Day 2 Activity File: Secure Your Web Application with SSL Certificates (Azure Premium and GoDaddy Domain Version)

Today, you will secure your web application. Specifically, you will:

- (1) **Create a key vault**.
- (2) **Create a self-signed certificate**.
- (3) **Import and bind your self-signed certificate to your web app**.
- (4) **Create and bind an app service managed certificate**.
- (5) **Answer review questions**.

⚠️ Note: You will purposely create and add **two** types of SSL certificates to your web application to experience the advantages and disadvantages of each certificate.

Resources

- Azure Key Vaults
- What is a self signed certificate?
- Binding Certificates in Azure
- Azure App Service Managed Certificates
- Azure App Service Documentation
- If Microsoft Support is needed, visit How to open a support ticket

---

Getting Started/Prerequisites

Before you begin Day 2, you are required to have completed the following tasks from Day 1:

- Created your own web application.
- Created your own unique domain name.
- Deployed a Docker container to your web application.
- Customized your web application with your own unique content.

Instructions

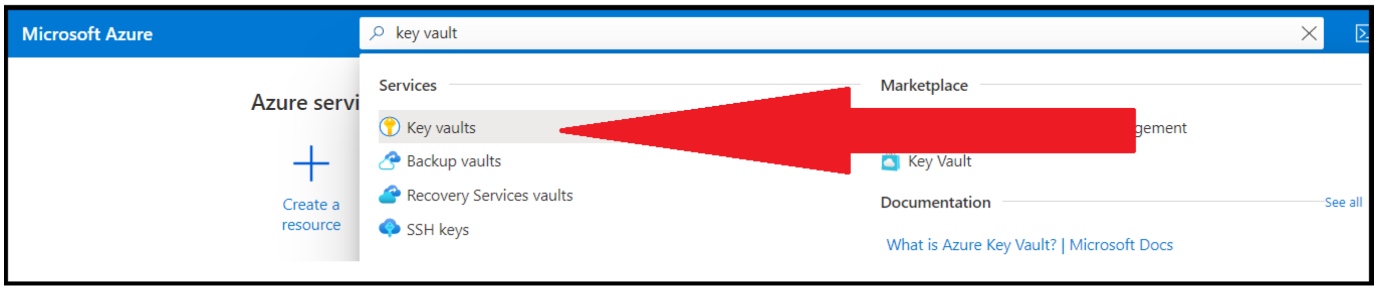
Part 1: Create a Key Vault

In this first part, you will create an Azure key vault. To do so, complete the following steps:

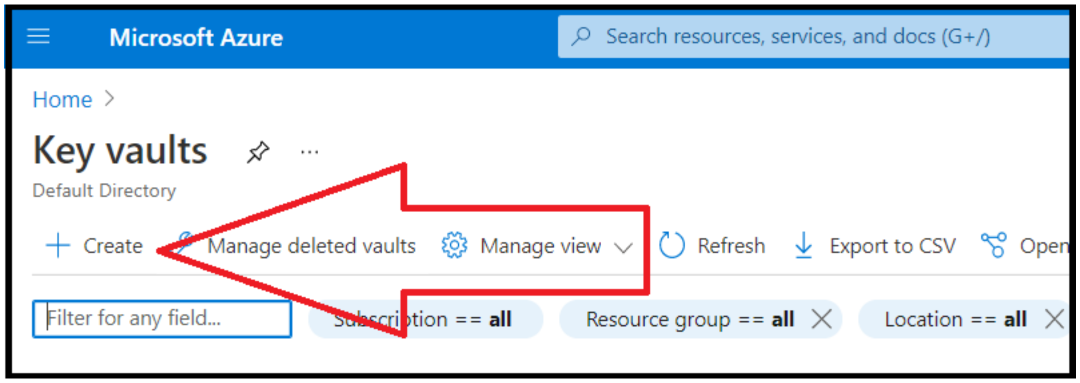
1. Begin by logging in to the Azure portal: https://portal.azure.com.

   - Make sure that you're logged in to your personal Azure account (not @Cyberxsecurity), where your Cloud Security–unit VMs are located.

2. Select "Key vaults" from the Azure search field at the top of the page, as the following image shows:



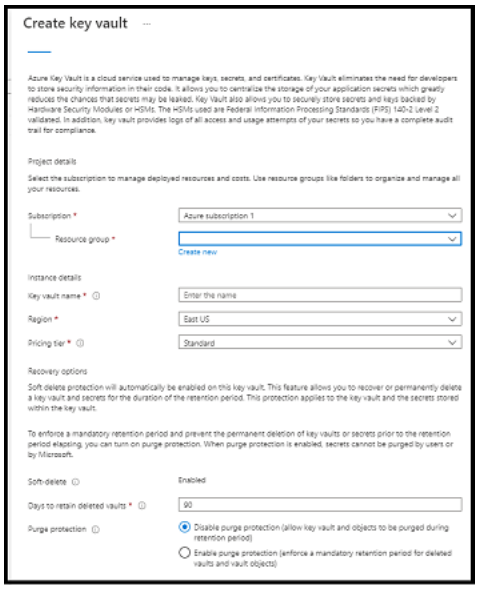
3. Select "+ Create" from the Key Vault page to create your key vault, as the following image shows:



4. On the "Create key vault" tab, make the following selections:

   - Subscription/Resource Group: Select the same subscription and resource groups that you selected on Day 1.
   - Key Vault Name: Choose a key vault name, such as `project1_KeyVault`. (Note: This name must be globally unique, so you will be prompted to choose a different name if the one you enter has been used before.)
   - Region: Select the same region that you selected on Day 1.
   - Pricing tier: Select the "Standard" tier.
   - Leave the default options for all of the other tabs (Access Policy, Networking, Tags).

   The following image shows the completed "Create key vault" tab:



Create key vault

Azure Key Vault is a cloud service used to manage keys, secrets, and certificates. Key Vault eliminates the need for developers to store security information in their code. It allows you to centralize the storage of your application secrets which greatly reduces the chances that secrets may be leaked. Key Vault also allows you to securely store secrets and keys backed by Hardware Security Modules or HSMs. The HSMs used are Federal Information Processing Standards (FIPS) 140-2 Level 2 validated. In addition, key vault provides logs of all access and usage attempts of your secrets so you have a complete audit trail for compliance.

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *: Azure subscription 1

Resource group *:

Create new

Instance details

Key vault name *: Enter the name

Region *: East US

Pricing tier *: Standard

Recovery options

Soft delete protection will automatically be enabled on this key vault. This feature allows you to recover or permanently delete a key vault and secrets for the duration of the retention period. This protection applies to the key vault and the secrets stored within the key vault.

To enforce a mandatory retention period and prevent the permanent deletion of key vaults or secrets prior to the retention period elapsing, you can turn on purge protection. When purge protection is enabled, secrets cannot be purged by users or by Microsoft.

