MyID PIV
Version 12.10

System Security Checklist

Copyright

Conventions used in this document

- Lists:
    - Numbered lists are used to show the steps involved in completing a task when the order is important.
    - Bulleted lists are used when the order is unimportant or to show alternatives.
- **Bold** is used for menu items and for labels.

  For example:
    - Record a valid email address in **'From' email address**.
    - Select **Save** from the **File** menu.
- *Italic* is used for emphasis:

  For example:
    - Copy the file *before* starting the installation.
    - Do *not* remove the files before you have backed them up.
- ***Bold and italic*** hyperlinks are used to identify the titles of other documents.

  For example: "See the ***Release Notes*** for further information."

  Unless otherwise explicitly stated, all referenced documentation is available on the product installation media.
- A `fixed width` font is used where the identification of spaces is important, including filenames, example SQL queries and any entries made directly into configuration files or the database.
- **Notes** are used to provide further information, including any prerequisites or configuration additional to the standard specifications.

  For example:

  Note: This issue only occurs if updating from a previous version.
- Warnings are used to indicate where failure to follow a particular instruction may result in either loss of data or the need to manually configure elements of the system.

  For example:

  Warning: You must take a backup of your database before making any changes to it.

Contents

1 Introduction

MyID® provides you with all the tools you need to secure your MyID system, ensuring that your system is not vulnerable to attack.

However, when you first set up MyID, you may not want to lock down the system completely, allowing you to complete the initial test and setup more easily. Accordingly, the security features have not been made mandatory within MyID.

It is important that you understand how to secure your system correctly before issuing cards on a live system. If you do not:

- Your system may be vulnerable to attack.
- Your cards may not be FIPS 201-compliant.

If you have any questions about securing your system, contact customer support:

support@intercede.com

Note: MyID is often integrated with multiple third-party components and systems, such as PKI, HSMs, smart cards, mobile devices and devices with hardware security features. It is the customer's responsibility to ensure that these are appropriately configured to meet the organization's security requirements.

2 Securing Devices

When you install MyID, the settings on the **Device Security** page of the **Security Settings** workflow are configured to require you to use customer GlobalPlatform keys and random Security Officer PINs (SOPINs). The system is also configured to display warnings if your system is not securely configured:
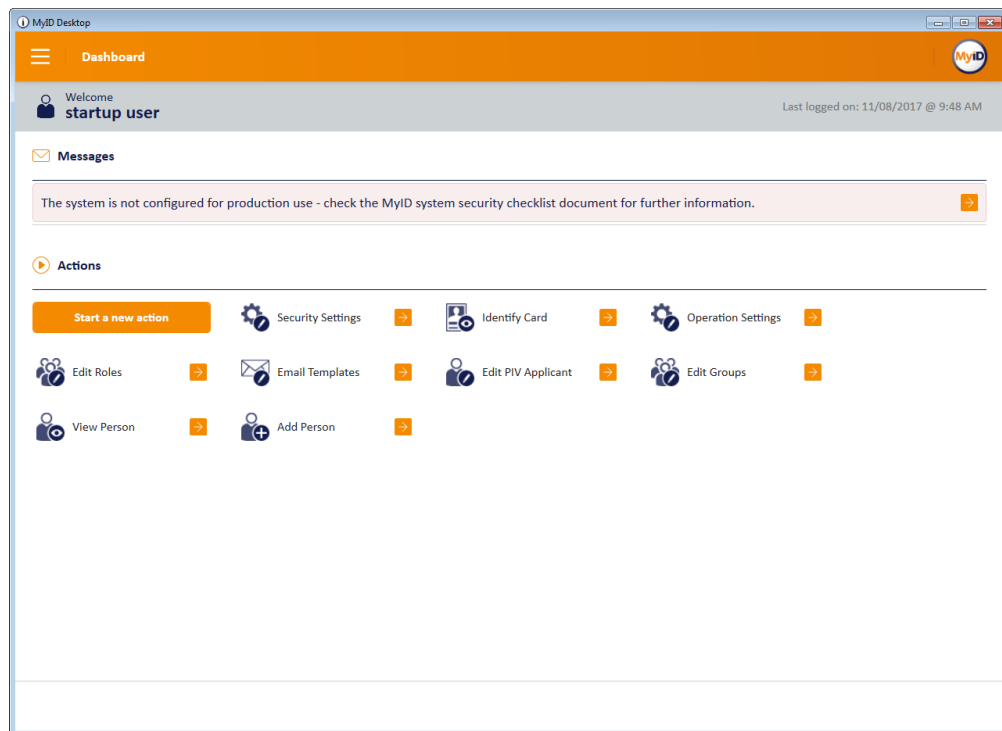
The message is:

```
The system is not configured for production use - check the MyID system
security checklist document for further information.
```

If this warning appears, you must review the settings on the **Device Security** tab on the **Security Settings** workflow:

Setting	Default value	Description
Display warnings for unsecured issuance	Yes	Displays a warning on the login screen if the system is not securely configured and an attempt is made to issue credentials. You must ensure that your system is configured appropriately according to the guidance provided in the System Security Checklist and your own security policy. If you want to run MyID with secure settings disabled (for example, for test or demonstration systems) and this option is not available to be edited on your system, contact customer support to discuss your requirements, quoting reference SUP-273.
Enable Customer GlobalPlatform Keys	Yes	Whether the installation supports Java applets. If you do not have this option set, you will be unable to write customer GlobalPlatform keys to your cards.
Require Random Security Officer PIN	Yes	If this is set to `Yes` but the **Security Officer PIN Type** is set to `Factory`, cards cannot be issued.
Security Officer PIN Type	Random	`Random` – Generate a random SOPIN and set it on the card to be initialized (higher security). `Factory` – Leave the default SOPIN on the card (low security).
Show all devices	No	When set to `No`, restricts the list of devices on this page to the smart cards known to support GlobalPlatform or PIV 9B keys. When set to `Yes`, displays all devices known to MyID.

Note: You can also set the requirements for customer GlobalPlatform and PIV 9B keys for each device type supported by your system. If the option is set to Yes, and the card supports the feature, MyID requires the customer key to be configured before issuing devices of this type.

If you change any of the options on this screen away from the default, your system will be potentially insecure, and MyID will display an appropriate warning when logging in to MyID or when issuing a smart card that would be affected.

2.1 System security

When attempting to issue a card, you may also see a message similar to the following:

```
System is not set up to issue this card
```

This is because MyID is not configured to issue this type of card in accordance with the security requirements on the **Device Security** tab.

The **System Events** report may include further information about the system security. The following codes appear in the report:

- `s` – MyID is not correctly configured to swap the SOPIN to a randomized value at issuance.

- `G` – MyID is not correctly configured to swap the GlobalPlatform key to a customer value at issuance.

