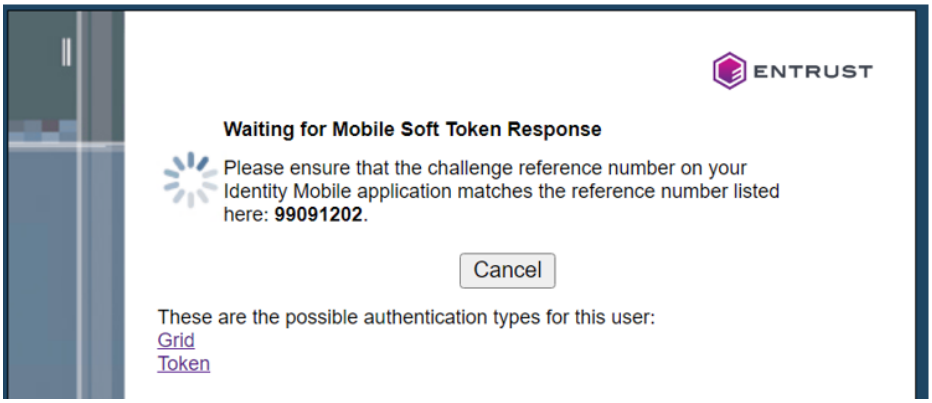
The `pollinginterval` indicates how long the authentication application waits, in seconds, before calling Entrust Identity Enterprise Server after receiving a request from the user’s browser to try and authenticate the mobile soft token (TVS) authentication challenge.

`When fallbacktoClassic=”false”` only mobile soft token authentication using online mode is permitted (by which users select **Confirm** in the app to authenticate).

`Alternate=”true”` indicates that the authenticator must be listed as a link below the primary authenticator. If it is not, it is displayed as the primary authenticator

4 Save and close `IdentityGuardAuthAppConfiguration.xml`.

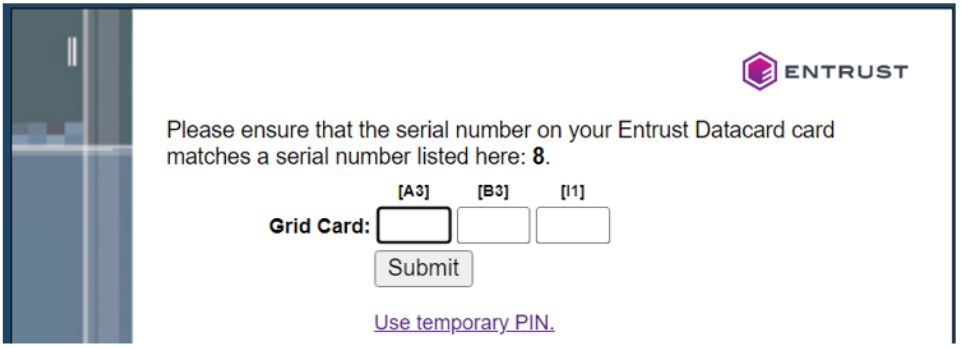
With this configuration, an online activated user sees the following screen:



If the activated user is offline, the user sees the following message:

```
Your account has not been activated to user second-factor
authentication. Please contact your administrator.
```

If a user does not have a soft token, a grid challenge is presented as shown in the following screen:



If the user clicks the **Token** link, the following message appears: “Your account has not been activated.”

Using personal verification numbers (PVN)

You can strengthen your second-factor authentication by requiring that users enter a numeric personal verification number (PVN) with their second-factor authenticator. PVNs can be used with grid, token, OTP, and temporary PIN authentication.

The characteristics of PVN use in your organization are defined and enforced through Entrust Identity Enterprise policy. Policies include minimum PVN length and whether a PVN is required or optional.

There is no configuration necessary in the ISAPI Filter solution to activate the use of PVNs; all of the configuration is done in Entrust Identity Enterprise. See the *Entrust Identity Enterprise Administration Guide* for information about PVN policies.

The ISAPI Filter solution supports PVN entry, PVN creation, and PVN change. Both PVN creation and PVN change can be initiated by the system or by the user.

If authentication requires a PVN, but the user does not yet have one, ISAPI Filter solution allows the user to create a PVN.

Attention:
For users authenticating with Authentify OTP, a PVN is mandatory. If the user does not have a PVN, the Create PVN screen appears, but the Authentify OTP is not delivered. The user must have a PVN before the Authentify delivery method is used.

Configuring external authentication

Use one of the following procedures in this section to configure first-factor authentication to Microsoft Active Directory using LDAP or Kerberos.

- “To configure external authentication for two-step authentication” on page 220
- “To configure external authentication for one-step authentication” on page 221

To complete the configuration, you must also configure external authentication on your Entrust Identity Enterprise Server. For detailed instructions, see the *Entrust Identity Enterprise Administration Guide*.

To configure external authentication for two-step authentication

1 On the system where the ISAPI Filter 13.0 is installed, open the `IdentityGuardLogin.aspx.cs` file. Its default location is

```
C:\Program Files
\Entrust\Identity\WinIS\webapp\IdentityGuardAuth
```

2 Locate the following line:

```
AuthenticationType authType = AuthenticationType.PASSWORD;
```

This line is present in the method `protected void Page_Load()`.

3 Change `AuthenticationType.PASSWORD` to `AuthenticationType.EXTERNAL`. For example, change the PASSWORD line as shown below:

```
// Setting the desired AuthenticationType.
// If you want to configure for External Authentication
// set authtype to AuthenticationType.EXTERNAL.
AuthenticationType authType = AuthenticationType.PASSWORD;
```

to

```
AuthenticationType authType = AuthenticationType.EXTERNAL;
```

Attention:
This text is case sensitive.

4 Save and close `IdentityGuardLogin.aspx.cs`.

5 Restart the World Wide Web Publishing Service.

To configure external authentication for one-step authentication

1 On the system where the ISAPI Filter 13.0 is installed, open the `IdentityGuardOneStepTokenFormsLogin.aspx.cs` file. Its default location is

```
C:\Program Files
\Entrust\Identity\WinIS\webapp\IdentityGuardAuth
```

2 Locate the following line:

```
AuthenticationType authType = AuthenticationType.PASSWORD;
```

This line is present in the `method protected void Page_Load()`.

3 Change `AuthenticationType.PASSWORD` to `AuthenticationType.EXTERNAL`. For example, change the PASSWORD line as shown below:

```
// Setting the desired AuthenticationType.
// If you want to configure for External Authentication
// set authtype to AuthenticationType.EXTERNAL.
AuthenticationType authType = AuthenticationType.PASSWORD;
```

to

```
AuthenticationType authType = AuthenticationType.EXTERNAL;
```

Attention:
This text is case sensitive.

4 Locate the following line:

```
GenericAuthenticateResponse authResponse =
authService.AuthenticateGenericChallenge
```

This line is present in the method `void AuthenticateIGPassword(IGUser igUser, string passWord, string newPassWord, UseCase usecase)`.

5 In the second parameter in the method call AuthenticateGenericChallenge(), change `AuthenticationType.PASSWORD` to `AuthenticationType.EXTERNAL`, as follows:

```
GenericAuthenticateResponse authResponse =
authService.AuthenticateGenericChallenge(igUser,
AuthenticationType.PASSWORD, challengeResponse, null, null,
IdentityGuardAuthServiceV9API.SecurityLevel.NORMAL, null, null,
false, false, context, newPassWord, null, null);
```

to

```
GenericAuthenticateResponse authResponse =
authService.AuthenticateGenericChallenge(igUser,
AuthenticationType.EXTERNAL, challengeResponse, null, null,
```

```
IdentityGuardAuthServiceV9API.SecurityLevel.NORMAL, null, null,
false, false, context, newPassWord, null, null);
```

6 Save and close `IdentityGuardOneStepTokenFormsLogin.aspx.cs`.

7 Restart the World Wide Web Publishing Service.

Configuring a protected host

By default the ISAPI Filter creates one `ProtectedHost` element in the filter configuration file, and uses the settings in this element to protect all hosts on your server.

This topics covered in this section are:

- “Modifying the protected host” on page 223
- “Adding or removing protected and unprotected URLs” on page 226
- “Configuring protected and unprotected URLs” on page 230

To configure protected and unprotected URLs for SharePoint, see “Configuring the ISAPI Filter for SharePoint” on page 99

Modifying the protected host

If you want to change the settings associated with this default `ProtectedHost` element, follow the instructions below.

If you have multiple hosts in your environment that you want to protect, but with different settings for each, or if you want to exclude some of your hosts from protection, see “Handling multiple hosts on one server” on page 232.

To configure a protected host

1. Open the file `IdentityGuardFilterConfiguration.xml` either manually using a text editor or using the Configuration Console (see“Configuring ISAPI Filter using the Configuration Console” on page 127).
2. Find the `ProtectedHost` element. For example:

   ```
   <ProtectedHost host="" port="80,443">
   ...
   </ProtectedHost>
   ```

   Edit the `host` and `port` attributes as needed; for example:

   ```
   <ProtectedHost host="" port="443">
   ```

   The value of **`host`** depends on whether you are configuring one protected host or multiple protected hosts. See also “Handling multiple hosts on one server” on page 232.
3. Edit the `IdentityGuardLogin` element.

   Enter the URL for your authentication validation Web service; for example:

   ```
   <IdentityGuardLogin url="https://www.main.com/IdentityGuardAuth">
   ```

4 Ensure that the value of the `LoginPage` element refers to the default value of `Login.aspx`; for example:

```
<LoginPage>
  Login.aspx
</LoginPage>
```

5 Ensure that `AuthValidationServices` is configured.

By default the `AuthValidationServices` element contains one child element, `AuthValidationService`. For example:

```
<AuthValidationServices>
  <AuthValidationService uid=""
  url="https://serverone.mydomain.com/IdentityGuardAuth/AuthValidationService.asmx" />
</AuthValidationServices>
```

`AuthValidationService` has two attributes:

a `uid` is optional in an IIS-only installation..

b `url` specifies the URL for the `AuthValidationServices` Web services file. For example:

```
https://banksite.company.com/IdentityGuardAuth/AuthValidationService.asmx
```

If the URL begins with `https`, you must ensure that the `CACertFile` element points to your CA certificate, and that the host name in the URL matches the common name or a DNS name in the `subjectAltName` extension in the server certificate of the IIS server where the authentication application is running. The host name comparison is case-sensitive so it must match exactly.

For more information see "Replacing and renewing certificates" on page 237.

6 Configure the `Cookies` element.

`Cookies` is an mandatory element containing three child elements, which contain the names of cookies. The cookie names you enter here must match the cookie names in the `cookie` element in `IdentityGuardAuthAppConfiguration.xml`. If you change a cookie name in one file, you must change the matching name in the other file. See "Configuring authentication cookies" on page 235 for more information.

a `UserID` contains the name of the cookie that identifies the current logged-in Entrust Identity Enterprise user. For example:

```
<Cookies>
  <UserID>IGUser</UserID>
  <SessionID>IGSession</SessionID>
```

```
  <SessionKey>IGSessionKey</SessionKey>
  <AuthValidationServiceUID>IGAuthValidationServiceUID</AuthVal
idationServiceUID>
</Cookies>
```

b `SessionID` contains the name of the cookie that identifies the current ASP.NET session of the authentication application. For example:

```
<Cookies>
  <UserID>IGUser</UserID>
  <SessionID>IGSession</SessionID>
  <SessionKey>IGSessionKey</SessionKey>
  <AuthValidationServiceUID>IGAuthValidationServiceUID</AuthVal
idationServiceUID>
</Cookies>
```

c `AuthValidationServiceUID` allows you to ensure that each authentication application remains paired with the session it started in. For example:

```
<Cookies>
  <UserID>IGUser</UserID>
  <SessionID>IGSession</SessionID>
  <SessionKey>IGSessionKey</SessionKey>
  <AuthValidationServiceUID>IGAuthValidationServiceUID</AuthVal
idationServiceUID>
</Cookies>
```

See "Configuring failover in an IIS environment" on page 254 for more information about how to use the `AuthValidationServiceUID` element.