Soft-delete: Enabled

Days to retain deleted vaults *: 90

Purge protection: [✓ Disable purge protection (allow key vault and objects to be purged during retention period)] Enable purge protection (enforce a mandatory retention period for deleted vaults and vault objects)

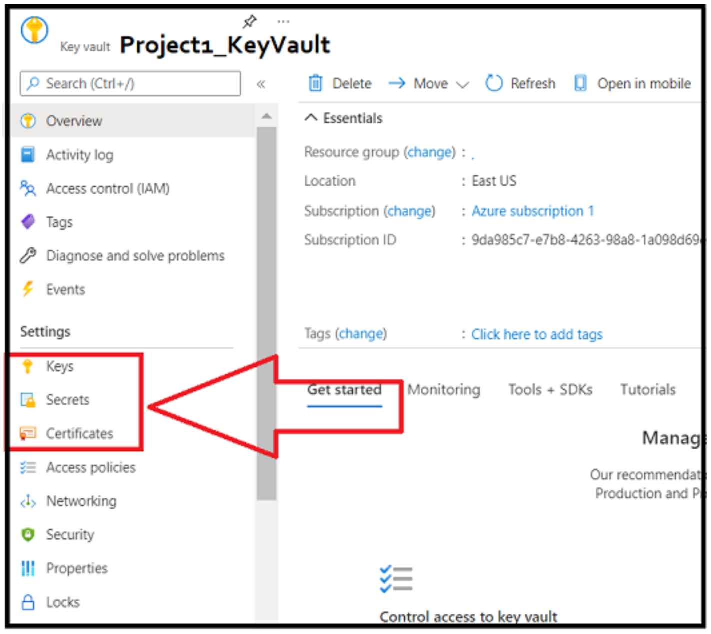
   - Finally, select "Review + Create" to create your key vault.

5. After your key vault has been created, select your new resource to view your new key vault.

6. Preview the options available on your key vault to store secure information, including:

   - **Keys**
   - **Secrets**
   - **Certificates**

   The following image shows these options:

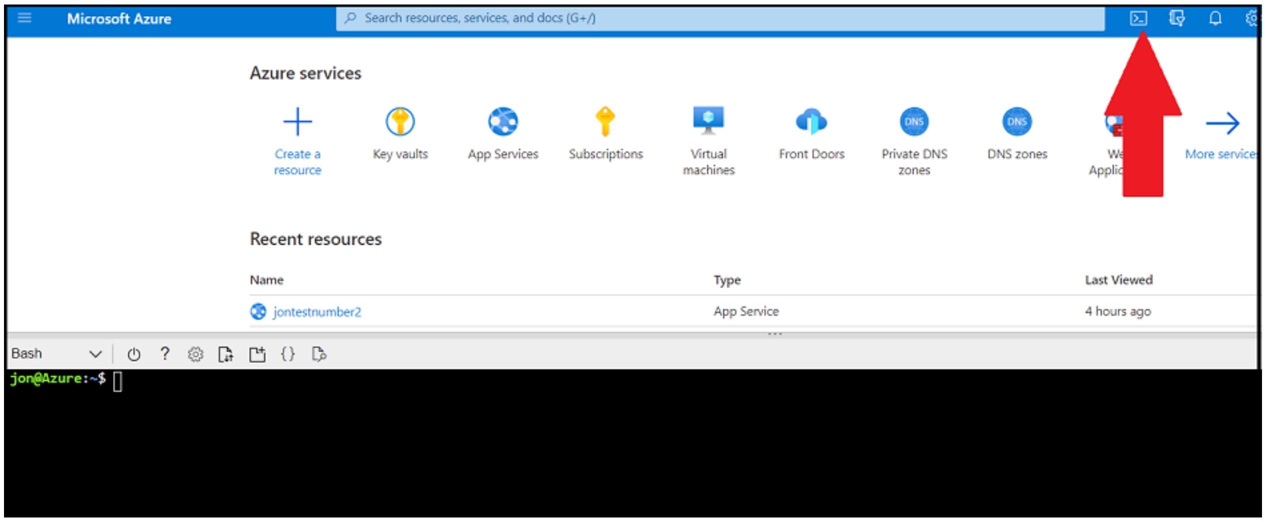


```
Resource group (change) : 
Location                : East US
Subscription (change)   : Azure subscription 1
Subscription ID         : 9da985c7-e7b8-4263-98a8-1a098d69
Tags (change)           : Click here to add tags
```

Part 2: Create a Self-Signed Certificate

In this second part, you will return to the command line to create a self-signed certificate using OpenSSL. To do so, complete the following steps:

1. From your Azure portal, access the same Cloud Shell that you accessed on Day 1 to load the Docker container, as the following image shows:

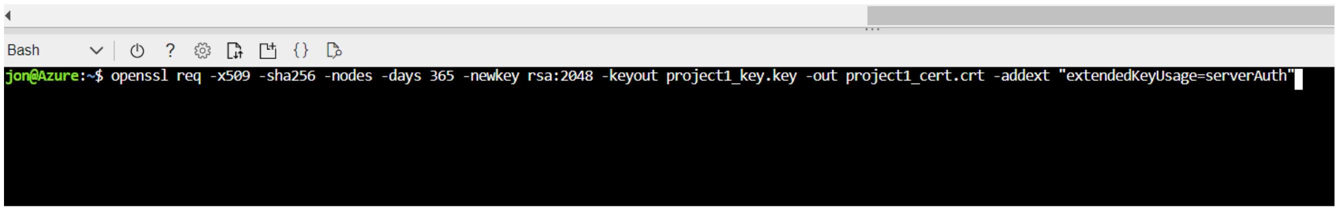


```
jon@Azure:~$ 
```

   - From this command line, you will now use the open source cryptography and SSL/TLS "toolkit" **OpenSSL** (it is preinstalled).
     - Recall that during Cryptography week, we used OpenSSL to generate keys and an IV to encrypt a message.

2. Next, you will use OpenSSL to generate a **self-signed certificate**.
   - A self-signed certificate is a certificate that has not been signed by a certificate authority.
   - These certificates are simple to create and have no expense.
   - We will explore their advantages and disadvantages in today's review questions.

3. From the command line, enter the following command:
   ```
   openssl req -x509 -sha256 -nodes -days 365 -newkey rsa:2048 -keyout <privatekeyname.key> -out <certificatename.crt> -addext "extend
   ```
   - For example:
     ```
     openssl req -x509 -sha256 -nodes -days 365 -newkey rsa:2048 -keyout project1_key.key -out project1_cert.crt -addext "extendedKey
     ```
   - The following image shows this step:



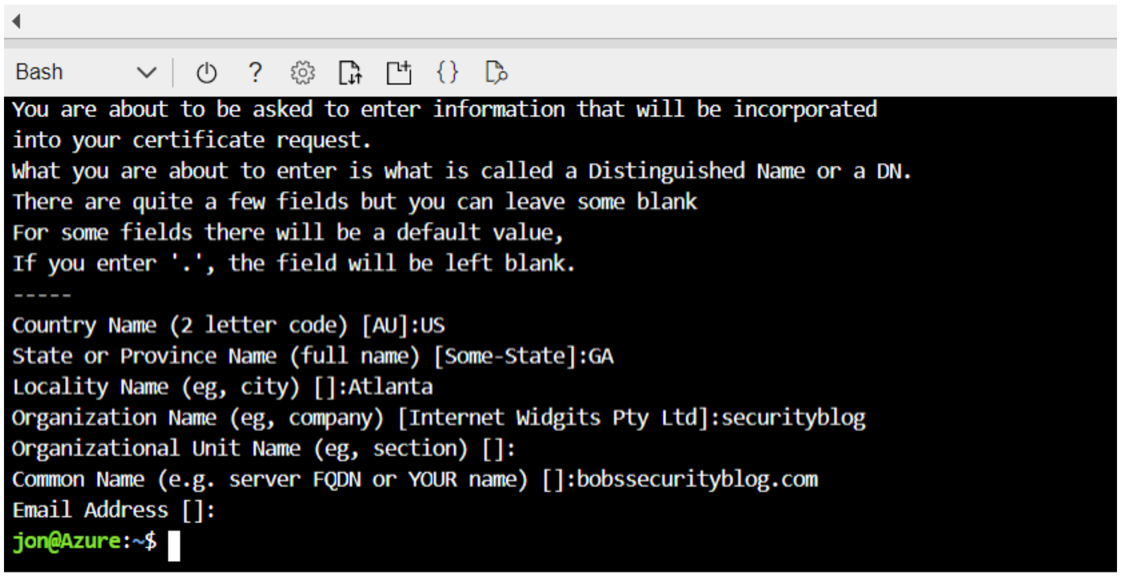
```
jon@Azure:~$ openssl req -x509 -sha256 -nodes -days 365 -newkey rsa:2048 -keyout project1_key.key -out project1_cert.crt -addext "extendedKeyUsage=serverAuth"
```

- We added the following options:
  - **-x509**: Indicates for OpenSSL to create an SSL certificate.
  - **-sha256**: Uses the sha256 hashing algorithm.
  - **-nodes**
  - **-days 365**: States the certificate will be valid for one year.
  - **-newkey rsa:2048**: Uses a 2048-bit RSA key.
  - **-keyout project1_key.key**: The outputted name of the private key.
  - **-out project1_cert.crt**: The outputted name of the certificate.
  - **-addext "extendedKeyUsage=serverAuth"**: Indicates how a public key can be used.
- Refer to the following document for additional information on OpenSSL options.

4. After pressing Enter, you will be asked several questions about your certificate. Answer the following:

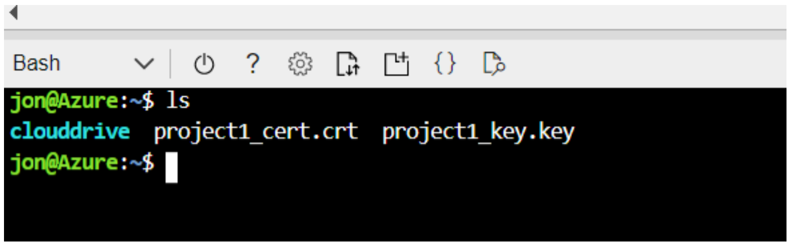
   - **Country Name (2 letter code) [AU]**: Enter your country.
   - **State or Province Name (full name) [Some State]**: Enter your state.
   - **Locality Name (e.g., city) [ ]**: Enter your city.
   - **Organization Name (e.g., company) [Internet Widgits Pty Ltd]**: Enter "Student".
   - **Organizational Unit Name (e.g., section) [ ]**: Leave blank by pressing Enter.
   - **Common Name (e.g., server FQDN or YOUR name) [ ]**: Enter your full domain name, such as "bobsblog.com".
   - **Email Address [ ]:** Leave blank by pressing Enter.

   The following image shows this step:



```
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:GA
Locality Name (eg, city) []:Atlanta
Organization Name (eg, company) [Internet Widgits Pty Ltd]:securityblog
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:bobssecurityblog.com
Email Address []:
jon@Azure:~$
```

5. Now, view your newly created key (`.key`) and certificate (`.crt`) by running `ls` , as the following image shows:



```
jon@Azure:~$ ls
clouddrive  project1_cert.crt  project1_key.key
jon@Azure:~$
```

- Note that Azure requires a PFX format for its certificates.
  - The PFX format is the server certificate and the private key combined into a single encrypted file.

6. To create a PFX format, run the following command:
   `openssl pkcs12 -export -out <new_certificatename.pfx> -inkey <keyname.key> -in <certificename.crt>`

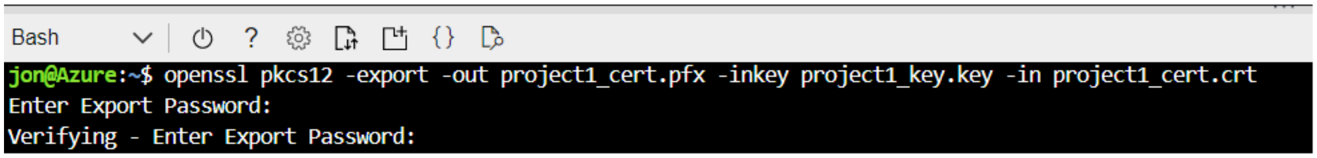
   - For example: `openssl pkcs12 -export -out project1_cert.pfx -inkey project1_key.key -in project1_cert.crt`

   - We added the following options:

     - **pkcs12**: Indicates for OpenSSL to create a PFX certificate.
     - **-export -out project1_cert.pfx**: States what to name the PFX file.
     - **-inkey project1_key.key**: This is the current private key that you are importing.
     - **-in project1_cert.crt**: This is the current certificate that you are importing.

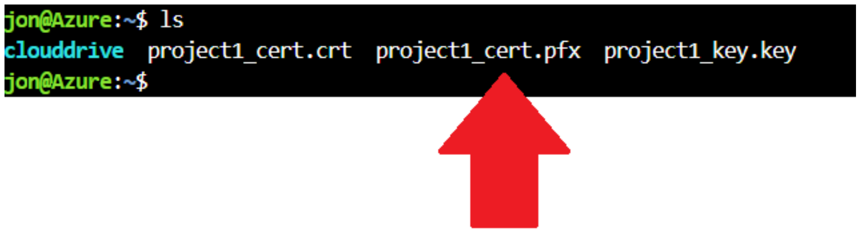
7. After pressing Enter, you will be prompted for a password to encrypt your PFX key.

   - Don't forget your password, as you will be prompted for it again shortly!
   - The following image shows this step:



```
jon@Azure:~$ openssl pkcs12 -export -out project1_cert.pfx -inkey project1_key.key -in project1_cert.crt
Enter Export Password:
Verifying - Enter Export Password:
```

8. View your new PFX certificate by running `ls`, as the following image shows:

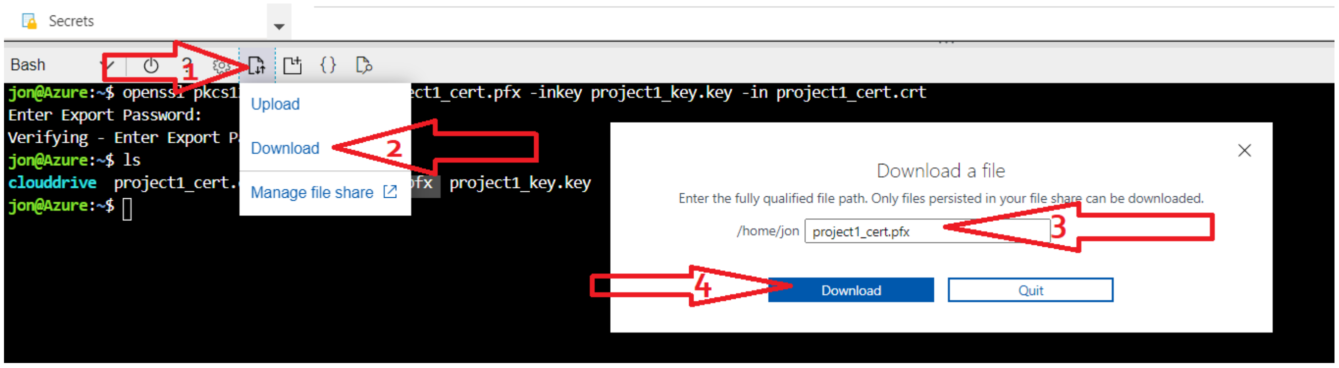


```
jon@Azure:~$ ls
clouddrive  project1_cert.crt  project1_cert.pfx  project1_key.key
jon@Azure:~$
```