- `P` – MyID is not correctly configured to swap the PIV9B key to a customer value at issuance.

See section *3.1*, *SOPINs*, section *4.1*, *PIV 9B keys*, and section *4.2*, *GlobalPlatform key sets* for information about configuring SOPINs, PIV 9B keys, and GlobalPlatform keys to ensure that your system is secure and configured for production use.

For further information on these system security messages, contact customer support quoting reference SUP-273.

3 Securing PINs

Securing the PINs for your cards is essential to maintaining the security of your system.

You must consider the following:

- Security Officer PINs (SOPINs)
- PIN complexity

3.1 SOPINs

The SOPIN (Security Officer PIN) is an unlocking code that allows the cardholder's PIN to be set to a new value in the event that the PIN has been forgotten or the card is PIN locked. The SOPIN is sometimes referred to as the PIN Unlock Key (PUK).

Note: Any card that does not have a PIN (for example, a door access contactless-only card) will not have an SOPIN. All cards with a PKI applet have an SOPIN.

If the card has an SOPIN, MyID must know and manage the SOPIN to issue, cancel or unlock the card.

Cards are delivered with a fixed factory SOPIN.

3.1.1 Risks

- If cards are issued with the SOPIN still set to a factory value then the SOPIN will be the same on every card of that type; therefore, it is possible that it is known to unauthorized parties.
- An unauthorized party with the SOPIN can reset the cardholder's PIN to a value of their choice.
- Having reset the PIN, an unauthorized party would have access to signature and decryption operations to impersonate the cardholder or access their private data.

3.1.2 Solution

Configure MyID to randomize the PIN during issuance; this means that each card is issued with a unique SOPIN known only to MyID. MyID securely manages the randomized SOPIN for each card so that it can continue to cancel or unlock the cards. Unauthorized parties will therefore be unable to modify the cardholder's PIN.

3.1.3 Implementation

Within the **Security Settings** workflow, on the **Device Security** page, set the **Security Officer PIN Type** options to **Random**.

See the *Device Security page (Security Settings)* section in the ***Administration Guide*** for details.

If you are upgrading from an earlier version of MyID, check the **Configuration** tab on the **System Status** workflow – if the `SHOW SOPIN` configuration option appears in the list, contact customers support quoting reference SUP-220 for guidance on setting this option to `NO` or removing the option entirely.

3.1.4 Considerations

When you issue a card with a random SOPIN, if you intend to use the card on a different MyID installation, you must first cancel the card on the system on which it was issued – this changes the SOPIN back to the factory setting.

3.1.5 Recommendations

You must configure your system for random SOPINs before your production system goes live.

3.2 PIN complexity

MyID allows you to set up rules for the length and complexity of the PINs used for devices. For example, you can set the PINs to require uppercase, lowercase, numeric and symbol characters.

3.2.1 Risks

Insufficiently complex PINs may be guessed by a third party who could then gain access to your system.

3.2.2 Solution

Implement a PIN complexity policy that supports your requirements for security and standards.

3.2.3 Implementation

Within MyID, set up a card profile and use the **PIN Settings** section to set up the PIN length and supported characters.

See the *Credential profile options* section in the *Administration Guide* for details.

3.2.4 Considerations

Some card types and middlewares may be unable to support the full range of PIN complexity rules that MyID offers. Make sure that the rules you set up in MyID match the PINs that are accepted by your cards and middleware.

Some cards do not allow the PIN rule enforcement to be stored on the card; MyID will enforce the PIN rules, but external software may be able to change the PIN on the card without the rules being enforced.

Your system may need to comply with standards for particular purposes; for example, FIPS 201. These standards may contain regulations regarding the allowed PIN complexity rules. Check the documentation for details.

You may need different PIN policies for different situations (end user cards and administrator cards, different card types) in which case you can set up different card profiles for each purpose.

3.2.5 Recommendations

Set a PIN policy that is sufficiently complex to allow for good security, while matching the limitations of your devices or middleware, and conforming to any standards appropriate for your system.

4 Securing Keys

This section describes the most common use of smart card keys in MyID. If your installation has been customized to make use of additional key types, (for example, ICAO Applet keys), contact Intercede customer support for further information.

The following features allow you to make your issued cards secure:

- PIV 9B keys (only applicable for PIV cards)
- GlobalPlatform keys

4.1 PIV 9B keys

Note: 9B keys are applicable only for PIV cards, and cards based on PIV technology.

The PIV 9B key (also known as the PIV admin key) is a symmetric key on every PIV card. MyID needs to know the 9B key to write data or generate RSA keypairs on a PIV card.

Cards are delivered with a fixed factory PIV 9B key. You must set up MyID with the factory key for the appropriate device type. This allows MyID to authenticate to the card and write PIV data during the issuance.

4.1.1 Risks

- The factory PIV 9B key is the same on every card of that type; two cards of the same model from the same manufacturer will have the same factory 9B key. Therefore, it is possible that the key is known to unauthorized parties.
- An unauthorized party with the PIV 9B key can modify the content of the PIV card.
- A PIV card that has an unchanged factory PIV 9B key is not FIPS 201 compliant. You must issue your cards with diversified customer keys that are stored on an HSM.

4.1.2 Solution

Set up MyID to replace the factory PIV 9B key with a customer PIV 9B key – this is a key known only to the customer's system. Unauthorized parties will not have access to this customer PIV 9B key, and therefore cannot perform any unauthorized modifications of the PIV cards issued by MyID.

Set the following options on your customer keys:

- **Key Diversity: Diverse** – each card is issued with a different key, derived from a master key. Even in the unlikely situation that one card is compromised, no other cards would be compromised. Use static keys only for test systems; you must use diverse keys when you issue production cards.
- **Automatically Generate Encryption Key on HSM** – the PIV 9B master key, used to derive the keys for the cards, is randomly generated on your HSM. It is a requirement of FIPS 201 that you generate keys on a FIPS 201-approved HSM for your PIV system.

4.1.3 Implementation

Use the **Key Manager** workflow to set up your customer PIV 9B keys, using the **Key Diversity: Diverse** and **Automatically Generate Encryption Key on HSM** options.

See the *PIV card application administration key (9B)* section of the **[PIV Integration Guide](#)** for details of using the **Key Manager** workflow.

To verify that the system has been configured correctly, issue a card, then examine the audit logs for the issuance. A row should appear in the audit logs indicating that the PIV 9B keyset was changed to `Customer`.

4.1.4 Considerations

When you issue a card with a customer PIV 9B key, if you intend to use the card on a different MyID installation, you must first cancel the card on the system on which it was issued – this changes the PIV 9B key back to the factory setting.