7 Set the logging level to log the amount of information you want to see in the log files.

For more information see "Configure ISAPI Filter for Identity as a Service" on page 142.

8 Optional. Fill in values for the `UserIdHeader` element.

When the ISAPI Filter allows access to a protected resource, it sets an HTTP header using the `name` attribute of the `UserIdHeader` element. The protected application can check this header to see which user is currently authenticated. The `UserIdHeader` element has two attributes.

- The `name` attribute specifies the name of the HTTP header to use for the Entrust Identity Enterprise user ID. The name attribute is set by the ISAPI Filter. For example:

```
<UserIdHeader name="IdentityGuardUserId" encoding="base64"/>
```

where `IdentityGuardUserId` is the name of the HTTP header that contains the Entrust Identity Enterprise user ID.

Note:
The Entrust Identity Enterprise user ID includes the name of the Entrust Identity Enterprise user group if groups are used in Entrust Identity Enterprise. You know groups are used if any of the following are true:

- You are using generic forms-based authentication and the user types in a group name.
- A group name is configured in the authentication application configuration file.
- You are using Windows authentication with custom mapping and the custom mapping function returns a group name.

If groups are used, then the HTTP header will also include the user's group along with their user name (such as igtestgroup/igtestuser). If groups are not used, then the HTTP header will only contain the user name (such as igtestuser).

- The `encoding` attribute specifies the encoding to use for the value of the HTTP header.

  Valid values for the encoding attribute:

  – `none`
  – `base64`
  – `url`

  For example:

  ```
  <UserIdHeader name="IdentityGuardUserId" encoding="base64"/>
  ```

9 Edit the `ProtectedURLs` element to add, remove or modify URLs that you want to protect, or exclude from protection.

  For more information see "Adding or removing protected and unprotected URLs" on page 226.

10 Save and close `IdentityGuardFilterConfiguration.xml`.

Adding or removing protected and unprotected URLs

During installation, you may have entered several URLs for the ISAPI Filter to protect or not protect. That information was stored in the filter configuration file `IdentityGuardFilterConfiguration.xml`. You can edit the entries in the file if you want to modify the list of protected or unprotected URLs.

By default, URLs are unprotected, so you only need to specify unprotected URLs if you specify protected URLs with wild card characters, and you want to exclude specific URLs that match a protected URL pattern.

For more information, see also "Configuring protected and unprotected URLs" on page 230. If you are using SharePoint, you should have already made the changes specified in "Configuring the ISAPI Filter for SharePoint" on page 99.

To add or remove protected and unprotected URLs

1 Open the `IdentityGuardFilterConfiguration.xml` file for editing either manually using a text editor or using the Configuration Console (see"Configuring ISAPI Filter using the Configuration Console" on page 127).

2 Find the `ProtectedURLs` element under the `ProtectedHost` element. For example:

```
<ProtectedHost host="" port="80,443">
...
  <ProtectedURLs authlevel="1" firstfactorid="iwa">
    <URL authlevel="1">/privatefolder/*</URL>
  </ProtectedURLs>
  <UnprotectedURLs/>
</ProtectedHost>
```

The example shows one protected folder, `/privatefolder/*`.

- `authlevel` is the authentication level to be applied to these URLs. This must be a positive integer, and must have a matching definition in the `Level` element under `AuthenticationLevels` in `IdentityGuardAuthAppConfiguration.xml`. For example:

  – In `IdentityGuardFilterConfiguration.xml`:

  ```
  <ProtectedURLs authlevel="1" firstfactorid="iwa">
  ...
  </ProtectedURLs>
  ```

  – In `IdentityGuardAuthAppConfiguration.xml`:

  ```
  <AuthenticationLevels>
    <Level number="1">
      <AuthMethod ref="iwa" />
    </Level>
  </AuthenticationLevels>
  ```

- `firstfactorid` is a string that refers to the type of first-factor authentication to be used with these URLs. It must have a matching

definition in the `FirstFactors` element in `IdentityGuardAuthAppConfiguration.xml`. For example:

– In `IdentityGuardFilterConfiguration.xml`:

```
<ProtectedURLs authlevel="1" firstfactorid="iwa">
  ...
</ProtectedURLs>
```

– In `IdentityGuardAuthAppConfiguration.xml`:

```
<FirstFactors>
  <FirstFactor id="iwa">
    <IWA/>
    ...
</FirstFactor>
...
</FirstFactors>
```

3 In the `ProtectedURLs` element, add or remove any URLs, as needed.

Example of adding a URL to be protected:

```
<ProtectedURLs authlevel="1" firstfactorid="iwa">
  <URL authlevel="1">/privatefolder/*</URL>
  <URL authlevel="1">/privatefoldernew/*</URL>
</ProtectedURLs>
```

- The `URL` element has an optional `authlevel` attribute. This has been set to `1` in the example. If you do not specify an `authlevel`, the URL inherits the `authlevel` from `ProtectedURLs`.
- The `URL` element must contain a string that resolves to a valid URL in your domain.

The `ProtectedURLs` element must have at least one `URL` child element.

During installation, you must specify at least one protected URL to continue; the filter configuration file must have at least one `URL` element under `ProtectedURLs` or the ISAPI Filter does not work.

You can delete `URL` elements that no longer require protection, as long as you leave at least one `URL` element.

4 Find the `UnprotectedURLs` element under the `ProtectedHost` element. For example:

```
<ProtectedHost host="" port="80,443">
...
  <ProtectedURLs authlevel="1" firstfactorid="iwa">
    <URL authlevel="1">/privatefolder/*</URL>
  </ProtectedURLs>
  <UnprotectedURLs>
    <URL>/privatefolder/publicsubfolder/*</URL>
  </UnprotectedURLs>
  </ProtectedHost>
```

The example shows one unprotected folder, `publicsubfolder`, which is a sub-folder of `privatefolder`.

5 In the `UnprotectedURLs` element, add or remove any URLs, as needed.

Example of adding a URL to be excluded from protection:

```
<UnprotectedURLs>
  <URL>/privatefolder/publicsubfolder/*</URL>
  <URL>/privatefolder/publicsubfoldernew/*</URL>
  <URL>/owa/auth/expiredpassword.aspx</URL>
</UnprotectedURLs>
```

The `URL` element must contain a string that resolves to a valid URL in your domain. It should preferably be a subfolder of a protected URL. If it is not, then it is already excluded from protection by default, and there is no need to include it under `UnprotectedURLs`.

You may delete URLs that you no longer want to exclude from protection. If you have no URLS to unprotect, then you can collapse the `UnprotectedURLs` element as shown below.

```
<UnprotectedURLs/>
```

6 Save and close `IdentityGuardFilterConfiguration.xml`.

Note:
If you add '`/*`' to protect all URLs on the site, the install wizard automatically adds '`/IdentityGuardAuth/*`' to the list of unprotected URLs to exclude the authentication application itself from the protected list.
When manually editing the filter configuration file, `IdentityGuardFilterConfiguration.xml`, ensure that you do not include the authentication application in the `ProtectedURLs` section. If you see `/IdentityGuardAuth/*` in the `UnprotectedURLs` section, you may safely leave it there.

Configuring protected and unprotected URLs

If you configured the ISAPI Filter for OWA or RD Web Access during installation, the installer automatically configures the protectedURLs and unprotected URLs, so no changes are necessary.

If you did not select OWA or RD Web Access during installation, and you need to protect these at a later date, or add a log-off URL, edit the `IdentityGuardFilterConfiguration.xml` file to add the configuration settings required, depending on the case you are implementing.

The default URLs for each OWA and RD Web Access case are shown in Table 5.

The table includes options and URLs for which further explanation is required. See below for the explanation for those.

- `authlevel=`

  See “Authentication levels” on page 27 for details.

- `<URL forcelogin=”true”>/owa/auth/logon.aspx</URL>`

  The above string causes the ISAPI OWA first-factor login page to be displayed when a user tries to access one of the URLs specified under `ProtectedURLs`, even if the user has previously logged in. For details, see “Forcing login” on page 196 for details.

- `<URL localaccessonly=”true”>/owa/auth/owaauth.dll</URL>`

  The above URL allows the Entrust Identity Enterprise Web Authentication application to communicate with OWA.

- `<URL localaccessonly=”true”>/IdentityGuardAuth/AuthValidationService.asmx</URL>`

  The above URL allows the Entrust ISAPI Filter to communicate with the Entrust Identity Enterprise Web Authentication application.

- `localaccessonly=”true”`

  When `localaccessonly` is `true` , then the specified URL is accessible from the computer on which it resides without requiring authentication. If the URL is accessed from a remote computer, then authentication is required. Always leave this set to `true` unless instructed to change it by an Entrust representative.

Table 5: Configuration file contents for various OWA and RD Web Access cases

Case	URLs required in `IdentityGuardFilterConfiguration.xml`

OWA 2016 and OWA 2019 on IIS supports Exchange server 2016 and 2019	`<ProtectedURLs authlevel="1" firstfactorid="owa2016">`
	`  <URL authlevel="1">/owa</URL>`
	`  <URL authlevel="1">/public</URL>`
	`  <URL forcelogin="true">/owa/auth/logon.aspx</URL>`
	`  <URL localaccessonly="true">/owa/auth/owaauth.dll</URL>`
	`  <URL authlevel="1">/exchange</URL>`
	`  <URL`
	`  localaccessonly="true">/IdentityGuardAuth/AuthValidati`
	`onService.asmx</URL>`
	`  <URL authlevel="1">/ecp</URL>`
	`  <URL localaccessonly="true">/owa/auth.owa</URL>`
	`</ProtectedURLs>`
	`<UnprotectedURLs>`
	`  <URL>/owa/auth/expiredpassword.aspx</URL>`
	`  <URL>/owa/redir.aspx</URL>`
	`</UnprotectedURLs>`

Remote Desktop Web Access with log-off URL	`<ProtectedURLs authlevel="1" firstfactorid="rdWeb">`
	`  <URL authlevel="1">/rdweb/*</URL>`
	`  <URL localaccessonly="true"`
	`  redirect="https://{rdwa.anycorp.com}/rdweb/">/rdweb/Pages/`
	`  en-US/login.aspx</URL>`
	`  <URL localaccessonly="true">`
	`  /IdentityGuardAuth/AuthValidationService.asmx</URL>`
	`</ProtectedURLs>`
	`<UnprotectedURLs />`
	`<LogoffURLs>`
	`  <URL redirect="/rdweb/">/rdweb/Pages/en-US/logoff.aspx</URL>`
	`</LogoffURLs>`

Handling multiple hosts on one server

By default the ISAPI Filter creates one `ProtectedHost` element in the filter configuration file, and uses the settings in this element to protect all hosts on your server. If you have multiple hosts in your environment that you want to protect, but with different settings for each, or if you want to exclude some of your hosts from protection, you can do so by editing the filter configuration file.