9. To download your new PFX certificate, complete the following four steps:

   - (1) Click the "Upload/Download" icon in the toolbar above your Cloud Shell window.
   - (2) Select "Download."
   - (3) Enter the name of your PFX certificate in the "Download a file" window.
   - (4) Click "Download."

   The following image shows these steps:



```
jon@Azure:~$ openssl pkcs12 ...ct1_cert.pfx -inkey project1_key.key -in project1_cert.crt
Enter Export Password:
Verifying - Enter Export P
jon@Azure:~$ ls
clouddrive  project1_cert.  ...pfx  project1_key.key
jon@Azure:~$
```

Download a file

Enter the fully qualified file path. Only files persisted in your file share can be downloaded.

/home/jon project1_cert.pfx

Download Quit

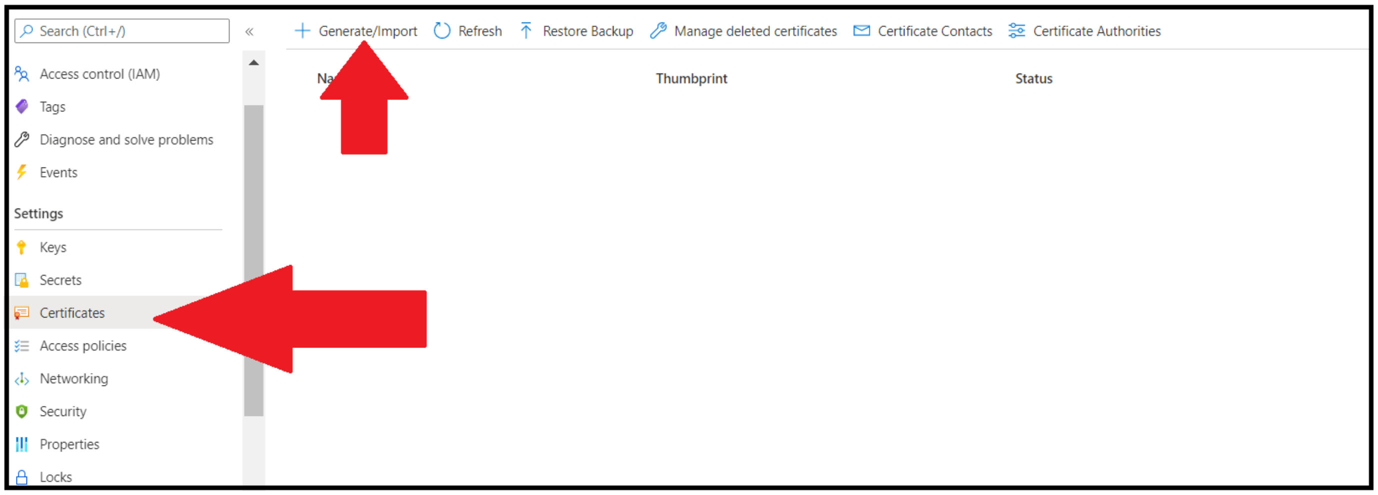
Part 3: Import and Bind Your Self-Signed Certificate to Your Web App

In this part, you will use Azure to import and bind the certificate that you just added to your web application. To do so, complete the following steps:

1. From the Azure Portal, select "Key Vaults."

   - Select the key vault that you created in Part 1.

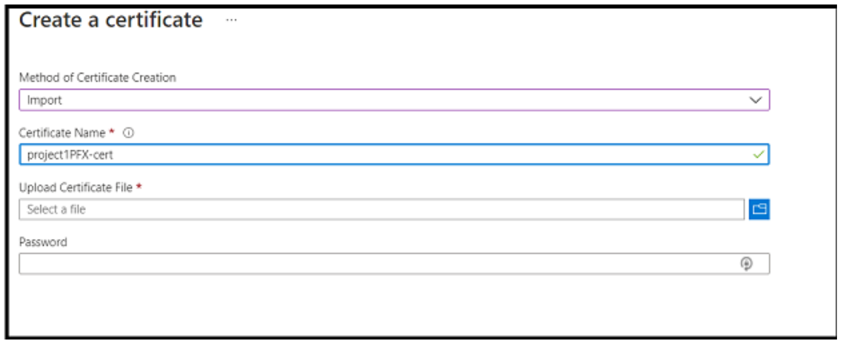
2. From your key vault, select "Certificates" and then "+ Generate/Import," as the following image shows:



3. On the "Create a certificate" page, select the following:

   - **Method of Certificate Creation**: Import
   - **Certificate Name**: project1PFX-cert
   - **Upload Certificate File**: Select your PFX certificate (it's likely in your `Downloads` folder)
   - **Password**: Enter the password that you created in Part 2

   The following image shows these steps:



Create a certificate

Method of Certificate Creation: Import

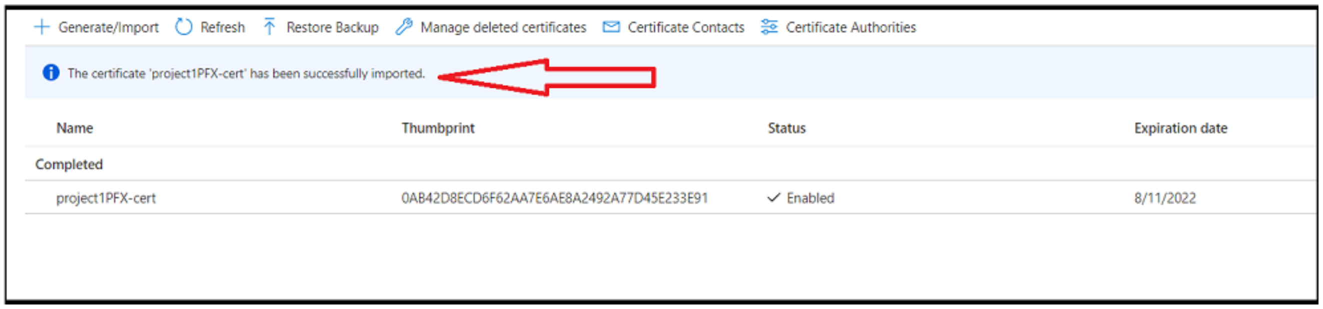
Certificate Name *: project1PFX-cert

Upload Certificate File *: Select a file

Password:

4. Select "Create" to upload your certificate.

   - The following success message should appear to confirm that your PFX certificate has been uploaded to your key vault:

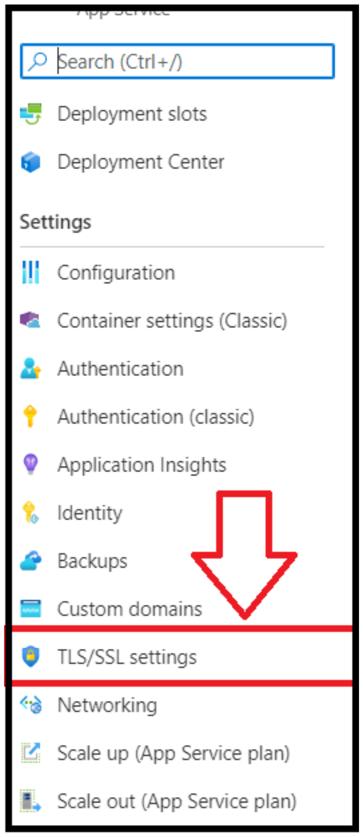


The certificate 'project1PFX-cert' has been successfully imported.