4.1.5 Recommendations

- You must configure your system for customer PIV 9B keys before your production system goes live.
- You must set up the PIV 9B keys to be diversified and HSM-generated.
- If you add a new device type to your system, you must set up the customer PIV 9B key for it separately.
- Use the audit logs to confirm that the PIV 9B keys are being changed to customer values.

4.2 GlobalPlatform key sets

A GlobalPlatform key set is a set of symmetric keys on every GlobalPlatform card – which includes most PIV cards. Its exact usage depends on the device type, but in general the GlobalPlatform key is required to carry out key management operations and activations on the cards over a secure channel.

Cards are delivered with a factory GlobalPlatform key. You must set up MyID with the factory GlobalPlatform key for the appropriate device type. This allows MyID to carry out operations such as card activation, changing the PIV 9B key, and working with archived certificates.

4.2.1 Risks

- The factory GlobalPlatform key may be the same on other cards of that type; therefore, it is possible that it is known to unauthorized parties.

- Some cards are manufactured with dedicated factory keys specific to the end customer that may also be diversified; in this situation, however, the card manufacturer knows the key for each card, and you have no control over their information security.

- An unauthorized party with the GlobalPlatform key can modify the content of the card.

4.2.2 Solution

Set up MyID to replace the factory GlobalPlatform key with a customer GlobalPlatform key – this is a key known only to the customer's system. Unauthorized parties will not have access to this customer GlobalPlatform key, and therefore cannot perform any unauthorized modifications of the cards issued by MyID.

For further security, you can set the following options on your customer keys:

- **Key Type: Diverse** – each card is issued with a different key, derived from a master key. Even in the unlikely situation that one card is compromised, no other cards would be compromised.

- **Automatically Generate Key In HSM** – the GlobalPlatform master key, used to derive the keys for the cards, is randomly generated on your HSM.

4.2.3 Implementation

Use the **Manage GlobalPlatform Keys** workflow (**Manage Open Platform Keys** workflow on older systems) to set up your customer GlobalPlatform keys, using the options for diversification and HSM key generation.

You must set up a customer key for each algorithm; for example, if SCP01 IDPrime PIV cards are issued, you must create a 2DES customer GP key which will be used for those cards, but if OT-SCP03 Oberthur ID-One cards are issued on the same system you must also create an AES128 customer key.

Note: The *Smart Card Integration Guide* contains tables detailing the appropriate combinations of secure channels, algorithms, and cryptographic key types for GlobalPlatform factory and customer keys for your particular type of smart card. In general:

Secure Channel Type (Factory tab)	Key Algorithm (Customer tab)
SCP01/SCP02	2DES
OTSCP03	AES128
SCP03	AES128/AES192/AES256, depending on the algorithm chosen on the Factory tab.

For 10.2 systems and later:

- In the **Security Settings** workflow, on the **Device Security** page, set the **Enable Customer GlobalPlatform Keys** option to `Yes`. If you do not have this option set, MyID will not attempt to write customer GlobalPlatform keys to your cards.

For systems before 10.2:

- Within the **Operation Settings** workflow, on the **Devices** page, set the **Java Card Keyset** options to `Yes`. If you do not have the Java Card Keyset option set, MyID will not attempt to write customer GlobalPlatform keys to your cards.

Make sure you set the **Version** numbers of your factory and customer keys correctly, according to the instructions in the *Managing GlobalPlatform keys* section in the *Administration Guide*:

- The factory key version number should be available from your card manufacturer and will be a number between 0 and 127 or 255. A version of 255 should normally be used for cards delivered with an Initial Keyset.

- The customer key version number must be a different value from the version entered for any factory keyset; otherwise, the custom GlobalPlatform keyset will not be written to cards with that factory keyset. The highest allowed customer key version is 127.

- When the factory key version is configured, it is instructing MyID what the key version is on the cards when they are presented to MyID (fresh from the factory). However, the configuration of the customer key version is to set the key version that will be written to the card when MyID replaces the factory key with the customer key.

- If you have specified a factory keyset version of 255, you cannot use a customer keyset version of 1; otherwise, the custom GlobalPlatform keyset will not be written to cards with that factory keyset.

- You are recommended not to use a customer keyset version of 1, as many cards have factory key version 1 or 255.

- The customer keyset version must be different from the value entered for any other Key Algorithm; for example, you can have version 2 for 2DES and version 3 for AES128.

See the *Managing GlobalPlatform keys* section in the *Administration Guide* for details (*Manage Open Platform Keys* in older versions).

To verify that the system has been configured correctly, issue a card, then examine the audit logs for the issuance. A row should appear in the **Audit Reporting** workflow indicating that the GlobalPlatform keyset was changed to Customer.

4.2.4 Considerations

When you issue a card with a customer GlobalPlatform key, if you intend to use the card on a different MyID installation, you must first cancel the card on the system on which it was issued – this changes the key back to the factory setting.

4.2.5 Recommendations

- You must configure your system for customer GlobalPlatform keys before your production system goes live.

- You must set up the GlobalPlatform key to be diversified and HSM-generated.

- Use the audit logs to confirm that the GlobalPlatform keys are being changed to customer values.

5 Passwords

5.1 Passwords for startup users

When you install MyID, you are given the option of creating startup users that can access the system using a standard set of passwords rather than using smart cards to log on. Some versions of MyID use the installation program to create the startup users, while later versions use GenMaster.

These startup users are intended only for bootstrapping the system.

5.1.1 Risks

Usernames and passwords created by the MyID installation program are identical across all MyID systems, and are listed in the MyID documentation. If you leave the startup users active, anyone who knows the startup usernames and passwords on any MyID system will be able to access your system.

Passwords created by GenMaster for the startup user are specified when you run the program; however, the startup username may still be known.

5.1.2 Solution

Once you can issue operator cards successfully, you can enroll a user and issue a physical card for each role; once you have done this, you must delete the startup users from the system.

5.1.3 Implementation

To remove the startup user, from the **People** category, click **Remove Person**.

5.1.4 Recommendations

As soon as you can issue operator cards, issue cards for each role, then delete the startup password users.

If you do not intend to allow any users to log on with passwords, you can prevent any access using security phrases: set the following configuration option in the **Logon Mechanisms** tab of the **Security Settings** workflow:

- **Password Logon** – `No`

If you want to allow password or authentication code logon to MyID for the purpose of PIN resets, but not for general logon, you can prevent password logon for cases where the user does not also have their card present: set the following configuration option in the **Logon** tab of the **Security Settings** workflow:

- **Prevent Direct Password Logon** – `Yes`

Note: If you need to recover a GenMaster-based startup user account, you can use the Recover Startup User utility; see the *Recover Startup User* section in the ***Implementation Guide***.

6 Backups

This section contains details of backup considerations.

6.1 HSM backups

In addition to your database, application server, and web server backup strategies, when an HSM is used for cryptographic security (for example, to generate customer keys or to store the master MyID key) you must make sure that your HSM is backed up, and that the procedures to restore the data from backup are documented and tested.