Note:
If you have multiple hosts on the same Web server, set up host headers as described in the Microsoft IIS documentation.

Use the procedures in this section only if there are multiple hosts on this Web server. After adding a specific host name or names to the configuration file, only requests using that host name are protected by the ISAPI Filter. If the host name in the URL does not match what is specified here, it is not protected by the filter. If an IP address is used in the URL instead of the host name, it is not protected by the filter.

Topics in this section:

Protecting a host and excluding other hosts from protection

If you have multiple hosts in your environment and want to exclude all but specifically listed hosts from protection, follow the procedure below.

Use this procedure only if you have multiple hosts on your Web server. After adding the host attribute, protection is limited to hosts specifically named in the configuration file.

To protect a single host

1. Open the file `IdentityGuardFilterConfiguration.xml` either manually using a text editor or using the Configuration Console (see"Configuring ISAPI Filter using the Configuration Console" on page 127).
2. Find the `ProtectedHost` element. For example:

```
<ProtectedHost host="" port="80,443">
```

3 Enter a `host` attribute value that corresponds to the host that you want to protect. For example:

```
<ProtectedHost host="owa.mydomain.com" port="80,443">
```

4 Delete any other `ProtectedHost` elements in the file.

5 Save and close `IdentityGuardFilterConfiguration.xml`.

In the example above, the ISAPI Filter protects `owa.mydomain.com`. It does not protect any other hosts in the environment.

Protecting multiple hosts with the same settings

You can protect all hosts in your environment, using the same settings for all of them. This is the default configuration after completing the ISAPI Filter installation.

To protect all hosts with the same settings

1 Open the `IdentityGuardFilterConfiguration.xml` file.

2 Find the `ProtectedHost` element. For example:

```
<ProtectedHost host="www.mydomain.com" port="80,443">
```

3 Remove the value from the `host` attribute. For example:

```
<ProtectedHost host="" port="80,443">
```

4 Modify the other child elements in the `ProtectedHost` element to match your needs for all hosts in your environment.

For instructions see "Configuring a protected host" on page 223.

5 Delete any other the `ProtectedHost` elements that may exist in the file.

6 Save and close `IdentityGuardFilterConfiguration.xml`.

With this configuration, the ISAPI Filter protects all hosts in the environment. It protects only those URLs that are defined under `ProtectedURLs`.

Protecting multiple hosts with different settings

If you have multiple hosts in your environment, you may want to protect several of them, using different settings for each one. For example you may want to use different authentication methods for each protected host. You can do this by defining multiple `ProtectedHost` elements in the filter configuration file.

To protect multiple hosts with different settings

1 Open the `IdentityGuardFilterConfiguration.xml` file.

2 Find the `ProtectedHost` element. For example:

```
<ProtectedHost host="owa.mydomain.com" port="80,443">
```

3 Select the entire section from `<ProtectedHost` to `</ProtectedHost>`.

4 Copy and paste the entire `ProtectedHost` section immediately below.

5 Edit the new `ProtectedHost` starting tag to refer to the new host that you want to protect. For example:

```
<ProtectedHost host="bank.mydomain.com" port="443">
```

6 Modify the other child elements in the new `ProtectedHost` element to match your needs for `bank.mydomain.com`.

For instructions see "Configuring a protected host" on page 223.

7 Repeat steps 3 to 6 for every host that you want the ISAPI Filter to protect.

8 Save and close `IdentityGuardFilterConfiguration.xml`.

With this configuration, the ISAPI Filter protects the hosts specified in each `ProtectedHost` element, using the settings defined in that element.

Protecting multiple hosts used in a single protected host

If you have multiple subdomains in the set of protected URLs covered by a single protected host, you must specify the common subdomain against the single protected host in the filter configuration file.

If you want to protect both `www.mydomain.com` and `sales.mydomain.com`, for example, you must include `mydomain.com` as a protected host.

To protect multiple hosts used in a single protected host

1 Open the `IdentityGuardFilterConfiguration.xml` file either manually using a text editor or using the Configuration Console (see"Configuring ISAPI Filter using the Configuration Console" on page 127).

2 Find the `ProtectedHost` element. For example:

```
<ProtectedHost host="" port="80,443">
```

3 Enter a `host` attribute value that corresponds to the host that you want to protect. For example:

```
<ProtectedHost host="mydomain.com" port="80,443">
```

4 Save and close `IdentityGuardFilterConfiguration.xml`.

This now protects `www.mydomain.com` along with any subdomains you have defined. For example, Entrust ISAPI Filter now protects:

```
www.mydomain.com
sales.mydomain.com
internal.mydomain.com
internal.mktg.mydomain.com
.....
```

Configuring authentication cookies

You can modify your cookies settings by editing the authentication application configuration file. For example, you may want to change the cookie names, or configure them for use with SSL or non-SSL connections. For cookie settings related to SharePoint integration, see “Configuring the filter to use persistent cookies with SharePoint” on page 104.

To configure authentication cookies

1. Open the `IdentityGuardAuthAppConfiguration.xml` file either manually using a text editor or using the Configuration Console (see“Configuring ISAPI Filter using the Configuration Console” on page 127).
2. Find the `AuthenticationWebApplication` element. There are two `Cookies` elements in the file; the `Cookies` element inside the `AuthenticationWebApplication` element is the one to edit.
3. Optional: Enter a value for the `domain` attribute if you want to enable single sign-on across multiple hosts. This attribute contains the name of the external domain that is accessed from the user’s browser. If multiple subdomains are used, then it contains the base domain. For example:

   ```
   <Cookies domain="mydomain.com">
   ```

Note:
Leave the `domain` attribute blank if either of these conditions is true:
- You have only one external host name.
- You have multiple hosts, but they do not share a common domain.

4. Enter the `secure` attribute and give it a value.

   This attribute has two possible values, `true` or `false`.

   When it is set to `true`, cookies are used only against secure connections. The browser does not send them over non-SSL connections.

   When it is set to `false`, the secure flag is not set on the cookies. This allows them to be sent over both SSL and non-SSL connections.

   If any of your protected applications, including the Entrust Identity Enterprise authentication application, can be accessed by a user using HTTP, you must set the `secure` attribute to `false` for the solution to work.

   For example:

   ```
   <Cookies domain="mydomain.com" secure=”false”>
   ```

5 Enter the `httpOnly` attribute only to change the default: `true`. Setting this to `false` could leave a site venerable to cross-site scripting if other barriers to hackers are not in place. For example:

```
<Cookies domain="mydomain.com" secure="false" httpOnly="true">
```

6 Enter values for the child elements as described below.

The `cookies` element contains three common child elements, whose values are cookie names. The cookie names in `IdentityGuardAuthAppConfiguration.xml` must match the cookie names in the `cookie` element of `IdentityGuardFilterConfiguration.xml`. If you change a cookie name in one file, you must change the matching name in the other file. See "Configuring a protected host" on page 223 for more information.

a `UserID` is the name of the cookie that identifies the current logged in Entrust Identity Enterprise user.

b `SessionID` is the name of the cookie that identifies the current ASP.NET session of the authentication application. If you change the name of this cookie you must make the corresponding change in the `sessionState` element in the `Web.config` file. The following example shows the `cookieName` attribute of the `sessionState` element:

```
<sessionState
  mode="InProc"
  stateConnectionString="tcpip=127.0.0.1:42424"
  sqlConnectionString="data
source=127.0.0.1;Trusted_Connection=yes"
  cookieless="false"
  cookieName="IGSession"
  timeout="20" />
```

c `AuthValidationServiceUID` is used to enforce session affinity to a specific authentication application. See "Configuring failover in an IIS environment" on page 254 or "Configuring failover in an IIS environment" on page 254 for more information. For example:

```
<Cookies>
  <UserID>IGUser</UserID>
  <SessionID>IGSession</SessionID>
  <SessionKey>IGSessionKey</SessionKey>
  <AuthValidationServiceUID>IGAuthValidationServiceUID</AuthVal
idationServiceUID>
</Cookies>
```

7 Save and close `IdentityGuardAuthAppConfiguration.xml`.

Replacing and renewing certificates

If you replace the SSL certificate that is used by the authentication application with a certificate from a different vendor, you must update the filter configuration file before using the new certificate. This allows the ISAPI Filter to find the new CA certificate (trusted root certificate file). This certificate is used by the ISAPI Filter to communicate with the authentication application.

When you renew certificates with the same CA, no configuration changes are needed.

Replacing a certificate

When you replace certificates created by a different CA, the new CA certificate file must replace the existing file. If the file name has changed, the filter configuration file must be updated to point to the new file.

To replace a certificate

1 Export the new CA certificate, following the procedure in "Exporting certificates to Base-64 format" on page 50.

Copy the certificate file to the server on which the ISAPI Filter component is installed. If you save the new certificate using the same file name and folder as the old certificate, then you do not need to perform any further configuration. The ISAPI Filter automatically points to the new certificate. The default location is `C:\Program Files\Entrust\Identity\WinIS\config`.

It is recommended that you use the same folder to simplify future maintenance.

2 If your new CA certificate uses a different file name or folder than the old CA certificate, update the filter configuration file to refer to the new file.

a Open `IdentityGuardFilterConfiguration.xml` either manually using a text editor or using the Configuration Console (see"Configuring ISAPI Filter using the Configuration Console" on page 127).

b Find the `CACertFile` element.

c Edit the value of the element so that it refers to the correct file name and folder for your new CA certificate; for example:

Old entry:

```
<CACertFile>C:\Program Files\
Entrust\Identity\WinIS\config\oldcacert.cer</CACertFile>
```

Updated entry:

```
<CACertFile>C:\Program Files\
Entrust\Identity\WinIS\config\newcacert.cer</CACertFile>
```

d Save and close `IdentityGuardFilterConfiguration.xml`.

Configuring compatibility with SharePoint

SharePoint 2010 uses .NET Framework 2.0. This version of the Entrust ISAPI Filter is optimized for SharePoint 2013 and 2016 which uses .NET Framework 4.x.

To use this version of the ISAPI Filter integration with SharePoint 2010, you must remove two lines that are not supported in .NET Framework 2.0 from the Web.config file.

To configure compatibility with SharePoint

1. Open the `Web.config` file. The default location of the file is:

   ```
   C:\Program Files
   \EntrustInt\Identity\WinIS\webapp\IdentityGuardAuth
   ```

2. Search for, and then remove, the following lines

   ```
   <pages enableViewState="false" enableSessionState="true"
   controlRenderingCompatibilityVersion="3.5" clientIDMode="AutoID"/>
   ```

3. Save and close the file.

Modifying other configurations

This section describes settings in the configuration files that are not described elsewhere in this guide.

Topics in this section:

Understanding filter configuration file settings

This section contains brief descriptions of settings in the ISAPI Filter configuration file, `IdentityGuardFilterConfiguration.xml`, that have not been described elsewhere in this guide.