Name	Thumbprint	Status	Expiration date
Completed			
project1PFX-cert	0AB42D8ECD6F62AA7E6AE8A2492A77D45E233E91	✓ Enabled	8/11/2022

5. Now that you have uploaded your certificate, it's time to add it to your web application. To do so, complete the following steps:

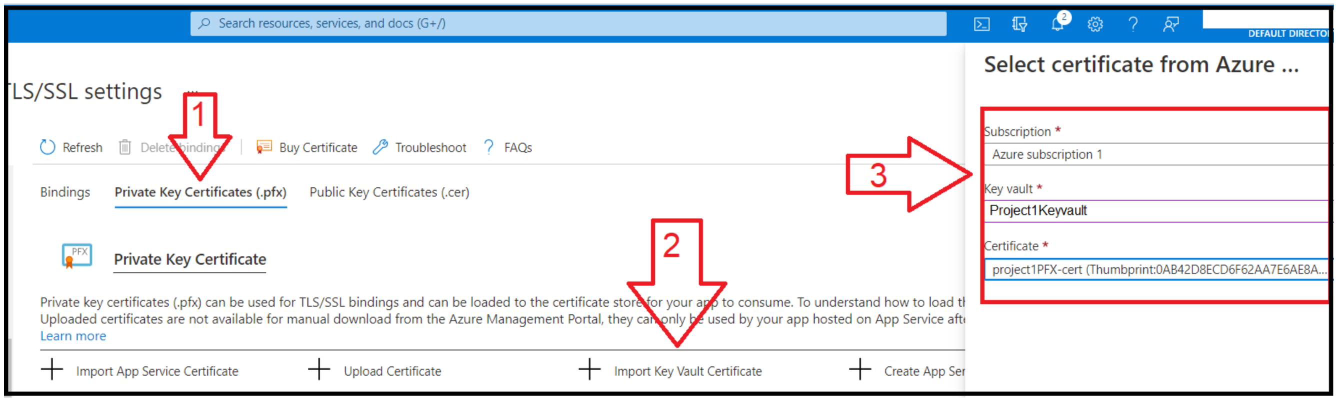
   - Return to the web application (under "App Services") that you created on Day 1.
   - On this page, select "TLS/SSL Settings," as the following image shows:



6. On this page, import your new PFX certificate from your key vault. To do so, complete the following steps:

    - (1) Select "Private Key Certificates."
    - (2) Click "+ Import Key Vault Certificate."
    - (3) When the pane opens on the right side of your screen, enter your subscription, key vault, and certificate that you just created.

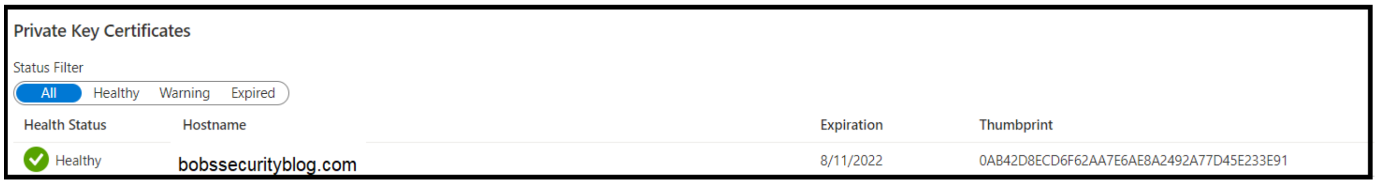
    The following image shows these steps:



Private key certificates (.pfx) can be used for TLS/SSL bindings and can be loaded to the certificate store for your app to consume. To understand how to load t
Uploaded certificates are not available for manual download from the Azure Management Portal, they can only be used by your app hosted on App Service aft

Subscription: Azure subscription 1
Key vault: Project1Keyvault
Certificate: project1PFX-cert (Thumbprint:0AB42D8ECD6F62AA7E6AE8A...

7. Your certificate should now appear as an available certificate to use for your web application, as the following image shows:



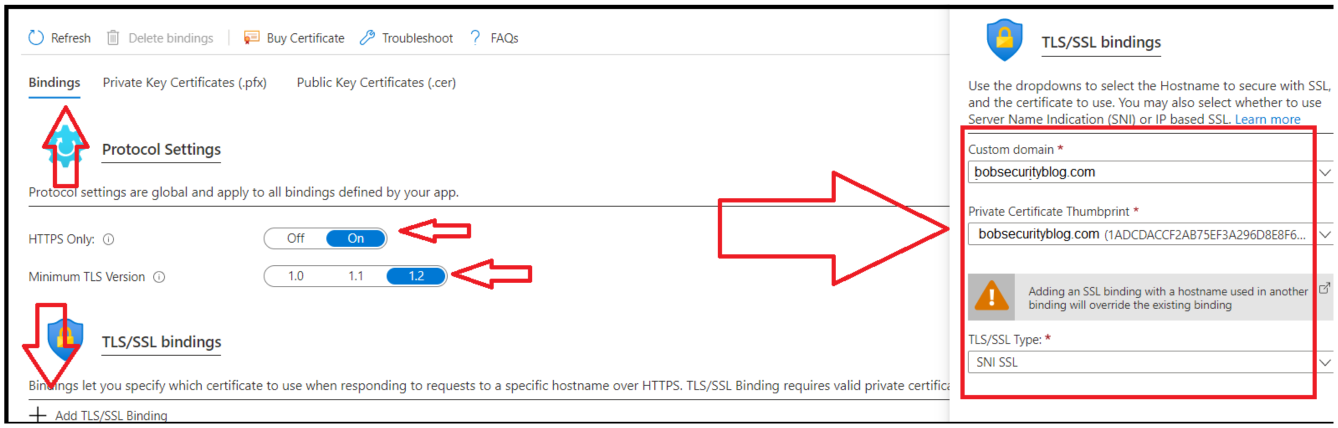
Health Status	Hostname	Expiration	Thumbprint
✓ Healthy	bobssecurityblog.com	8/11/2022	0AB42D8ECD6F62AA7E6AE8A2492A77D45E233E91

8. Currently your certificate is just available to use for your web application—now, it is time to bind it! To bind your certificate, complete the following steps:

    - Select "Bindings."