6.1.1 Risks

If your HSM fails, you must restore from your backup to a new HSM. If you cannot carry out this restoration, the keys that are required to manage the cards that have already been issued will be lost, and the master MyID key that allows you to log in to MyID will not be available.

If this happens, MyID will not be able to manage any of your previously-issued cards.

6.1.2 Solution

Make sure your HSM is backed up, and the PINs and cards used to secure the backup are stored safely.

6.1.3 Implementation

See your HSM documentation and contact your HSM vendor for advice on making and securing your HSM backup. You must also make sure that you can restore your HSM data from backup.

6.1.4 Recommendations

Implement a backup solution for your HSM.

7 Website Security

Setting up SSL on your web server prevents any interception of information sent to and from the MyID website.

You must review the following security considerations for the MyID website:

- SSL/TLS on the MyID website
- MyID server-to-server web services
- Firewall
- Secure session cookie
- Click jacking
- Remove details of the IIS server from the HTTP headers

Some of the suggestions in this section apply to more than the MyID website. There is a choice between applying these to the individual MyID websites or to the **Default Web Site**. If there no websites other than those belonging to MyID, you are recommended to apply the changes to the **Default Web Site**. Otherwise the IIS administrator is recommended to make an informed choice based on the requirements of the other websites that share a server with MyID.

7.1 MyID website

On production environments you must use TLS on your MyID website.

7.1.1 Risks

Traffic sent to and from the MyID website may be vulnerable to interception. MyID encrypts some sensitive data, but for full protection you must use SSL/TLS.

IIS provides the SSL/TLS transport layer security that is used by the MyID application. Over time there have been many versions of the SSL/TLS protocols, and many cipher suites available within each protocol. This allows web servers to be flexible and support a wide range of clients – when a client connects, a mutually supported SSL/TLS protocol version is agreed and a mutually supported cipher suite agreed as part of the initial handshake.

Some of these SSL/TLS protocols and cipher suites supported by IIS are stronger than others. The exact version of the protocol and cipher suites that are intended to be supported for a given installation of MyID depend on which clients must be supported.

IIS allows selected versions of the SSL/TLS protocol and cipher suites to be disabled – this configuration guarantees that older/weaker versions of the protocol/cipher-suite cannot be used. For more information, see section *10, Securing MyID with TLS 1.2*.

7.1.2 Solution

Implement SSL on your MyID website.

Review which SSL/TLS protocols and cipher suites are intended for use by the deployment, and disable unwanted SSL/TLS protocols and cipher suites.

Since MyID version 10.0, additional web services are installed that are intended to be accessed by end clients (for example, desktop PCs, mobile phones, and so on).

The following client web services are installed by MyID 10.0 or later:

- MyIDProcessDriver
- MyIDDataSource

Additionally, to provide backwards compatibility with older devices, older versions of these web services may be installed. These exist in version-numbered subfolders of the MyIDDataSource and MyIDProcessDriver web services.

The following IIS screenshot shows the MyID virtual directory, and also the MyIDDataSource and MyIDProcessDriver web services both under the **Default Web Site**.

IIS must be configured so that each of these folders requires SSL. While it is possible to configure this for each individual virtual directory, it is more efficient to configure the SSL requirements at the **Default Web Site** level, which means that this setting will be inherited by all virtual directories underneath.

The rest.core and web.oauth2 web services (used for the MyID Operator Client) are present from MyID 11.6 onwards. These use OAuth2 (rfc6749) which mandates the use of TLS. Therefore these components are configured out of the box to require TLS. For these to function, you must set up IIS with a TLS certificate.

If it is necessary to use rest.core and web.oauth2 without TLS (for example, in a developer environment) additional configuration of rest.core and web.oauth2 is required to allow them to function without TLS.
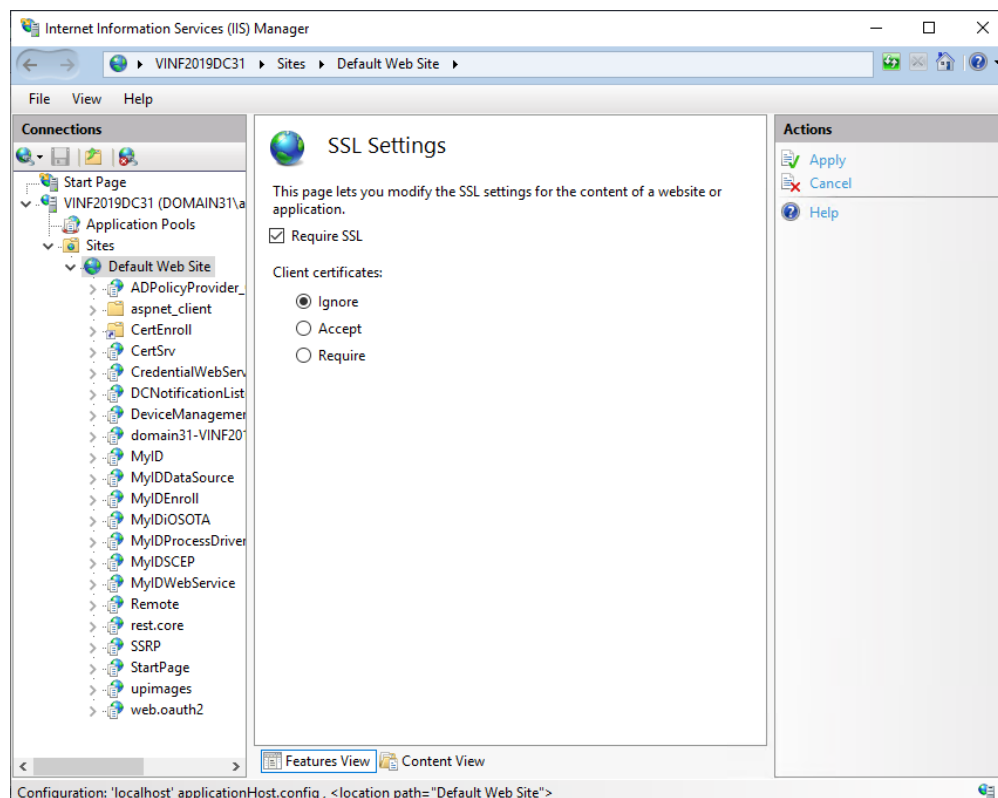
7.1.3 Implementation

See your IIS documentation for details of setting up SSL. For example, in IIS 7 or IIS 8, you must:

1. Obtain an appropriate certificate.

2. Create an HTTPS binding for the website using this certificate.

3. Set the **SSL Settings > Require SSL** option for the **Default Web Site**.

The disabling of SSL/TLS protocols and cipher suites is an IIS configuration (not part of the MyID application itself). For more information, see your Microsoft documentation.