Element name	Description
`Entrust-IdentityGuard-WinIS`	Mandatory root element.
`Version`	This configuration setting contains the version number associated with the installed Entrust ISAPI Filter. For example: `13.0`
`FilterSettings`	Mandatory container element for filter-wide configuration settings.
`FilterEnvironment`	Indicates the environment the ISAPI Filter is installed in. Possible value: • `IIS`
`ListenToAccessDeniedEvent`	This setting is used to enable ISAPI Filter to listen to SF_NOTIFY_ACCESS_DENIED events. Use this setting if you experience the problem described in "Error message appears when second-factor challenge expected" on page 298, cause 1.
`Headers`	Mandatory container element for HTTP header values used to pass information from the ISAPI Filter to the authentication application. If any HTTP header names are changed from the default, then the corresponding entries in the `Headers` element of the authentication application configuration file must be changed to match the new names.

Element name	Description
`LevelHeader`	Contains the HTTP header `LevelHeader`. Exactly one must be present. Attributes: • `name` Name of the HTTP header used by the ISAPI Filter to pass the authentication level to the authentication application. Cannot be empty. Must match the `LevelHeader` value in `IdentityGuardAuthAppConfiguration.xml`. Default value: `EntrustIdentityGuardLevel`
`FirstFactorIDHeader`	Contains the HTTP header `FirstFactorIDHeader`. Exactly one must be present. Attributes: • `name` Name of the HTTP header used by the ISAPI Filter to pass the first-factor authentication identifier to the authentication application. Cannot be empty. Must match the `FirstFactorIDHeader` value in `IdentityGuardAuthAppConfiguration.xml`. Default value: `EntrustIdentityGuardFirstFactorID`
`PassthroughIPHeader`	Contains the HTTP header `PassthroughIPHeader`. Exactly one must be present. Attributes: • `name` Name of the HTTP header used by the ISAPI Filter to pass the user's IP address to the authentication application. Cannot be empty. Must match the `PassthroughIPHeader` value in `IdentityGuardAuthAppConfiguration.xml`. Default value: `EntrustIdentityGuardClientIP`
`DisableHostnameVerification`	This is an optional setting with a default value of `false`. When set to `true`, it disables the hostname verification for SSL between filter and authentication application.

Element name	Description
`FilterUsernameHeader`	Contains the HTTP header `FilterUsernameHeader`. Exactly one must be present. Attributes: • `name` Name of the HTTP header used by the ISAPI Filter to pass the user name to the authentication application. Cannot be empty. Must match the `FilterUsernameHeader` value in `IdentityGuardAuthAppConfiguration.xml`. Default value: `EntrustIdentityGuardFilterUsername`
`ClientCertificateHeader`	Contains the HTTP header ClientCertificateHeader. Exactly one must be present. Attributes: • name Name of the HTTP header used by the ISAPI Filter to pass the certificate to the authentication application during a risk-based authentication (RBA). For details, see Step 22 on page 94. Cannot be empty. Must match the `ClientCertificateHeader` value in `IdentityGuardAuthAppConfiguration.xml`.

Understanding authentication application configuration settings

This section contains brief descriptions of settings in the authentication application configuration file, `IdentityGuardAuthAppConfiguration.xml`, that are not described elsewhere in this guide.

Element name	Description
`Entrust-IdentityGuard-WinIS`	Mandatory root element.

Element name	Description
FirstFactors	Defines the first-factor authentication methods All of the first-factor methods are present in the file when ISAPI Filter is installed. It is recommended that you comment out all FirstFactor settings you are not using. To comment out an unused setting, surround the line with: <!-- unused setting --> Example: <!-- FirstFactor id="iwa"--> Attributes: • <FirstFactor id> String value used to identify the first-factor set up. This name must match the firstfactorid setting in the filter configuration file. Example: <FirstFactor id="iwa"> The child element names are: • FormsBased • IWA • OWA • RDWebAccess
FormsBased	Child element of FirstFactor. Use if you want to use generic forms-based first-factor authentication. Attributes: • page An optional attribute, containing a string value which refers to a login page for forms-based first-factor authentication. Can be any valid string, ending in .aspx. – ApplicationLogin.aspx is used for forms-based authentication – IdentityGuardLogin.aspx is used for Entrust Identity Enterprise password using forms-based authentication If it is left out, the ISAPI solution defaults to ApplicationLogin.aspx.
IWA	Child element of FirstFactor. Use IWA if you want to use Integrated Windows Authentication for first-factor authentication.

Element name	Description
OWA	Child element of FirstFactor. Use OWA if you want to use Outlook Web Access for your first-factor authentication. Contains several child elements, described below: • Version • BaseURL • AuthService • TestURL • Cookies • CharSet • RememberQueryString
RDWebAccess	Child element of FirstFactor. Use RDWebAccess if you want to use Remote Desktop Web Access for first-factor authentication. Contains several child elements, described below: • LoginPage • UpdatePasswordPage • Cookies • SSO • RDWPublicModeSessionTimeoutInMinutes • RDWPrivateModeSessionTimeoutInMinutes
AuthService	Child element of OWA. Contains the URL to the OWA authentication DLL to which the user's user name and password are sent. Example: <AuthService url="https://{exchange.company.com}/owa/auth/owaauth.dll"/>
BaseURL	Child element of OWA. Contains the URL used to access OWA. Example: <BaseURL url="https://{exchange.company.com}/owa"/>
CharSet	Optional. Contains the character set that should be used to encode the user ID and password when sending them to OWA for authentication.

Element name	Description
RememberQueryString	Optional configuration setting for Firstfactor id owa2016. Default value is true. When set to false, ISAPI Filter does not append any query strings in the redirection URL after completing authentication. Example: <RememberQueryString>false</RememberQueryString>
TestURL	Child element of OWA. Contains the URL used by the ISAPI solution to check whether the user is logged in. Example: <TestURL url="https://{exchange.company.com}/owa/redir.aspx"/>
Version	Mandatory child element of OWA. Valid values are: • 2016 (used when supporting OWA Exchange 2016, and 2019) Example: <OWA version="2016">
LoginPage	Child element of RDWebAccess. Contains the URL used to access Remote Desktop Web Access.
RDWPrivateModeSessionTimeoutInMinutes	Child element of RDWebAccess. Sets a session timeout value.
RDWPublicModeSessionTimeoutInMinutes	Child element of RDWebAccess. Sets a session timeout value.
SSO	Child element of RDWebAccess. Enables or disables single sign on. Example: <SSO enabled="false"> <WorkSpaceID>{workSpaceID}</WorkSpaceID> <RDPCertificate /> </SSO>

Element name	Description
Cookies	Mandatory child element of OWA and RDWebAccess. The Cookies element contains the names of cookies used and passed by the ISAPI solution to the user's browser. Attributes: • domain Optional: Contains the name of the external domain that is accessed from the user's browser. If multiple subdomains are used, then it contains the base domain. For example: <Cookies domain="mydomain.com"> • privateClientCookieLifetime For RDWebAccess only, it contains the lifetime in minutes of the cookies. For example: <Cookies privateClientCookieLifetime="30"> For OWA, the Cookies element contains two child elements, SessionID and cadata, which contain the names of cookies used by OWA. The ISAPI solution passes these names to the user's browser. It is recommended that you do not change the default values. Default values: <SessionID>sessionid</SessionID> <cadata>cadata</cadata> For RDWebAccess, the Cookies element contains two child elements, AuthClientSideCookie and AuthHttpOnlyCookie, which contain the names of cookies used by RDWebAccess. The ISAPI solution passes these names to the user's browser. It is recommended that you do not change the default values. Default values: <AuthClientSideCookie>TSWAAuthClientSideCookie</AuthClientSideCookie> <AuthHttpOnlyCookie>TSWAAuthHttpOnlyCookie</AuthHttpOnlyCookie>
Headers	Mandatory container element for HTTP header values; used to pass information from the ISAPI Filter to the authentication application. If any HTTP header names are changed from the default, then the corresponding entries in the Headers element of the filter configuration file must be changed to match the new names.

Element name	Description
`LevelHeader`	Contains the HTTP header `LevelHeader`. Exactly one must be present. Attributes: • `name` Name of the HTTP header used by the ISAPI Filter to pass the authentication level to the authentication application. Cannot be empty. Must match the `LevelHeader` value in `IdentityGuardFilterConfiguration.xml`. Default value: `EntrustIdentityGuardLevel`
`FirstFactorIDHeader`	Contains the HTTP header `FirstFactorIDHeader`. Exactly one must be present. Attributes: • `name` Name of the HTTP header used by the ISAPI Filter to pass the first-factor authentication identifier to the authentication application. Cannot be empty. Must match the `FirstFactorIDHeader` value in `IdentityGuardFilterConfiguration.xml`. Default value: `EntrustIdentityGuardFirstFactorID`
`PassthroughIPHeader`	Contains the HTTP header `PassthroughIPHeader`. Exactly one must be present. Attributes: • `name` Name of the HTTP header used by the ISAPI Filter to pass the user's IP address to the authentication application. This attribute cannot be empty, and it must match the `PassthroughIPHeader` value in `IdentityGuardFilterConfiguration.xml`. Default value: `EntrustIdentityGuardClientIP`

Element name	Description
`FilterUserNameHeader`	Contains the HTTP header `FilterUsernameHeader`. Exactly one must be present. Attributes: • `name` Name of the HTTP header used by the ISAPI Filter to pass the user name to the authentication application. This attribute cannot be empty, and it must match the `FilterUsernameHeader` value in `IdentityGuardFilterConfiguration.xml`. Default value: `EntrustIdentityGuardFilterUsername`
`ClientCertificateHeader`	Contains the HTTP header `ClientCertificateHeader`. Exactly one must be present. Attributes: • `name` Name of the HTTP header used by the ISAPI Filter to pass the certificate to the authentication application during a risk-based authentication (RBA). For details, see Step 22 on page 94. This attribute cannot be empty, and it must match the `ClientCertificateHeader` value in `IdentityGuardFilterConfiguration.xml`. Default value: `EntrustIdentityGuardClientCertificate`

6

Configuring failover in your environment

The ISAPI Filter solution gives you the ability to use failover by increasing the number of Entrust Identity Enterprise Servers, so that if the preferred server fails or is unavailable, the next server can take over.

Topics in this chapter:

Configuring failover for Entrust Identity Enterprise Servers

The ISAPI Filter solution gives you the capability of setting up a failover architecture by increasing the number of Entrust Identity Enterprise Servers. With multiple Entrust Identity Enterprise Servers, failover works as follows:

1. Upon startup, the first Entrust Identity Enterprise Server in the list, (also called the preferred server), is used to process all authentication requests.
2. When a successful connection cannot be made to the current, active Entrust Identity Enterprise Server, then the ISAPI solution fails over to the next available Entrust Identity Enterprise Server, always starting from the preferred server, and skipping over any unavailable servers.
3. At defined intervals that you can configure, the ISAPI solution attempts to reconnect to the preferred Entrust Identity Enterprise Server. The default interval is one hour.

You can configure failover for Entrust Identity Enterprise Servers by editing the authentication application file.