- Toggle "HTTPS Only" to "On."
- Toggle "Minimum TLS Version" to "1.2."
- Click "+ Add TLS/SSL Binding."
- When the pane appears on the right side of your screen, select your domain, the matching certificate, and "SNI/SSL" for the TLS/SSL type.

  The following image shows these steps:



Protocol settings are global and apply to all bindings defined by your app.

HTTPS Only: Off On

Minimum TLS Version 1.0 1.1 1.2

Bindings let you specify which certificate to use when responding to requests to a specific hostname over HTTPS. TLS/SSL Binding requires valid private certifica

TLS/SSL bindings

Use the dropdowns to select the Hostname to secure with SSL, and the certificate to use. You may also select whether to use Server Name Indication (SNI) or IP based SSL. Learn more

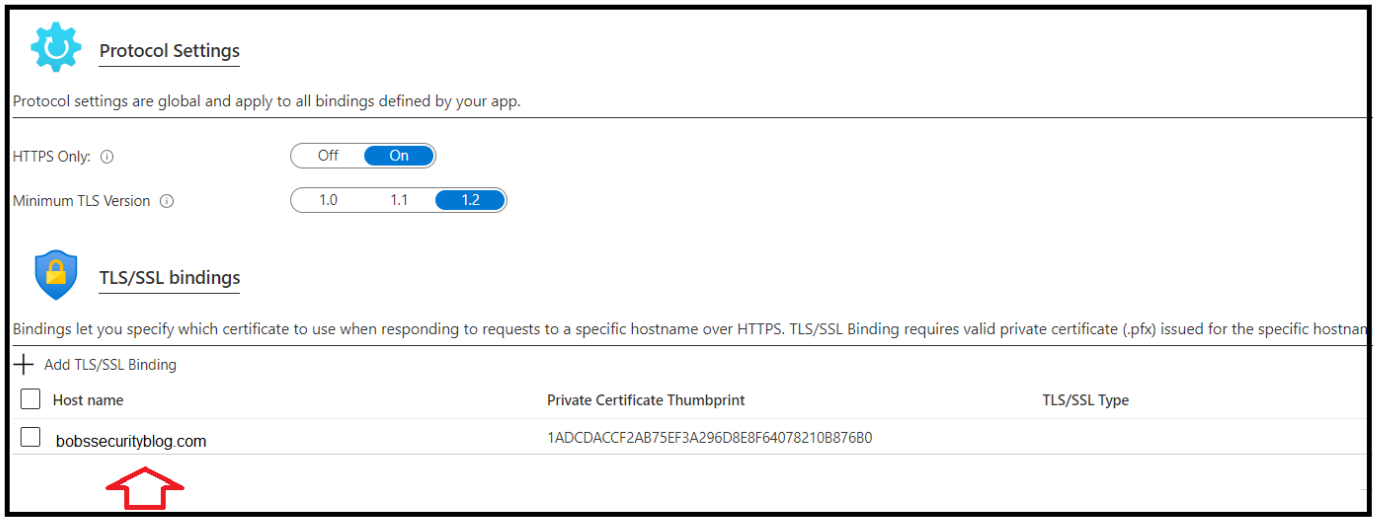
Custom domain *: bobsecurityblog.com

Private Certificate Thumbprint *: bobsecurityblog.com (1ADCDACCF2AB75EF3A296D8E8F6...

Adding an SSL binding with a hostname used in another binding will override the existing binding

TLS/SSL Type: *: SNI SSL

9. After selecting "Add Binding," your binding should appear in the "TLS/SSL bindings" list, as the following image shows:



Protocol settings are global and apply to all bindings defined by your app.

HTTPS Only: Off On

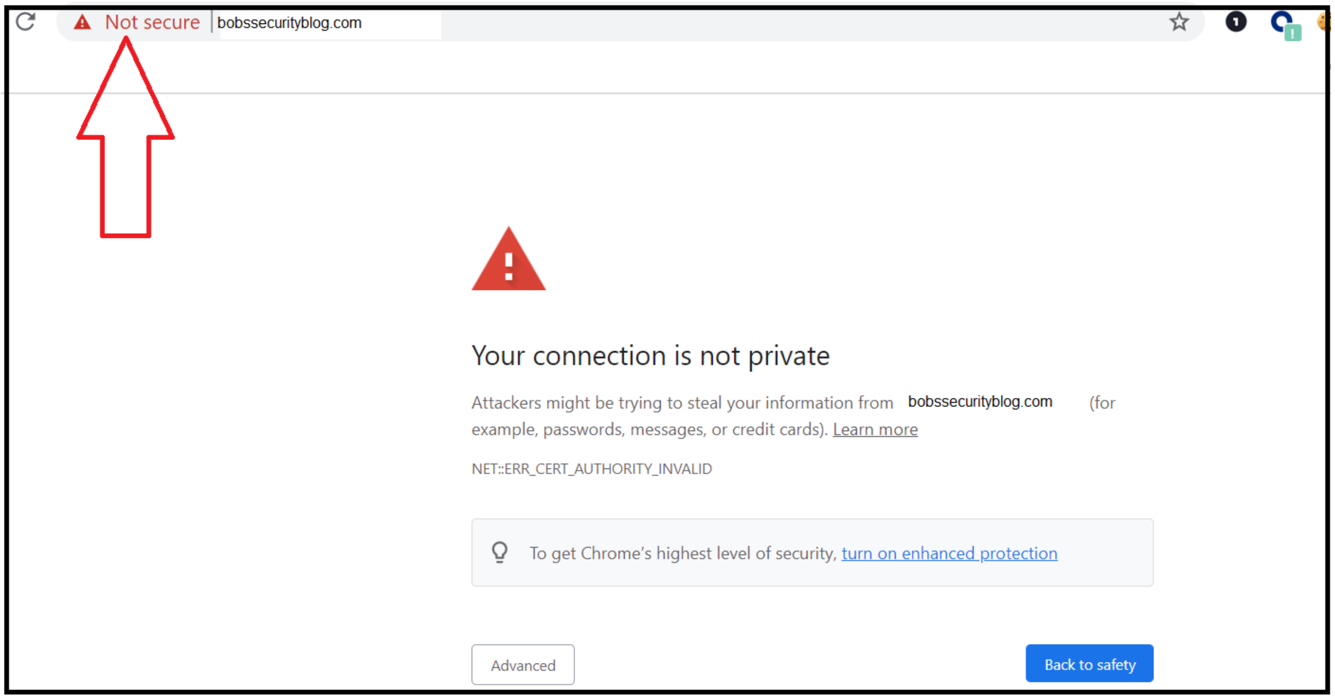
Minimum TLS Version 1.0 1.1 1.2

Bindings let you specify which certificate to use when responding to requests to a specific hostname over HTTPS. TLS/SSL Binding requires valid private certificate (.pfx) issued for the specific hostnan

Host name	Private Certificate Thumbprint	TLS/SSL Type
bobssecurityblog.com	1ADCDACCF2AB75EF3A296D8E8F64078210B876B0	

10. Now, open a browser and access your web application.

    - Did your browser return an error like the one shown in the following image?