7.1.4 Recommendations

Set up SSL on your web server, and require SSL on your IIS website hosting MyID.

7.2 MyID server-to-server web services

This section describes locking down access to server-to-server web services. While the client web services described above are intended to be accessed by clients, these server-to-server web services are intended to be accessed only by trusted servers. End clients will not normally be allowed to access these web services.

For information on how to lock down the web services and sites, see the *Securing Websites and Web Services* document.
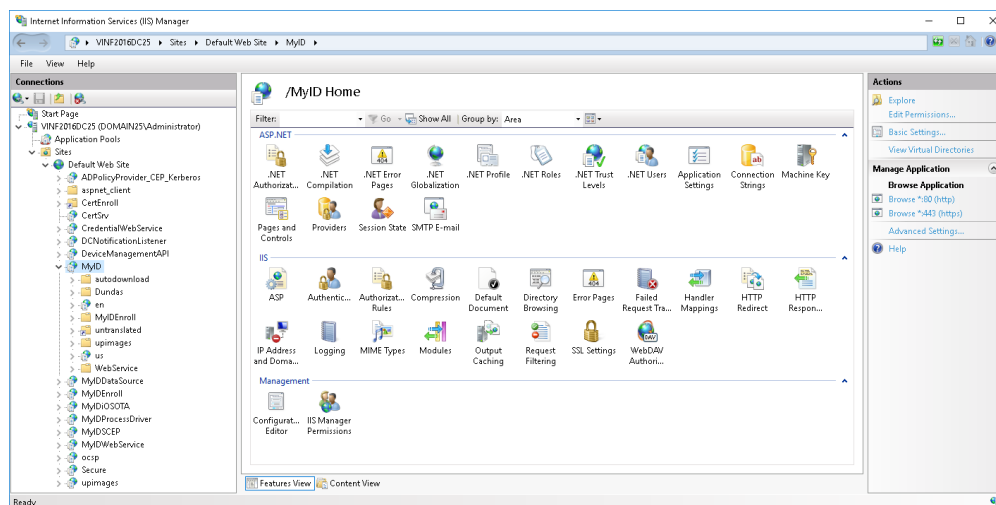
The above screenshot shows the directory structure on the web server.

Note: Depending on the specific version of MyID you are running, and the options selected during your installation, you may not have all of the folders listed.

Conversely, if you have additional customizations, you may have additional web service folders or virtual directories on your system. In this case, the information provided here concerned with locking down the standard MyID web services will also apply to additional custom web services.

On older MyID systems prior to MyID 10.0 there is an import folder under MyID.

This means that there are potentially two routes to access some of these features; for example, you can access the import ASP page using:

```
http://myserver.example.com/import/import.asp
```

or:

`http://myserver.example.com/MyID/import/import.asp`

You must make sure you lock down both routes to prevent unauthorized access; for example, you can disable Anonymous Authentication permissions for the import folder.

The virtual directories for these features are separate to the MyID website virtual directory; however, the files are stored within the MyID Web folder on the web server.

7.2.1 Risks

If you do not lock down access to the import features and server API web services, unauthorized users might be able to create users, update users, enable or disable users, request cards, or retrieve management data.

7.2.2 Solution

Set up a secure authentication method for the virtual directories by using the security features of IIS: enforce 2-way SSL, configure Windows authentication, or limit incoming IP addresses.

You must block access to the import folder within the MyID virtual directory; this folder is set up for anonymous access by default.

7.2.3 Implementation

See your IIS documentation for details of setting the security on the following virtual directories, if present:

- import
- MyIDEnroll
- WebService

The SOAP-based web services (for example, MyIDEnroll and WebService) can be accessed only through their own virtual directories. However, the import web service is ASP-based, and can be accessed either through its own virtual directory, or through the folder in the main MyID virtual directory. To prevent access to the import web service in the MyID virtual directory, select the import folder (rather than the import virtual directory) in IIS Manager and disable Anonymous Authentication along with any other enabled authentication mechanisms. This ensures that all access to the import web service is through the import virtual directory, for which you can set security according to your IIS documentation.

7.2.4 Recommendations

Set up two-way SSL on your server-to-server web service virtual directories.

If you are not using the import features, remove these virtual directories from your system.

Block access to the import folder within the MyID virtual directory on versions of MyID before 10.0.

7.3 Firewall to protect MyID website

You must set up a firewall to protect the MyID web server against unwanted network traffic from the external network.

7.3.1 Risks

Computers on the network could attempt to mount an attack on the MyID web server machine.

7.3.2 Solution

Setup a firewall to protect the MyID web server against network traffic that is not required for operation of MyID.

7.3.3 Implementation

A typical production setup will allow only the https port (by default 443) from the external network through to the web server.

By allowing only the required ports into the web server, the potential attack surface of the web server machine is greatly reduced.

Depending on your system and customizations, you may be using additional features that require different ports to be open.

Note: Confirm that your environment's security (for example, your load balancer or firewall) has been configured to allow full access to the REST web services; while some systems may be locked down to allow only GET and POST, the MyID web services require the full range of verbs, including (but not limited to) GET, POST, PATCH, OPTIONS, and DELETE.

7.3.4 Recommendations

Protect your web server from unwanted traffic from the external network with the use of a firewall.

7.4 Secure session cookie

The session cookie mechanism is built into IIS, and is therefore a web infrastructure issue rather than an application issue.

You can configure IIS to add `Secure` and `HttpOnly` attributes to the cookie:

- The `Secure` attribute on the cookie tells the browser to send the cookie only when https (TLS) is used.

- The `HttpOnly` attribute on the cookie tells the browser to prevent client side script code from accessing the cookie.

7.4.1 Implementation

For the `Secure` attribute to work for session cookies, TLS must be configured, and the MyID and MyIDProcessDriver applications in IIS must have been configured (under **SSL Sessions**) to **Require SSL**.

To configure IIS to add the `Secure` attribute to the sessions cookie:

1. Configure the IIS property `KeepSessionIdSecure`.

    This is shown in the IIS interface under **ASP>Session Properties** as **New ID On Secure Connection**.

    This option also means that if you switch between HTTP and HTTPS you get a new session cookie.

    Note: This property defaults to true.

Apart from the session cookie, MyID does not set cookies in response headers. To configure IIS to use `HttpOnly` session cookie:

1. Install URL Rewrite from Microsoft's iis.net website:

   www.iis.net/downloads/microsoft/url-rewrite

2. Edit the `web.config` file for the following IIS applications:

   - `MyID` – by default, the `web.config` file is in the following folder:

     `C:\Program Files\Intercede\MyID\Web\`

   - `MyIDProcessDriver` – by default, the `web.config` file is in the following folder:

     `C:\Program Files\Intercede\MyID\SSP\MyIDProcessDriver`

   - `MyIDDataSource` – by default, the `web.config` file is in the following folder:

     `C:\Program Files\Intercede\MyID\SSP\MyIDDataSource`

If the `web.config` file does not exist, you must create it. If the file already exists, merge the additional changes below into the existing content.