To configure failover for Entrust Identity Enterprise Servers

1. Open the file `IdentityGuardAuthAppConfiguration.xml` for editing.
2. Find the `IdentityGuardServers` element under `AuthenticationWebApplication`. For example:

```
<AuthenticationWebApplication>
  <IdentityGuardServers >
      ...
  </IdentityGuardServers>
      ...
</AuthenticationWebApplication>
```

   Define the attributes of `IdentityGuardServers` as described in the following sub-steps. The attributes defined within this element apply to all the Entrust Identity Enterprise Servers.

   a. Define the `numberOfRetries` attribute. For example:

```
<IdentityGuardServers numberOfRetries="1">
    ...
</IdentityGuardServers>
```

      If the first connection attempt to a server fails, this setting indicates how many further attempts must be made before marking this server as failed.

If not specified, the default value is `1`; that is, after an initial (failed) attempt, one further attempt is made.

b Define the `delayBetweenRetries` attribute. For example:

```
<IdentityGuardServers numberOfRetries="1"
    delayBetweenRetries=”500”>
  ...
</IdentityGuardServers>
```

`delayBetweenRetries` is used with the `numberOfRetries` attribute. It specifies how long to wait (in milliseconds) between connection attempts. The default value, if not specified, is `500` milliseconds. If `numberOfRetries` is `0`, then `delayBetweenRetries` is not used.

c Define the `failedServerHoldOffTime` attribute. For example:

```
<IdentityGuardServers numberOfRetries="1"
    delayBetweenRetries=”500”
    failedServerHoldOffTime=”600” >
  ...
</IdentityGuardServers>
```

`failedServerHoldOffTime` defines the minimum amount of time (in seconds) that must elapse before attempting to contact a server that has previously been marked as failed. The default value, if not specified, is `600` seconds (10 minutes).

d Define the `restoreTimeToPreferred` attribute. For example:

```
<IdentityGuardServers numberOfRetries="1"
    delayBetweenRetries=”500”
    failedServerHoldOffTime=”600”
    restoreTimeToPreferred="3600">
  ...
</IdentityGuardServers>
```

When the current active, connected server is not the preferred server (that is, the first server in the list), then the `restoreTimeToPreferred` setting defines how frequently (in seconds) to try to reconnect to the preferred server. The default value, if not specified, is `3600` seconds (one hour). Setting a value of `0` (zero) means that the solution continues to use the current active server, and does not attempt to reconnect to the preferred server.

3 Find the `ServerList` element under `IdentityGuardServers`. For example:

```
<IdentityGuardServers numberOfRetries="1"
     delayBetweenRetries=”500”
```

```
      failedServerHoldOffTime=”600”
      restoreTimeToPreferred="3600">
   <ServerList>
    ...
  </ServerList>
</IdentityGuardServers>
```

`ServerList` contains definitions of all the Entrust Identity Enterprise Servers in your environment.

4 Add an `IdentityGuardServer` element under `ServerList`. For example:

```
<ServerList>
  <IdentityGuardServer>
     ...
  </IdentityGuardServer>
</ServerList>
```

Each `IdentityGuardServer` element defines one of the Entrust Identity Enterprise Servers in your environment.

5 Add an `AuthenticationService` element under `IdentityGuardServer`. For example:

```
<ServerList>
   <IdentityGuardServer>
    <AuthenticationService />
  </IdentityGuardServer>
</ServerList>
```

The `AuthenticationService` element contains the URL for the authentication service of the Entrust Identity Enterprise Server being defined.

6 In the `AuthenticationService` element, enter the URL for your first Entrust Identity Enterprise Server. For example:

```
<ServerList>
  <IdentityGuardServer>
    <AuthenticationService
url="https://igserver1.mydomain.com:8443/IdentityGuardAuthService/
services/AuthenticationServiceV11"/>
  </IdentityGuardServer>
</ServerList>
```

This completes the definition of one Entrust Identity Enterprise Server.

7 Repeat steps 4 to 6 for each additional Entrust Identity Enterprise Server in your environment. For example:

```
<ServerList>
  <IdentityGuardServer>
    <AuthenticationService
url="https://igserver1.mydomain.com:8443/IdentityGuardAuthService/
services/AuthenticationServiceV11"/>
  </IdentityGuardServer>

  <IdentityGuardServer>
    <AuthenticationService
url="https://igserver2.mydomain.com:8443/IdentityGuardAuthService/
services/AuthenticationServiceV11"/>
   </IdentityGuardServer>

  <IdentityGuardServer>
    <AuthenticationService
url="https://igserver3.mydomain.com:8443/IdentityGuardAuthService/
services/AuthenticationServiceV11"/>
  </IdentityGuardServer>
</ServerList>
```

8 Save and close `IdentityGuardAuthAppConfiguration.xml`.

9 Restart IIS for your configuration changes to take effect.

You have completed the configuration of failover for your Entrust Identity Enterprise Servers.

Configuring failover in an IIS environment

ISAPI Filter can be installed on multiple IIS servers using load-balancing or failover technologies as long as session stickiness is in place to ensure that user sessions, once established, stay on the same IIS server.

In round-robin DNS, the DNS server returns a different IP address for a host name to spread the load across two physical hosts. For example, clients use host name `mail.company.com`, but there are two physical hosts—`mail1.company.com` and `mail2.company.com`. DNS lookups for `mail.company.com` return the IP address for `mail1.company.com` part of the time and the IP address for `mail2.company.com` the rest of the time.

Working in an environment with multiple IIS services

The ISAPI Filter and authentication application need to communicate with services (OWA login and Entrust Identity Enterprise authentication validation) running on the same host. If the service URLs are not specified correctly, authentication failures happen intermittently because sometimes the ISAPI Filter is routed to the other host (`mail2.company.com`, for example) rather than its own host (`mail1.company.com`, for example).

To prevent this problem when using this configuration you must ensure that each IIS server resolves the host name to the local host rather than going through load-balancing. Use the following guidelines:

- Use server-specific URLs—`BaseURL`, `AuthService`, and `TestURL`—in the OWA URLs section of the `IdentityGuardAuthAppConfiguration.xml` file on each IIS server. For more information about these settings, see "`OWA`" on page 243.

  As an example, in the `IdentityGuardAuthAppConfiguration.xml` file on the mail1 server, the OWA URLs contain `mail1.company.com`.

  In the `IdentityGuardAuthAppConfiguration.xml` file on the mail2 server, the OWA URLs contain `mail2.company.com`.

- In the `IdentityGuardFilterConfiguration.xml` file on each server, ensure that the `AuthValidationService` URL points to a physical host, not the virtual host name.

  For example, on server `mail1.company.com`, `AuthValidationService` is defined as:

  ```
  http://mail1.company.com/IdentityGuardAuth/AuthValidationService.asmx
  ```

  On server `mail2.company.com`, `AuthValidationService` is defined as:

  ```
  http://mail2.company.com/IdentityGuardAuth/AuthValidationService.asmx
  ```

7

Migrating users to Entrust Identity Enterprise or Identity as a Service

User migration is the process of making all your end users of Entrust Identity Enterprise or Identity as a Service users who access your protected resources through the ISAPI Filter. The ISAPI Filter solution has user migration features you can configure to allow your users to continue to access your protected resources while you deploy your solution.

You can either force or phase in migration.

Topics in this chapter:

Implementing user migration

There are two ways to migrate users to Entrust Identity Enterprise:

- “Forcing migration” on page 256
- “Phasing in migration” on page 257

Forcing migration

In this scenario, after you install the Entrust ISAPI Filter solution, you force all users to enroll with Entrust Identity Enterprise and to activate a second-factor authentication method. Until they complete the enrollment, they cannot access protected resources.

Forced migration works well when you have a small number of end users. It is recommended that you implement a cutoff date before which all users must complete the enrollment.

If you have a large number of end users, they could all attempt the migration at once, causing heavy demand on your servers. To avoid this problem, you may want to implement “Phasing in migration”.

To set fallback to Entrust Identity Enterprise

If your users exist in Entrust Identity Enterprise but not Identity as a Service and you have set the AuthenticationAPI to Identity as a Service but you want to set fallback authentication to Entrust Identity Enterprise, you need to set the `<AllowFallbackToIG>` to **`true`**.

1. Stop World Wide Web Publishing Service.
2. Go to `<ISAPIFilter_install>\WinIS\config` and open the `IdentityGuardAuthAppConfiguration.xml` file.
3. Locate `<AllowFallbackToIG></AllowFallbackToIG>`.
   - If `AllowFallbackToIG` is **`true`** and if the user exists in Identity as a Service, continue MFA using the Identity as a Service API, if not fall back to Entrust Identity Enterprise.

     The Entrust Identity Enterprise user is not enrolled in the Identity as a Service Admin Portal but the user wants to access the protected resources through ISAPI Filter.
   - If `AllowFallbackToIG` is **`false`** and if the user exists in Identity as a Service, continue MFA using Identity as a Service API, if not show an error message.

     The Entrust Identity Enterprise user is not enrolled in the Identity as a Service Admin Portal and you want to restrict access to the protected resources through ISAPI Filter.

4 Save and close `IdentityGuardAuthAppConfiguration.xml`.

5 Restart World Wide Web Publishing Service.

To implement forced migration

1 Create Entrust Identity Enterprise or Identity as a Service user IDs for all your end users.

2 Modify the `<UserMigration>` element in the file `IdentityGuardAuthAppConfiguration.xml` as shown below:

```
<SkipAuthNoExist enabled="false"/>
<SkipAuthNoActive enabled="false"/>
```

3 Instruct your end users that they cannot access protected resources until they enroll with Entrust Identity Enterprise and activate a second-factor authentication method.

Phasing in migration

In this scenario, you migrate your users in groups. Migrate the first group of users. Wait a few days to test the system, and eliminate any problems before proceeding to migrate the next group.

To implement phased migration

1 Create Entrust Identity Enterprise or Identity as a Service user IDs for your first batch of users.

2 Have those users enroll in Entrust Identity Enterprise or Identity as a Service and assign second-factor authentication methods to them.

- For Entrust Identity Enterprise, you can enroll your users, or you can have them self-register using client software such as Entrust Identity Enterprise Self-Service Module or Entrust Identity Enterprise Desktop for Microsoft Windows.
- For Identity as a Service, you can create users or migrate users in bulk from Entrust Identity Enterprise or Active Directory.

Users who are already enrolled are not affected by the modifications described in the following steps.

3 Decide how you want the ISAPI Filter to handle the users who are not yet migrated, and modify the `<UserMigration>` element in the file

`IdentityGuardAuthAppConfiguration.xml` using the scenarios described below:

- If you want to block unmigrated users completely from the protected resource:

  ```
  <SkipAuthNoExist enabled="false"/>
  <SkipAuthNoActive enabled="false"/>
  ```

- If you want to allow unmigrated users unrestricted access to the protected resource:

  ```
  <SkipAuthNoExist enabled="true"/>
  <SkipAuthNoActive enabled="true" />
  ```

- If you want to redirect unrestricted users to another Web page:

  ```
  <SkipAuthNoExist enabled="true"
  url="https://adminigsss.mycorp.com:8445/IdentityGuardSelfService"/>
  <SkipAuthNoActive enabled="true"
  url="https://adminigsss.mycorp.com:8445/IdentityGuardSelfService"/>
  ```

  Put in your own URL for the Web page, instead of the example shown above.

There are other possible scenarios depending on how you want the ISAPI Filter to handle your unmigrated users. See "Modifying user migration settings" on page 259 for the effect of each setting.