Your connection is not private

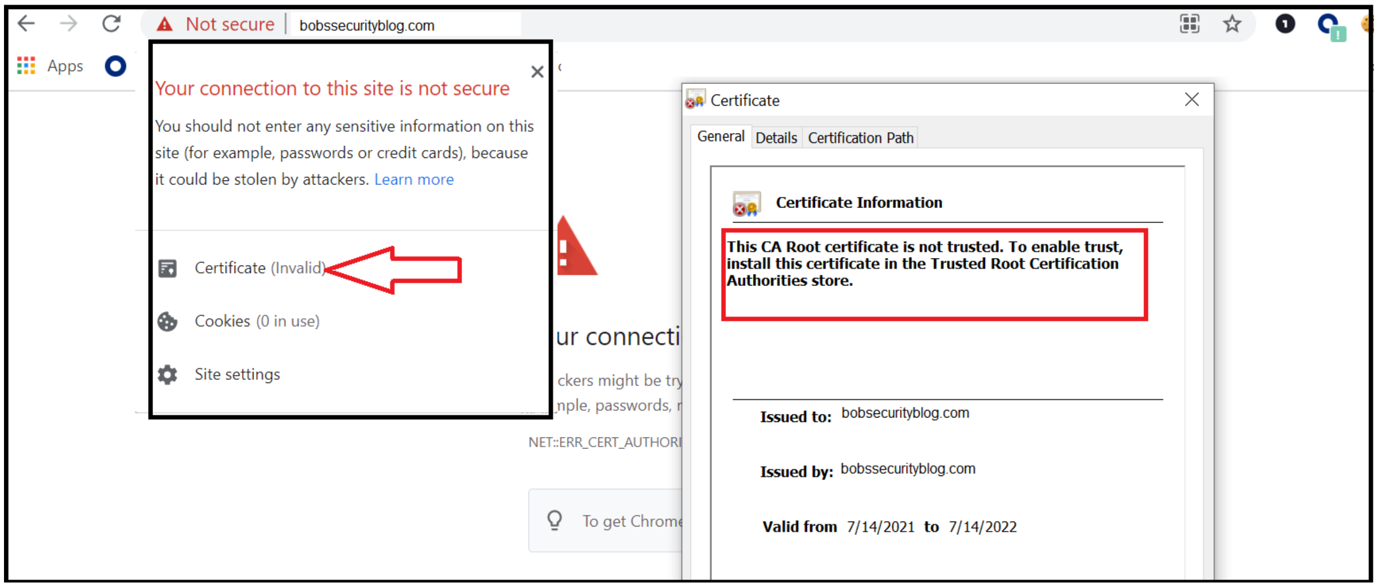
Attackers might be trying to steal your information from bobssecurityblog.com (for example, passwords, messages, or credit cards). Learn more

NET::ERR_CERT_AUTHORITY_INVALID

To get Chrome's highest level of security, turn on enhanced protection

- Note that this image is from the Chrome browser; the message may look slightly different depending on your browser.

11. Let's examine the certificate that you just added. Click "Not secure" in the search bar if you are in Chrome, or a similar message depending on your browser, as shown in the following image:



Your connection to this site is not secure

You should not enter any sensitive information on this site (for example, passwords or credit cards), because it could be stolen by attackers. Learn more

Certificate Information

This CA Root certificate is not trusted. To enable trust, install this certificate in the Trusted Root Certification Authorities store.

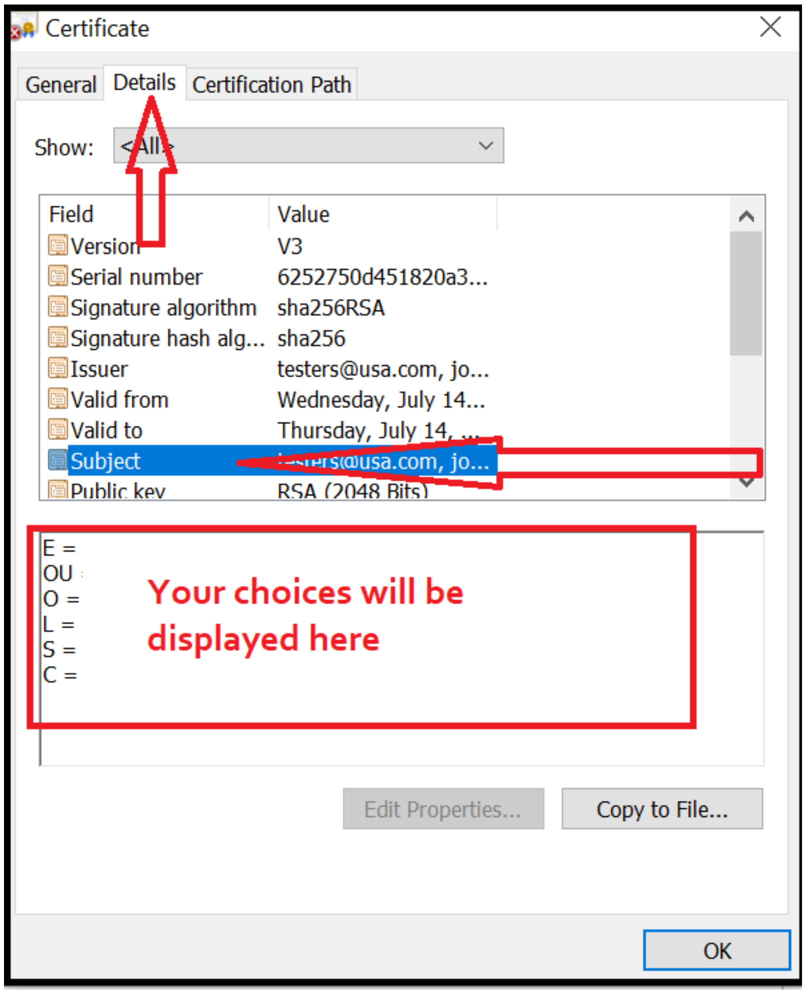
Issued to: bobsecurityblog.com

Issued by: bobsecurityblog.com

Valid from 7/14/2021 to 7/14/2022

- After selecting "Not secure," select "Certificate (Invalid)" from the menu to examine your certificate.
- Note the reason for your error based on the message on your certificate. This message is due to the fact that your certificate was created by you and not a trusted CA.

12. Next, click the "Details" tab of your certificate, then select the "Subject" option, as the following image shows:



Field	Value
Version	V3
Serial number	6252750d451820a3...
Signature algorithm	sha256RSA
Signature hash alg...	sha256
Issuer	testers@usa.com, jo...
Valid from	Wednesday, July 14...
Valid to	Thursday, July 14, ...
Subject	testers@usa.com, jo...
Public key	RSA (2048 Bits)

E =
OU
O =
L =
S =
C =

- Note the results that now display in the box on the bottom; these were the options that you selected when you created your certificate with OpenSSL.

You have successfully created a self-signed certificate and bound it to your web application using Azure!

:warning: Checkpoint :warning:
Before continuing, make sure that you have completed the following critical tasks:
:heavy*check*mark: Created your Azure key vault.
:heavy*check*mark: Created a self-signed certificate using Open SSL.
:heavy*check*mark: Imported and bound your self-signed certificate to your web app.