Edit each `web.config` file so that it contains the following content.

```xml
<configuration>
  <system.webServer>
    <rewrite>
      <outboundRules>
        <rule name="Add HttpOnly" enabled="true">
          <match serverVariable="RESPONSE_set_cookie" pattern=".*" />
          <action type="Rewrite" value="{R:0}; HttpOnly" />
          <conditions>
            <add input="{RESPONSE_set_cookie}" pattern=";
HttpOnly" negate="true" />
            <add input="{RESPONSE_set_cookie}" pattern="." />
          </conditions>
        </rule>
      </outboundRules>
    </rewrite>
  </system.webServer>
</configuration>
```

3. For the `web.config` file in the MyIDProcessDriver folder, make the following additional change:

   - Under the `<system.web>` section, ensure the following node is present:

     `<httpCookies httpOnlyCookies="true" requireSSL="true" />`

7.4.2 Recommendations

Make sure that `KeepSessionIdSecure` has not been changed from the default.

For all systems, edit the `web.config` file to set `HttpOnly`.

7.5 Prevent click jacking

You can configure IIS to prevent click jacking.

7.5.1 Implementation

On the MyID website, add a custom header: `x-frame-options=sameorigin`

In the `web.config` file it will look like this:

```
<configuration>
  <system.webServer>
    <httpProtocol>
      <customHeaders>
        <add name="x-frame-options" value="sameorigin" />
      </customHeaders>
    </httpProtocol>
  </system.webServer>
</configuration>
```

If you are running the MyID web from within another site, you must add the other website to the header; for example:

```
<add name="x-frame-options" value="sameorigin; allow-from
https://myserver/customerApp" />
```

Only one other site can be specified.

7.5.2 Recommendations

Set up IIS to prevent click jacking using the configuration file.

7.6 Remove details of the IIS server

Edit the `web.config` file on these websites:

- MyID: `C:\Program Files\Intercede\MyID\Web\WebPIV\web.config`

- MyIDDataSource: `C:\Program Files\Intercede\MyID\SSP\MyIDDataSource\Web.config`

- MyIDProcessDriver: `C:\Program Files\Intercede\MyID\SSP\MyIDProcessDriver\Web.config`

If the file does not exist, you must create it. If the file already exists, merge the additional changes below into the existing content.

Note: Older versions of MyID may contain a web service called `CertificateCheck`. You must make the same changes on this website if it exists.

Install URL Rewrite from Microsoft's iis.net website:

www.iis.net/downloads/microsoft/url-rewrite

7.6.1 Remove the Server header

```
<configuration>
  <system.webServer>
    <security>
      <requestFiltering removeServerHeader="true" />
    </security>
  </system.webServer>
</configuration>
```

7.6.2 Remove the X-Powered-By header

```xml
<configuration>
  <system.webServer>
    <httpProtocol>
      <customHeaders>
        <remove name="X-Powered-By" />
      </customHeaders>
    </httpProtocol>
  </system.webServer>
</configuration>
```

7.6.3 Remove the X-AspNet-Version header

```xml
<configuration>
  <system.webServer>
    <rewrite>
      <outboundRules>
        <rule name="Remove Server">
          <match serverVariable="RESPONSE_X_AspNet_Version" pattern=".*" />
          <action type="Rewrite" />
        </rule>
      </outboundRules>
    </rewrite>
  </system.webServer>
</configuration>
```

7.6.4 Recommendations

You are recommended to set up IIS to prevent this information from being provided.

7.7 Blocking HTTP host header injection

HTTP host header injection is a mechanism where an attacker can try to trick a web server or web service that they are operating on a different web domain than they are; this is an attempt to subvert the behavior of the web server.

You can configure IIS to filter incoming requests through the host header to reject requests that do not have the expected header value.

7.7.1 Implementation using URL Rewrite

You can use the URL Rewrite module to block HTTP host header injection. If it is not already installed on your system, you can install URL Rewrite from Microsoft's iis.net website:

www.iis.net/downloads/microsoft/url-rewrite

To block HTTP host header injection using URL Rewrite:

1. On each MyID web server, open the Internet Information Service (IIS) Manager.

2. Select the web site under which the MyID websites and services are installed.

   By default, this is the **Default Web Site**.