4 After you have migrated your first group of users, you can repeat steps 1 to 3 to migrate the next group.

Repeat until you have migrated all your users. After all your users are registered, you can disable the user migration feature, if desired, by changing the enabled attribute to `false` for both `<SkipAuthNoExist>` and `<SkipAuthNoActive>`.

Modifying user migration settings

When you deploy your ISAPI solution, you may have end users in different states with regard to Entrust Identity Enterprise.

- Users may not have a user ID created in Entrust Identity Enterprise or Identity as a Service.
- Users may have a user ID created in Entrust Identity Enterprise, but do not yet have an Entrust Identity Enterprise or Identity as a Service password or second-factor authentication method assigned and activated.
- Users may have a user ID created in Entrust Identity Enterprise or Identity as a Service, and they have an Entrust Identity Enterprise or Identity as a Service password or second-factor authentication method assigned and activated.

The user migration settings in the authentication application configuration file allow you to choose how you handle the three types of users when they attempt to access a protected URL. User migration is configured globally for the entire solution. The user migration settings apply to all authentication methods in the solution.

You control the behavior of these features by modifying settings in the `<UserMigration>` element of `IdentityGuardAuthAppConfiguration.xml`.

The `<UserMigration>` element has two child elements:

- "Modifying the SkipAuthNoExist element" on page 259
- "Modifying the SkipAuthNoActive element" on page 260

Modifying the SkipAuthNoExist element

This element applies to users who have not yet been added to Entrust Identity Enterprise or Identity as a Service.

Users who have already been added in Entrust Identity Enterprise or Identity as a Service are not affected by the settings of this element.

`SkipAuthNoExist` has an attribute called `enabled`, which has two possible values, `true` or `false`. The default is `false`. It has the optional attribute `url`. You can use the element in several different ways.

If you set…	The effect is…
`<SkipAuthNoExist enabled="false"/>`	Non-Entrust Identity Enterprise and Identity as a Service users are blocked from the protected resource. This is the default setting.

If you set…	The effect is…
`<SkipAuthNoExist enabled="true"/>`	Non-Entrust Identity Enterprise and Identity as a Service users are allowed access to the protected resource without a second-factor challenge.
`<SkipAuthNoExist enabled="true" url="IdentityGuardEnrollment.aspx"/>`	Non-Entrust Identity Enterprise and Identity as a Service users are not allowed to access the protected resource, and they are redirected to the given URL. This URL could be a page informing the user to contact support, or a self-service interface for registering. The example shows the default page. It informs the user that they have not yet been enrolled in Entrust Identity Enterprise or Identity as a Service.

Modifying the SkipAuthNoActive element

This element applies to users who have been added to Entrust Identity Enterprise, but do not yet have any assigned and activated second-factor authentication methods, such as grid, token, Q&A, or OTP.

Users who already have activated second-factor methods are not affected by the settings of this element.

`SkipAuthNoActive` has an attribute called `enabled`, which has two possible values, `true` or `false`. The default is `false`. It has the optional attribute `url`. You can use the element in several different ways.

If you set…	The effect is…
`<SkipAuthNoActive enabled="false"/>`	Entrust Identity Enterprise or Identity as a Service users who do not yet have assigned and activated second-factor authentication methods are blocked from the protected resource. This is the default setting.
`<SkipAuthNoActive enabled="true"/>`	Entrust Identity Enterprise or Identity as a Service users who do not yet have assigned and activated second-factor authentication methods are allowed access to the protected resource without a second-factor challenge.

If you set…	The effect is…
`<SkipAuthNoActive enabled="true" url="IdentityGuardActivation.aspx"/>`	Entrust Identity Enterprise or Identity as a Service users who do not yet have assigned and activated second-factor authentication methods are not allowed to access the protected resource, and they are redirected to the given URL. This URL could be a page informing the user to contact support, or a self-service interface for registering. The example shows the default page informing the user that they do not yet have an active second-factor authentication method.

8

Customizing the Entrust ISAPI Filter Authentication Web application

This chapter describes ways of customizing the solution, such as adding your own logo and colors to the login pages, changing the font and layout of the Web forms, and modifying text and error messages that are displayed to the user.

You can make these customizations by modifying the contents of login forms and other files. These files contain sample code (that is commented out) for your convenience.

Note:
Before you make any customizations, back up the files so that you can return to the original settings if required.

Note:
These customizations are lost during upgrade or patch installs. Back up your customized files before upgrading or installing patches.

Note:
To edit the text files, use a text editor such as Editplus, which does not change the line codes. Notepad is not recommended.

Topics in this chapter:

Changing the appearance of the application pages

You can customize the appearance of the Web pages by modifying the master page and style sheet.

Customizing the master page

The Web pages in the authentication application get their layout from a master page, `AuthApp.master` located at

```
C:\Program Files\
Entrust\Identity\WinIS\webapp\IdentityGuardAuth\
```

This file defines the layout for all the pages. Individual `.aspx` pages refer to the master page for their overall layout. See the Microsoft documentation for information about how to modify master pages.

Customizing the style sheet

The master page uses a single style sheet, `AuthApp_style.css`, to define the look and feel of the Web pages.

You can use your own style sheet and modify the master page to refer to it, or you can modify this file to customize the look and feel of the pages. You can find this file at

```
C:\Program
Files\Entrust\Identity\WinIS\webapp\IdentityGuardAuth\
```

Replacing the default logo with a custom logo

You can change the logo that appears on the pages. By default, the image files are in `C:\Program Files\
Entrust\Identity\WinIS\webapp\IdentityGuardAuth\Images`

They have names such as `logo_sm.gif` or `logo2.gif`. Replace them with your own customized image files. If you do not want to replace the existing files, you can place your own image files in that directory, then modify the login files to point to your custom image files, as shown in the example below.

To replace logo2.gif with a file named customlogo.gif

1 If you are using forms-based login, open the `ApplicationLogin.aspx` file and find the `Image ID` tag; for example:

```
<asp:Image ID="Image1" ImageUrl="~/Images/logo2.gif"
CssClass="logoLarge".../>
```

2 Modify it to:

```
<asp:Image ID="Image1" ImageUrl="~/Images/customlogo.gif"
CssClass="logoLarge".../>
```

This is an example; replace `customlogo.gif` with your logo file name.

3 Save and close `ApplicationLogin.aspx`.

4 If you are using OWA login, follow the procedure with `OWALogin.aspx`.

5 If the new image size is different than that of the default image:

a Open the `AuthApp_style.css` file for editing.

b Locate the `width` and `height` attributes in the `logoLarge` selector.

c Edit the property values for width and height.

d Save and close `AuthApp_style.css`.

After you make these changes, you do not need to restart the ISAPI Filter. It automatically picks up the changes.

Customizing user interface strings, including error messages

All user-visible text, including error messages (with rare exceptions), are found in `IGISAPIGlobal.resx`, in the following directory:

```
C:\Program Files\
Entrust\Identity\WinIS\webapp\IdentityGuardAuth\
App_GlobalResources\
```

Note:
Certain fatal errors visible to users, such as an invalid filter configuration file, result in an error message that cannot be localized.

To customize message strings

1. Open the `IGISAPIGlobal.resx` file in a text editor the does not change line feed codes, for example, Editplus. (Notepad is not recommended.)
2. To customize user-visible text in this file,
   a. Edit the name/value pair of the message you want to alter, for example:

```
<data name="ig_int_display_nActive" xml:space="preserve">
  <value>Your account has not been activated to use
second-factor authentication. Please contact your
administrator.</</value>
</data>
```

   b. Change the text in the `value` tag to your desired text; for example:

```
<value>Your account must be activated for second-factor
authentication. Please contanct your administrator.</value>
```

3. To have each sentence appear on a separate line, add a line feed code (
) at the end of each line after which you want a break, as shown in bold in the following example:

```
<data name="ig_int_display_nActive" xml:space="preserve">
  <value>Your account must be activated for second-factor
authentication.&#13; Please contact your
administrator.</</value>
</data>
```

   After the change to the value and the addition of the line break, your resulting message would appear on-screen as follows:

Your account must be activated for second-factor authentication.

Please contact your administrator.

4 Save the changes to the `IGISAPIGlobal.resx` file. You must save the `.resx` file with UTF-8 encoding. You can select the **UTF-8** option in the **Save As** dialog box of most text editors.

5 With your text editor, open `AuthApp_style.css`, the cascading style sheet (CSS) file used by the ISAPI Filter. It can be found in the following directory

```
C:\Program
Files\Entrust\Identity\WinIS\webapp\IdentityGuardAuth\
```

6 Search for the `.error{}` section.

7 Add the `white-space: pre;` style attribute, which aligns the new line feed characters (if you added line feeds in Step 3), as shown in bold text.

```
.error
{
    font-size:80%;
    font-family:Arial, sans-serif;
    color:red;
    font-weight:bold;
    font-size:11px;
    text-align:left;
    white-space: pre;
}
```

Note:
This change to the style sheet affects all error messages, so any message that does not contain a line feed code will appear on a single line.

8 Save the CSS file. You do not need to restart the ISAPI Filter. It automatically picks up the changes.

Customizing HTTP error messages

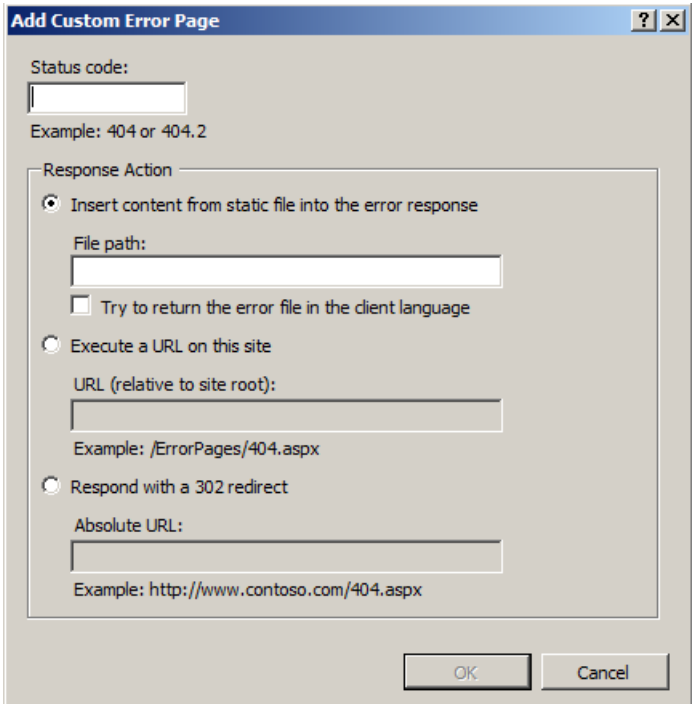
HTTP has a long list of error messages that browsers will display when a user tries to access an unknown, non-existent or forbidden URL. Some of these are very specific, for example: 403.7 - Client certificate required; 403.14 - Directory listing denied. The wording of the default messages may give a hacker a useful clue. For example “Directory listing denied” could indicate that the directory exists, and encourage the hacker to apply a brute force strategy to find files within the folder.

Internet Information Services (IIS) lets you assign custom error messages to specific HTTP error codes. You then configure Entrust Identity Enterprise to use those messages.