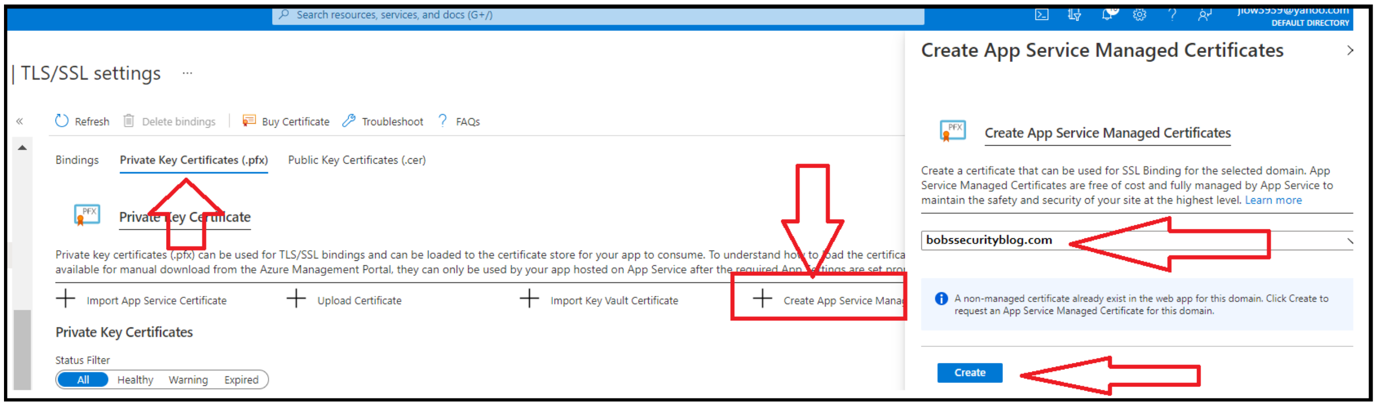
Part 4: Create and Bind an App Service Managed Certificate

In this part, you will use Azure's managed certificate to create and bind a more secure certificate to your web application.

You were just able to create and bind your own self-signed certificate to your web application to encrypt your web traffic. However, unfortunately, your browser displayed warnings to visitors that your website is not trusted and that there may be security risks associated with your web application. (Note that you will explore this issue further in the daily review questions.)

You will now create and bind a more secure, trusted SSL certificate to your web app using Azure's cloud services. To do so, complete the following steps:

1. First, return to "TLS/SSL settings" under your web application.
2. Select "Private Key Certificates."
3. Select "+ Create App Service Managed Certificates."
4. When the pop-up appears on the right side of your screen, select your domain and click "Create," as the following image shows:



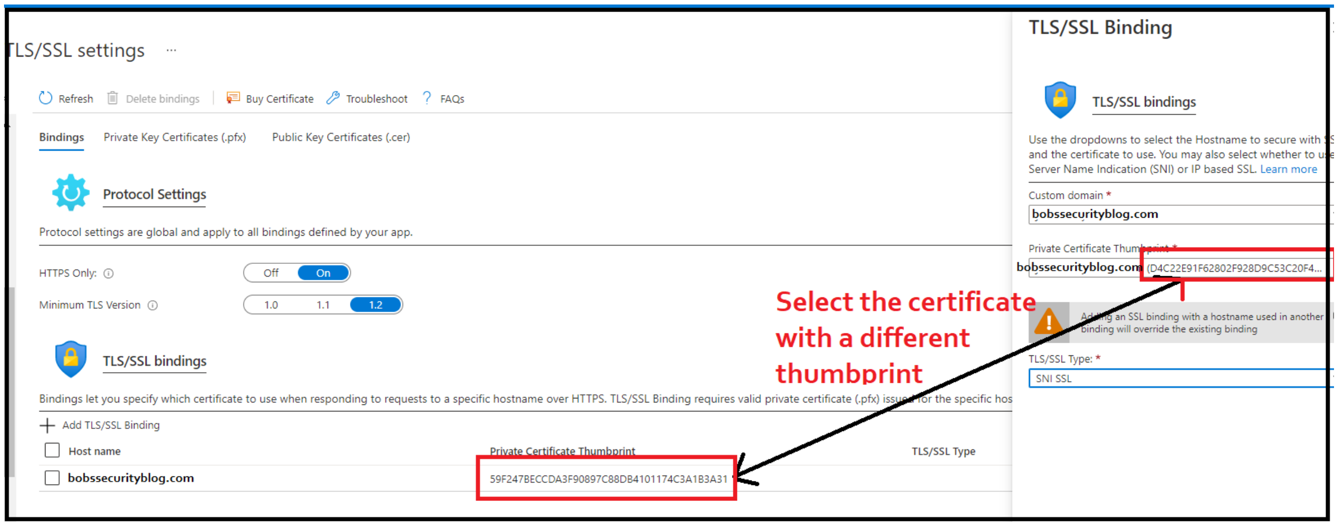
Private key certificates (.pfx) can be used for TLS/SSL bindings and can be loaded to the certificate store for your app to consume. To understand how to load the certifica available for manual download from the Azure Management Portal, they can only be used by your app hosted on App Service after the required App settings are set prov

Create a certificate that can be used for SSL Binding for the selected domain. App Service Managed Certificates are free of cost and fully managed by App Service to maintain the safety and security of your site at the highest level. Learn more

bobssecurityblog.com

A non-managed certificate already exist in the web app for this domain. Click Create to request an App Service Managed Certificate for this domain.

5. Once your app service managed certificate has been created, return to the "Bindings" tab.
6. Select "+ Add TLS/SSL Binding."
7. When the pane appears on the right side of your screen, select your domain and the new certificate that you just created.
    - Note the current thumbprint next to your domain under the "TLS/SSL bindings" menu, and select the certificate with the other thumbprint, as the following image shows:



Protocol settings are global and apply to all bindings defined by your app.

HTTPS Only: Off [✓ On]

Minimum TLS Version 1.0 1.1 [✓ 1.2]

Bindings let you specify which certificate to use when responding to requests to a specific hostname over HTTPS. TLS/SSL Binding requires valid private certificate (.pfx) issued for the specific hos

bobssecurityblog.com 59F247BECCDA3F90897C88DB4101174C3A1B3A31

Use the dropdowns to select the Hostname to secure with S and the certificate to use. You may also select whether to u Server Name Indication (SNI) or IP based SSL. Learn more

Custom domain * bobssecurityblog.com

Private Certificate Thumbprint * bobssecurityblog.com (D4C22E91F62802F928D9C53C20F4...

Adding an SSL binding with a hostname used in another binding will override the existing binding

TLS/SSL Type: * SNI SSL

8. Select "SNI/SSL" for the TLS/SSL type, then click "Add Binding."
9. Now that your new app services managed certificate has been bound to your web application, revisit your website. You should not see any warnings displayed this time!

Congratulations, you have now created a web application and secured it with a trusted SSL certificate!

Part 5: Answer Review Questions

- Open your copy of the review questions, make a copy of the document, and answer the Day 2 review questions.
    - Note that you will submit this document as one of your deliverables at the end of the project.

Day 2 Milestone

In today's class, you:

- (1) **Created a key vault**.
- (2) **Created a self-signed certificate**.
- (3) **Imported and bound your self-signed certificate to your web app**.
- (4) **Created and bound an app service managed certificate**.
- (5) **Answered review questions**.

Completing these steps required you to leverage your terminal, systems administration, cloud, cryptography, and networking skills. This is an impressive set of tools to have in your toolkit!