3. Double-click **URL Rewrite**.
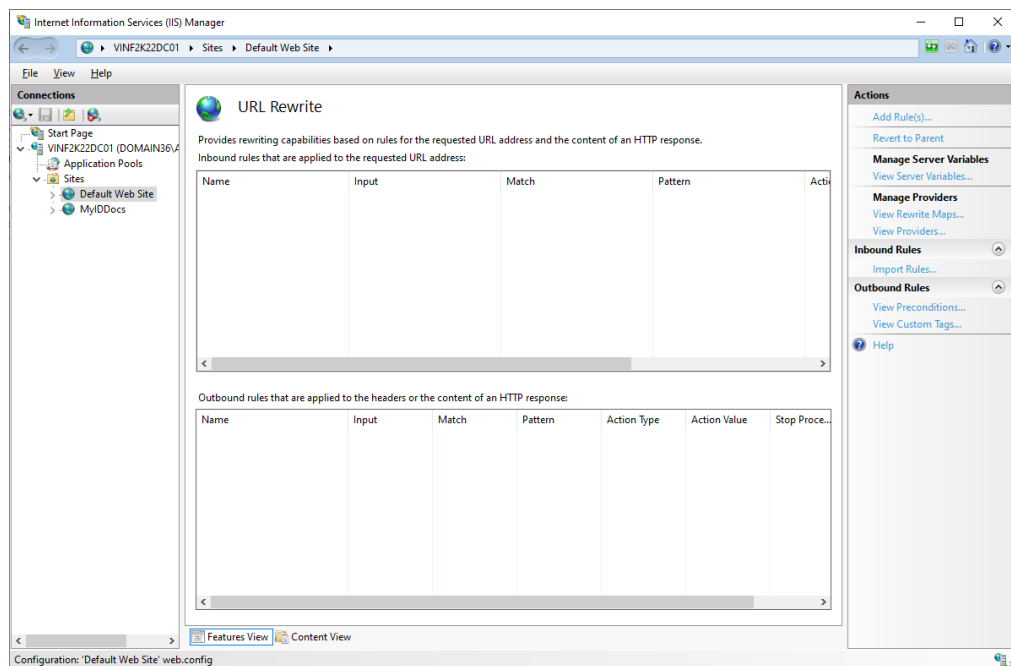
4. Click **Add Rule(s)**.

5. In the **Inbound rules** section, select **Request blocking**.
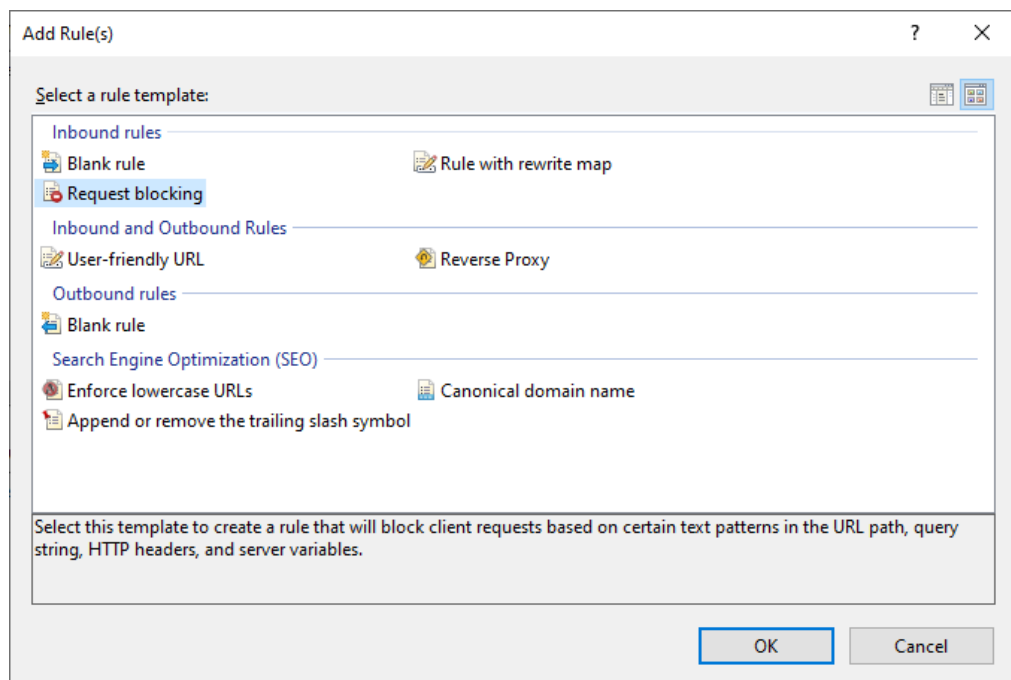
6. Click **OK**.

    The Add Request Blocking Rule dialog appears.

7. Set the options for the blocking rule.

    The options you set depend on your system configuration. See:

- section *7.7.1.1*, *Allowing a single host header*.

- section *7.7.1.2*, *Using a regular expression for multiple domains*.

8. Click **OK**.

7.7.1.1 Allowing a single host header

If you are filtering to allow only a single (or single wildcarded) host header, set the following in the Add Request Blocking Rule dialog:

- **Block access based on** – select **Host Header**.

- **Block request that** – select **Does not match the pattern**.

- **Using** – select **Wildcards**.

- **Pattern (Host Header)** – Type the expected web domain that incoming HTTP requests will use; for example, if HTTP requests are expected on:

  `https://myid.mycompany.com`

  type:

  `myid.mycompany.com`

  If required, you can use `*` as a wildcard; for example:

  `*.mycompany.com`

  to allow any subdomain of `mycompany.com`.

Note: Matching is case insensitive by default.

- **How to block** – Leave this at the default: **Send an HTTP 403 (Forbidden) Response**.

7.7.1.2 Using a regular expression for multiple domains

You may have more complicated matching rules. In this case, you can use regular expressions.

For example, If you have a load balancer, you may have most requests coming in using the load balancer web domain, but still want to allow connections directly to the web servers as well; in this case you would have two allowable host header values.

The following example shows the basic regular expression for matching a single value:

- **Block access based on** – select **Host Header**.

- **Block request that** – select **Does not match the pattern**.

- **Using** – select **Regular Expressions**.

- **Pattern (Host Header)** – Type the regular expression that the host header must match to be accepted.

  Note: The regular expression matching is case insensitive by default.

  Example regular expression for matching a single domain:

  ```
  ^myid\.mycompany\.com$
  ```

Note: ` ^ ` and ` $ ` mean that no prefix or suffix is allowed, any period (`.`) character must be escaped with a backslash (`\`).

- **How to block** – Leave this at the default: **Send an HTTP 403 (Forbidden) Response**.

The following example provides a list of allowable domains (in this example, `websrv1.mycompany.com` or `loadbalancer.mycompany.com`):

`^websrv1\.mycompany\.com$|^loadbalancer\.mycompany\.com$`

Note: The pipe (`|`) character is used as a separator for each allowable value.

You can specify any valid regular expression to implement more flexible rules as required.

7.7.2   Implementation for ASP.NET Core applications

For ASP.NET Core applications you can, as an alternative, configure the `AllowedHosts` entry in the `appsettings.production.json` file to perform the host filtering for each ASP.NET Core application.

See:

learn.microsoft.com/en-us/aspnet/core/fundamentals/servers/kestrel/host-filtering?view=aspnetcore-8.0

However, if you are using URL Rewrite to configure this protection at the IIS level, there is no benefit in also configuring it for each ASP.NET Core application.

7.7.3 Recommendations

Use URL Rewrite to force IIS to block requests that do not have the expected HTTP host headers.

8 Hardening Configuration

8.1 Visibility of user data

MyID has a feature for showing the name and photograph of the user during the MyID Desktop or MyID Operator Client logon process; that is, when you insert a smart card.

Since this occurs before successful authentication, an attacker could attempt to use this feature to harvest names and photographs of users of the system. By default, this feature is not enabled.

8.1.1 Implementation

For production environments, ensure this feature is disabled:

1. Within MyID, from the **Configuration** category, select **Security Settings**.

2. On the **Logon** tab, set the following options:

   - **Show Full Name at Logon** – ensure this option is set to `No`.

   - **Show Photo at Logon** – ensure this option is set to `No`.

3. Click **Save changes**.

9 Securing the Database

You must consider the following database security features:

- The Database Master Key
- Database communications

9.1 Database Master Key

MyID encrypts sensitive data that is stored in the database using the Master Key.

The Master Key is generated when GenMaster is first run, after MyID is installed.

9.1.1 Risks

- An unauthorized party could try to extract or copy the Master Key.
- The Master Key uses a cryptographic algorithm that is broken in the future.
- Regulatory compliance may demand a more modern cryptographic algorithm for the Master Key.

9.1.2 Solution

The risk of an unauthorized party copying the Master Key is best solved by ensuring that the Master Key is generated within an HSM. If at the time the MyID system was installed, an HSM was not available, and later you want to upgrade it to use an HSM, it is possible to migrate this to the HSM.