To configure custom messages through IIS

1. Open IIS. Access it through Administrative Tools.
2. In the **Connections** pane, expand the Web server name, expand **Sites**, and then navigate to the Web site or application that you want to configure custom error pages for.
3. In the **Home** pane, double-click the **Error Pages** icon (not the .**NET Error Pages** icon).
4. In the **Actions** pane, click **Add**.

The **Add Custom Error Page** dialog appears.



5 In **Status Code**, enter the number for the error code to which you want to assign a custom error message (for example, 403.14).

6 Under **Response Action**, select one of the three options and specify, as applicable, a path or URL that points to the custom error message file on your server.

Note the path or URL: you will need it when configuring Entrust Identity Enterprise.

7 Click **OK**.

To configure Entrust Identity Enterprise to use the custom error messages

1 Open the `web.config` file for editing in a text editor, such as NotePad. It is located in `\Entrust\Identity\WinIS\webapp\IdentityGuardAuth\`.

2 Scroll down to the `<system.webServer>` element.

3 Just above `</system.webServer>`, add code similar to the following, where the attribute values reflect the code you selected and the path or URL you assigned in IIS.

```
<httpErrors errorMode="DetailedLocalOnly"
  defaultResponseMode="File">
    <remove statusCode="403.14" />
```

```
    <error statusCode="403.14"
        prefixLanguageFilePath="C:\errors\”
        path="message403-24.html" />
</httpErrors>
```

The `httpErrors` element is a standard IIS construct. Information on is available on the Internet including at: http://www.iis.net/ConfigReference/system.webServer/httpErrors

4 Save and close the file.

5 Restart IIS.

Attention:
Do not change the default setting for `customErrors mode=` in the `web.config` file. A setting of `off` could negate your `httpErrors` configuration.

Adding support for another language

The resource file, `IGISAPIGlobal.resx`, which contains all the user interface strings and error messages, is in English. Separate resource files are used for different countries, and different languages. The ISO 3166 standard is followed for the country code. Name the file using the country and language code, in the format `<language>-<country>`.

- `<language>` is an ISO-369 Alpha 2 language code; for example, the language code for Spanish is `es`.
- `<country>` is an ISO-3166 Alpha 2 country code; for example, the country code for Mexico is `MX`.

For example:

`IGISAPIGlobal.resx` is used for the en-US locale. This is the default resource file.

`IGISAPIGlobal.es.resx` is the resource file for Spanish, and would be used when the user locale is set to Spain.

`IGISAPIGlobal.es-mx.resx` is the resource file for Mexican Spanish, and would be used when the user locale is set to Mexican Spanish.

To add support for a locale

1. Ensure that your end users set the locale on their browsers to match the desired country and language; for example `Spanish (Mexico)[es-MX]`.
2. Make a copy of `IGISAPIGlobal.resx`, and rename it to match the country and language of your end users; for example `IGISAPIGlobal.es-mx.resx`.
3. Edit the user interface strings and error messages in the resource files, so that they match the country and language of your end users.
4. Save the file.

After you make these changes, you do not need to restart the ISAPI Filter. It automatically picks up the changes.

Note:
You must save all `.resx` files with UTF-8 encoding. You can select the **UTF-8** option in the **Save As** dialog box of most text editors.

Customizing first-factor login form pages

When you install the ISAPI Filter, a `\webapp` folder is created under

`C:\Program Files\Entrust\Identity\WinIS\`

The files you use to customize the Authentication Web application are located in

`C:\Program Files\Entrust\Identity\WinIS\webapp\IdentityGuardAuth\`

There are three main first-factor login forms that you can customize.

Table 6: Customizing login forms

To customize first-factor login for...	Make changes to...
generic forms-based login	`ApplicationLogin.aspx` This is a login form used to identify the user and provide the first-factor authentication. If you want to modify the business logic of your first-factor login form, see "Implementing your first-factor login in generic forms-based authentication" on page 108
Outlook Web Access	`OWALogin.aspx` This is the input form used to provide the first-factor authentication for logging into Outlook Web Access for Microsoft Exchange 2016 or 2019 when using the IIS-only deployment.
Entrust Identity Enterprise password login	`IdentityGuardLogin.aspx` This is a login form used to identify the user and provide the first-factor authentication using Entrust Identity Enterprise password.

9

Upgrading ISAPI Filter

This chapter describes how to upgrade from ISAPI Filter to 12.0.

Note:
Back up all your customizations before upgrading ISAPI Filter.

Topics in this chapter:

Upgrading from previous versions of Entrust ISAPI Filter

Follow the instructions below to upgrade ISAPI Filter from a previous version to 12.0.

To upgrade

1. Uninstall the previous version of the ISAPI Filter and authentication application. See “To uninstall previous versions of ISAPI Filter from an IIS server” on page 278.
2. Download and install the 12.0 version of the ISAPI Filter and the authentication application. See “Installing Entrust ISAPI Filter” on page 73.
3. Reapply your customizations. See “Recreating your customizations from a previous installation” on page 276.

Recreating your customizations from a previous installation

The ISAPI Filter stores configuration and language customizations in XML-like text files that are easy to edit.

After you complete the full upgrade, you may want to recreate any customizations that you made previously.

To recreate your customizations, you must manually add them to the new version. See “Customizing the Entrust ISAPI Filter Authentication Web application” on page 263 for more information.

10

Uninstalling the Entrust Identity Enterprise ISAPI solution

This chapter describe the procedures for uninstalling the ISAPI solution.

Topics in this chapter:

Uninstalling the filter from an IIS server

If you installed the ISAPI Filter on several IIS servers, you must uninstall them from each server.

To uninstall previous versions of ISAPI Filter from an IIS server

1. Stop the World Wide Web Publishing service.
   a. Open a command prompt.
   b. Enter:

      ```
      iisadmin /stop
      ```

   Alternatively, you can access the Services page, right-click the service, and select **Stop**.

   If the World Wide Web Publishing Service is running while you uninstall the solution from an IIS server, the uninstaller displays an error message, and allows you to stop the service before completing the uninstallation.
2. Click **Start** > **Settings** > **Control Panel** > **Add or Remove Programs**.
3. Select **Entrust ISAPI Filter**.
4. Click **Remove**.
5. Follow the instructions in the uninstaller.
6. Restart the World Wide Web Publishing Service. At the command prompt, enter:

   ```
   iisadmin /start
   ```
7. Manually remove the Entrust folder that remains in the Programs Files located at `C:\Program Files\Entrust\`

A

Appendix A: How the filter works

This appendix describes how the components of the ISAPI Filter work together.

Topics in this appendix:

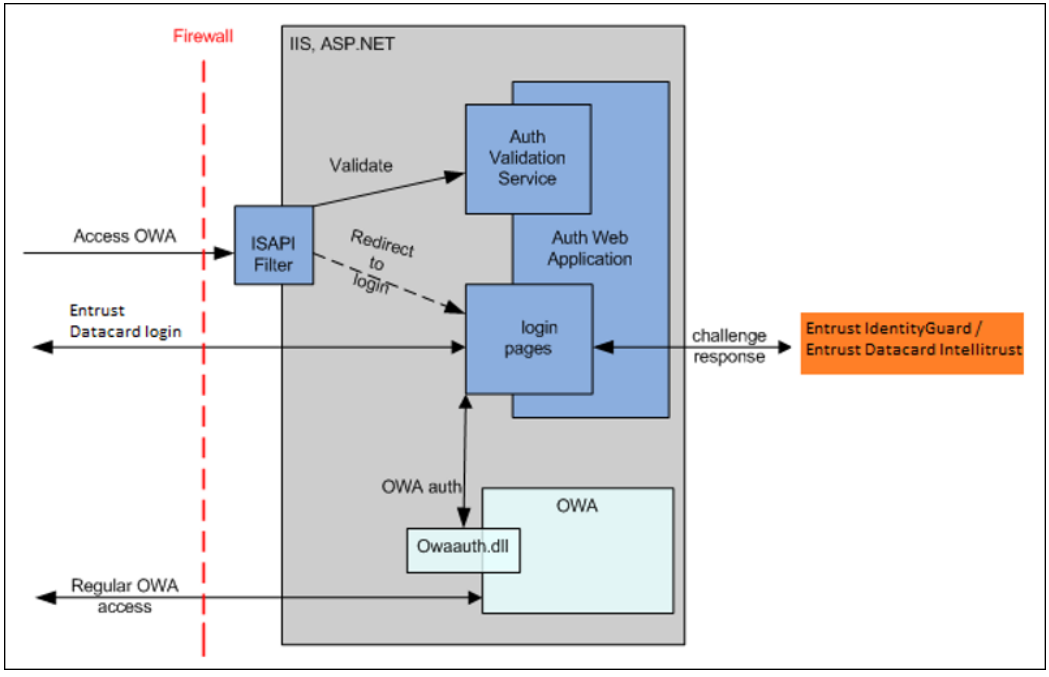
Understanding IIS-only deployment

The following diagram shows the ISAPI solution components in more detail, based on an IIS-only deployment. The ISAPI Filter component monitors traffic to protected URLs. The ISAPI Filter checks with the AuthValidation service (part of the authentication application) to verify whether the user is logged in.

When the user is not logged in, they are redirected to the authentication application login pages; otherwise they go directly to the protected resource, provided they are authenticated at the right level for that resource. If not, they meet a step-up authentication challenge.

For OWA, the login pages prompt the user for the first and second-factor authentication. The login pages validate the authentication against OWA (for first-factor), and Entrust Identity Enterprise (for second-factor). Some applications, such as Integrated Windows Authentication, may show only second-factor authentication. Others, such as generic-forms applications, show both first and second-factor. After the user is authenticated, they are redirected to the protected resource (OWA, for instance).

Figure 6: Entrust ISAPI Filter architecture with IIS deployment



Flow sequence in an IIS-only deployment

The following sequence of events occurs when a user attempts to access a protected URL after the ISAPI Filter solution is installed.

1. Using a browser, a user enters the URL of the Web application.

   The ISAPI Filter intercepts the request and checks if the URL is protected. If the resource is protected, the ISAPI Filter sends a redirect to the browser. The user is redirected to the first-factor login form.

Note:
The list of protected URLs and the URL of the first-factor login form, are specified in the configuration file for the ISAPI Filter.

2. The browser displays the first-factor login form.

   This page typically asks for a user name and password, but it can be customized.

3. The user enters credentials for the first-factor login.

4. If user verification is successful, Entrust Identity Enterprise generates a second-factor challenge for the user. This challenge can be one of the following:

   - Grid
   - Token
   - Out-of-band one-time password (OTP)
   - Knowledge-based (question and answer)

Note:
The challenge may be skipped if risk-based authentication is being used, and if the user has previously authenticated successfully from the same location. This is determined by Entrust Identity Enterprise policy settings.

5. The second-factor authentication form is displayed to the user.

   In the case of grid, token, and Q&A, this challenge is presented to the user in the second-factor login Web form. In case of OTP, the user selects a delivery method, the one-time password is sent to the user out-of-band (for example, email, or SMS) and the second-factor login Web form is presented to the user, prompting them to enter the one-time password after they receive it.

Depending on Entrust Identity Enterprise OTP policy, they may have unused OTPs. If so, they can enter an OTP from their list.