The risk of the cryptographic algorithm being broken in the future is addressed by MyID supporting upgraded cryptographic algorithms, and supporting the migration from one Master Key to a new Master Key with a new algorithm.

- Versions of MyID that were first installed before MyID 10.4 generated a 3DES (3TDEA) Master Key.
- Versions of MyID that were first installed at version 10.4 or later generated an AES256 Master Key. Note that AES256 is the stronger and more modern algorithm.
- When upgrading a version of MyID that was installed before MyID 10.4 to 10.4 or later, the Master Key remains as 3DES (3TDEA).
    - There is a process available that can convert an upgraded MyID 10.4 or later installation to use an AES256 Master Key even if it started with a 3DES (3TDEA) Master Key. Intercede recommends this process to ensure that all production MyID installations at 10.4 or later use an HSM-protected AES256 Master Key.

9.1.3 Implementation

When first installing MyID, ensure an HSM is used to protect the Master Key.

If upgrading a production MyID installation that was installed before MyID 10.4 to 10.4 or later, upgrade the system to have an AES256 Master Key. For more information on this process, contact Intercede support quoting SUP-193.

If you have a production MyID Installation that does not currently use an HSM to protect the Master Key, Intercede recommends that this is upgraded to use an HSM. For more information on this process, contact Intercede support quoting reference SUP-193.

9.2 Database communications

The MyID application server communicates with the MyID database over OLE DB, and this communication is secured by TLS.

The latest version of TLS supported in Microsoft Windows is TLS 1.2, which is not currently supported by MyID without further configuration. For information about securing your MyID system with TLS 1.2, see section *10*, *Securing MyID with TLS 1.2*.

10 Securing MyID with TLS 1.2

The MyID application server communicates with the MyID database over OLE DB, and this communication is secured by TLS. You are recommended to set up your system to use TLS 1.2; this involves configuring the MyID application servers to ensure that they can use TLS 1.2, and configuring the MyID web servers to disable SSL and versions of TLS earlier than TLS 1.2.

10.1 Risks

Over time, the SSL/TLS protocols have evolved. It is possible that security risks may be found in older versions. The latest version of TLS supported in Microsoft Windows is TLS 1.2, which is not currently supported by MyID without further configuration.

10.2 Solution

Configure the MyID application servers to ensure that they are capable of communicating using TLS 1.2, and configure the web servers to allow them to disable SSL and versions of TLS *earlier* than TLS 1.2, thereby forcing clients to use TLS 1.2.

10.3 Implementation

To update the registry to enable .NET 4.0 components to make TLS 1.2 connections:

1. On the MyID servers hosting the web services, open the registry editor.

   . In each of the following keys:

2. Locate the following keys:

   ```
   HKEY_LOCAL_
   MACHINE\SOFTWARE\WOW6432Node\Microsoft\.NETFramework\v4.0.30319
   ```

   and

   ```
   HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\.NETFramework\v4.0.30319
   ```

3. In each, set or create a DWORD `SchUseStrongCrypto` and set the value to `1`.

The procedure above configures MyID to allow the use of TLS 1.2. This means that your MyID system will continue to operate when you have disabled TLS versions lower than TLS 1.2. For more information about SSL/TLS, see section *7, Website Security*.

10.3.1 Disabling earlier versions of SSL/TLS

For information about disabling SSL/TLS, see your Microsoft documentation.

Note: If you are using certificate authorities that use a Java-based connector (for example, UniCERT UPI or Entrust) you must configure your Java client to use the same versions of SSL/TLS as the rest of your MyID system. For example, if you have configured IIS to disable any SSL/TLS versions below TLS 1.2, you must use the Java Control Panel > **Advanced** tab > **Advanced Security Settings** section to disable all SSL/TLS versions below TLS 1.2.

Important: For pre-MyID 11.0 versions, if you install any MyID patches on your system, you may experience problems with the installer being unable to communicate with the database if you do not re-enable TLS 1.0 – older patch installers use the previous OLE DB driver that requires TLS 1.0. After installing the patch, you can disable TLS 1.0 again.

Note: If you experience any problems on the database screen of MyID installation programs, update your SQL Server Native Client – earlier versions of the SQL Native Client may not have full support for TLS 1.2. MyID installers that support TLS 1.2 have been tested with SQL Server Native Client version 11.0.70001.0.

11 Security Checklist

✓	Security Feature	Section
	The system has been configured for random SOPINs.	section *3.1*, *SOPINs*
	The system has an appropriate PIN policy set up.	section *3.2*, *PIN complexity*
	The system has been configured for customer PIV 9B keys. (PIV cards only)	section *4.1*, *PIV 9B keys*
	The PIV 9B customer key is diversified. (PIV cards only)	
	The PIV 9B customer key is HSM-generated. (PIV cards only)	
	The customer PIV 9B key has been set up for each device type. (PIV cards only)	
	The audit logs have been checked to confirm that the PIV 9B keys are being changed to customer values. (PIV cards only)	
	The system has been configured for customer GlobalPlatform keys.	section *4.2*, *GlobalPlatform key sets*
	The GlobalPlatform key is diversified.	
	The GlobalPlatform key is HSM-generated.	
	The customer GlobalPlatform key has been set up for each device type.	
	The audit logs have been checked to confirm that the GlobalPlatform keys are being changed to customer values.	
	Startup users have been deleted from the system.	section *5.1*, *Passwords for startup users*
	Password logon has been disabled, if appropriate.	
	The HSM is securely backed-up.	section *6.1*, *HSM backups*
	The MyID website is secured with SSL.	section *7.1*, *MyID website*
	SSL/TLS protocol versions and algorithms reviewed. IIS configured to disable any unwanted SSL/TLS protocol versions and algorithms.	
	The MyID import, WebService and MyIDEnroll virtual directories are secured with two-way SSL.	section *7.2*, *MyID server-to-server web services*
	The import folder in the MyID virtual directory is made inaccessible. (Appropriate for versions of MyID before 10.0)	
	The MyID web servers have been protected from unwanted external traffic using a firewall.	section *7.3*, *Firewall to protect MyID website*

✓	Security Feature	Section
	Set up IIS for a secure session cookie.	section *7.4*, *Secure session cookie*
	Set up IIS to prevent click jacking.	section *7.5*, *Prevent click jacking*
	Set up IIS to remove details of the IIS server from the HTTP headers.	section *7.6*, *Remove details of the IIS server*
	Set up IIS to block HTTP host header injection.	section *7.7*, *Blocking HTTP host header injection*
	Confirm that the **Show Full Name at Logon** and **Show Photo at Logon** options are set to NO.	section *8.1*, *Visibility of user data*
	Set up MyID for TLS 1.2 communication with the database.	section *10*, *Securing MyID with TLS 1.2*