6 The user submits the response to the challenge.

7 After the response is verified by Entrust Identity Enterprise, the login form sets the session cookies, and sends a redirect to the browser to send the user to the protected Web application.

The user can now access the protected resource.

The session cookies remain valid until:

- The user closes the browser.
- The browser is idle longer than the session `timeout` parameter value set in the `Web.config` file. The `timeout` parameter default value is 20 minutes.
- The ISAPI Filter sees a request to the logoff URL.

B

Appendix B: Using wild card characters to specify URLs

This appendix describes how to use wild card characters to specify protected and unprotected URLs. You can also protect or unprotect file extensions.

When you specify URLs to protect or leave unprotected, you can use an asterisk '*' to specify a wild card character, as shown in the examples in this appendix.

If you installed the ISAPI Filter server and selected OWA as the first-factor authentication type, you do not need to configure the protected and unprotected URLs for Outlook Web Access. They are automatically configured. For more information, see "Configuring protected and unprotected URLs" on page 230.

You do have to configure them for your other applications.

This appendix contains the following sections:

Protecting or unprotecting URLs with wild card characters

The following table shows examples of how you can use wild cards in URLs.

/protected/*	Protect any URL under the /protected path
Protects:	/protected, /protected/abc
Does not protect:	/protected123
/pro*	protect a URL where the path element starts with pro
Protects:	/protected, /proabc, /pro
Does not protect:	/protected/abc, /prabc
/*tected	Protect a URL where the path element ends with tected
Protects:	/protected, /abctected, /tected
Does not protect:	/protected/abc, /123ted
/pro*ted	protect a URL where the path element starts with pro and ends with ted
Protects:	/protected, /proabctected, /proted
Does not protect:	/protected/abc, /pr123d
/p*o*t*d	Protect a URL where the path element has p, o, t, d in that order.
Protects:	/protected, /p1o2t3d, /potd, /pot1111d
Does not protect:	/protected/abc, /optd
/*/protected	Protect a URL called protected under any single URL path element
Protects:	/abc/protected
Does not protect:	/protected, /abc/123/protected

DOCISSUE30 If you want to protect OWA after the installation, see Adding or modifying OWA authentication, etc. etc. (what used to be Forcelogin)

Protecting or unprotecting a file extension

To unprotect all files with a certain extension, you can edit the `IdentityGuardFilterConfiguration.xml` configuration file. Use syntax similar to this example:

```
<UnprotectedURLs>
  <URL>/privatefolder/publicsubfolder/*jar</URL>
</UnprotectedURLs>
```

C

Appendix C: Enabling logging during the installation of the ISAPI Filter solution

This appendix describes how to enable logging during the installation of the ISAPI Filter.

To enable logging during the installation, run the ISAPI Filter installer from a command prompt, using the following syntax:

Logging Option:	Command:
Enable logging	`IG_ISAPI_Filter_13.0.msi /l`
Append to existing log file	`IG_ISAPI_Filter_13.0.msi /l+`

Enable logging and specify the name of the log file
Use this option if you do not want to accept the default log file name.
	`IG_ISAPI_Filter_13.0.msi /log <LogFile>`
	Example:
	`IG_ISAPI_Filter_13.0.msi /log mycorp1.log`

The installer writes the log file to the `Temp` folder. The default name of the file begins with "`Msi`" and ends with "`.log`".

To find the log file, enter one of the following at a command prompt:

```
cd %temp%
cd %tmp%
```

D

Appendix D: Disabling the ISAPI Filter without uninstalling it

This appendix describes how to disable the ISAPI Filter without uninstalling it. This can be a useful troubleshooting step, to see if the system is working properly without the ISAPI Filter.

Second-factor authentication is disabled when you do this, so it is recommended that you disable the ISAPI Filter only temporarily for troubleshooting purposes. After you have completed your troubleshooting, re-enable the Filter.

Topics in this appendix:

Disabling and enabling the ISAPI Filter on IIS

Follow these procedures when you want to disable and re-enable a filter that is installed on an IIS server.

Procedures in this section include:

- "To disable the ISAPI Filter on Microsoft IIS 10" on page 290
- "To enable the ISAPI Filter on Microsoft IIS 10" on page 290

To disable the ISAPI Filter on Microsoft IIS 10

1. Open Internet Information Services (IIS) Manager.
2. In the left pane, under **Sites**, select the Web site where you installed the ISAPI Filter, (for example **Default Web Site**).
3. If you are not in **Features View**, right-click the Web site, and select **Switch to Features View**.
4. Double-click **ISAPI Filters**.
5. Select **Entrust ISAPI Filter**.
6. In the right pane, click **Remove**.
7. Restart the IIS Admin Service. From a command prompt, enter:

   ```
   iisreset /noforce
   ```

After you have completed your troubleshooting procedures, you can re-enable the ISAPI Filter with one of the following procedures.

To enable the ISAPI Filter on Microsoft IIS 10

1. Open Internet Information Services (IIS) Manager.
2. In the left pane, under **Sites**, select the Web site where you installed the ISAPI Filter, (for example **Default Web Site**).
3. If you are not in **Features View**, right-click the Web site, and select **Switch to Features View**.
4. Double-click **ISAPI Filters**.
5. In the right pane, click **Add**.
6. In the **Add ISAPI Filter** dialog box, do the following:
   a. Enter the **Filter name**: ISAPI Filter name (for example, Entrust ISAPI Filter).
   b. Enter the **Executable**: Path of the ISAPI Filter dll (for example, `C:\Program Files\Entrust\Identity\WinIS\bin\IdentityGuardFilter64.dll`)

7 Restart the IIS Admin Service. From a command prompt, enter:

```
iisreset /noforce
```

E

Appendix E: Troubleshooting the ISAPI Filter solution

This appendix describes how to address some problems that you may encounter with the ISAPI Filter solution. See the log files for additional information.

Topics in this appendix:

Errors occur after changing your application pool settings

Problem:

Errors occur when you change the application pool settings so that your authentication Web application is in the same pool as Outlook Web Access or a protected application.

Cause:

The authentication Web application sometimes has conflicts with OWA or the protected resource if you placed them in the same pool. By default, the installer puts the authentication application in a different pool from OWA to prevent conflicts.

Solution:

If you change the application pool of any of these applications, ensure that you place them in separate application pools. It is recommended that you do not change the default application pool settings.

End users get repeated Entrust Identity Enterprise login prompts

Problem:

Users are repeatedly prompted to log into Entrust Identity Enterprise. In Firefox you might see the message **Redirect Loop**.

Causes:

1. This may be caused by the cookie domain being set incorrectly. It may also be caused if you have configured the solution to use secure cookies, but your protected application or authentication application can be accessed by a user using HTTP. See “Solution:”.
2. Another possible cause: First-factor anonymous access is allowed to some or all URLs in your protected application. See “User not automatically redirected from machine authentication page”.

Solution:

Modify the cookie domain settings in `IdentityGuardAuthAppConfiguration.xml`. Make sure the cookie domain and secure cookie settings are correct for your configuration. For details see “Configuring authentication cookies” on page 235.

Restart the World Wide Web Publishing Service on the IIS server hosting the authentication application.

User not automatically redirected from machine authentication page

Problem:

When using machine authentication, the user is not automatically redirected from the machine authentication page. See “To test your solution on IIS with Integrated Windows Authentication” on page 119.

Cause:

This may be caused by the user's browser having two different versions of Flash installed.

Solution:

Uninstall the older version of Flash.

To determine which versions of Flash are installed in the browser, navigate to `http://www.adobe.com/software/flash/about`. The page displays the Flash version numbers near the top.

See also “Configuring risk-based authentication” on page 185.

User ID from filter does not match the stored ISAPI Filter user name

Problem:

Users see unexpected Entrust Identity Enterprise challenges when clicking links in the protected Web application. The log shows that the user ID from the ISAPI Filter does not match the stored filter user name.

Example log message:

```
[2010-10-28 14:34:58,906] [0x00000ad0] [ERROR]
[.\IdentityGuardAuthValidationServiceClient.cpp(144)][checkIsAuthe
nticated] UserId from filter does not match the stored Filter
username ISAPI-SP\administrator from IG session
```

Causes:

1. You are using the IIS-only configuration with Integrated Windows Authentication (IWA), and some of the resources protected by the filter have anonymous access allowed at the IIS level. See “Solution 1:”.
2. You are using the IIS-only configuration to protect a SharePoint site that uses IWA, and you are protecting the URL pattern `/*`. SharePoint has images under `/_layouts/images` that have anonymous access enabled. See “Solution 2:”.

Solution 1:

Do one of the following:

- If you are using IWA, make sure the URLs marked “protected” in the filter configuration file have anonymous access disabled in IIS so that IWA is used for all URLs.
- Add new URLs that allow anonymous access, by adding them as unprotected URLs in the filter configuration file.

Solution 2:

Add `/_layouts/images` as an unprotected URL in the filter configuration file. See “Configuring protected and unprotected URLs for SharePoint” on page 100.

Error message appears when second-factor challenge expected

Problem:

Users see unexpected error message after entering valid first-factor credentials. Rather than being directed to the second-factor challenge page, the user sees:

```
You could not be authenticated...
```

Example log message:

```
[ERROR][Entrust.IdentityGuard.Integrations.ISAPI.Usecase.UseCaseFa
ctory.getUseCase] [session=none available] - Object reference not
set to an instance of an object.
```

Cause:

The ISAPI Filter authentication application is installed in the same Web site as SharePoint, and the session state module has not been added for the IdentityGuardAuth application.

Solution 1:

If the ISAPI Filter is installed on the same Web site as SharePoint, you must add the session state module to the Entrust Identity Enterprise authentication application.

In a command prompt, enter the following command.

```
appcmd add module /name:Session
/type:System.Web.SessionState.SessionStateModule
/preCondition:managedHandler
/app.name:"<SharePoint-site>/IdentityGuardAuth"
```

where `<SharePoint-site>` is the common Web site where SharePoint and the IdentityGuardAuth application are running.

For example:

```
appcmd add module /name:Session
/type:System.Web.SessionState.SessionStateModule
/preCondition:managedHandler
/app.name:"sharepoint.anycorp.com/IdentityGuardAuth"
```

Solution 2:

Change the `<ListenToAccessDeniedEvent>` setting in the `IdentityGuardFilterConfiguration.xml` file to **true**. For more information, see “Understanding filter configuration file settings” on page 239.

IWA: An authenticated Windows User ID is missing from the session.

Problem:

Users see unexpected error message after entering valid IWA first-factor credentials. Rather than being directed to the second-factor challenge page, the user sees:

```
The system cannot login you in. Please contact your
administrator.
```

Example log message:

[2020-02-10 01:52:21,876] [7] [ERROR] [IntegratedAuth.Page_Load] [session=vtpsf3u3uttkj2ni0jsac3jf] - An authenticated Windows userid is missing for this session. Authentication cannot proceed.

Causes 1:

You are using the IIS-only configuration with Integrated Windows Authentication (IWA), and the resource `/IdentityGuardAuth/` installed by the filter has Anonymous access allowed at the IIS level. See “Solution 1”.

Solution 1

If you are using IWA, resource `/IdentityGuardAuth/` installed by the filter should have Anonymous access disabled and Windows Authentication enabled at the IIS level.

Index

A

C

D

E

F

G

H

I

K

L

M

O

P

R

S

T

U

W

X