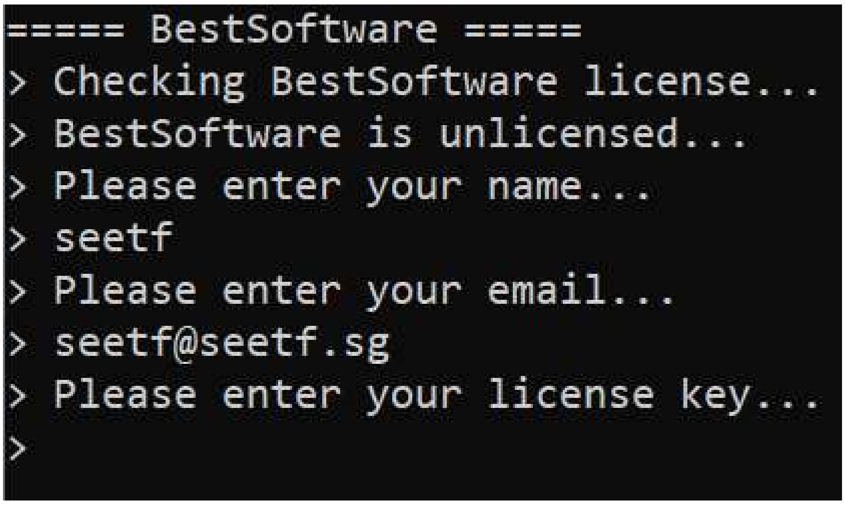
BestSoftware

Challenge Description:
Help! I purchased the license for a software called BestSoftware. However, I forgot the license key to it, could you help me?
My name is "seetf" and my email is "seetf@seetf.sg (mailto:seetf@seetf.sg)". Thank you!

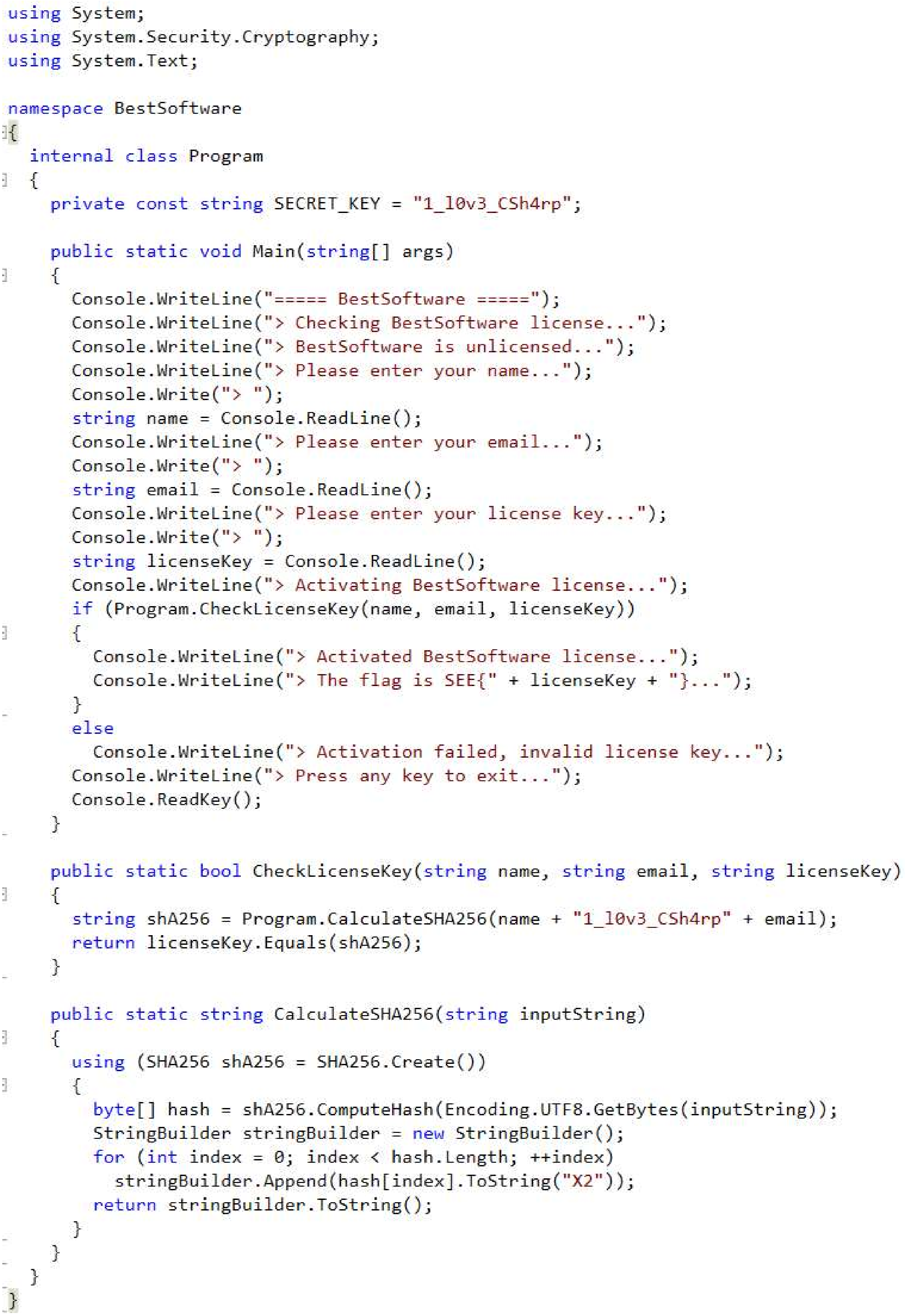
Challenge Author: Gelos
Genre: Reversing
Provided: "BestSoftware" executable program

We want to retrieve the key from the program provided, likely by decompiling it. First, we observe the program. Upon running the application, the user's name and email are required. Provide the information as given above. The license key is now required.



```
===== BestSoftware =====
> Checking BestSoftware license...
> BestSoftware is unlicensed...
> Please enter your name...
> seetf
> Please enter your email...
> seetf@seetf.sg
> Please enter your license key...
>
```

Since no other clues are provided, we proceed with decompiling the application. For this challenge, JetBrain's dotPeek has been recommended for beginners (https://www.jetbrains.com/decompiler/ (https://www.jetbrains.com/decompiler/)). Open the application using dotPeek. We can see a short program written for the application.



```
using System;
using System.Security.Cryptography;
using System.Text;

namespace BestSoftware
{
  internal class Program
  {
    private const string SECRET_KEY = "1_l0v3_CSh4rp";

    public static void Main(string[] args)
    {
      Console.WriteLine("===== BestSoftware =====");
      Console.WriteLine("> Checking BestSoftware license...");
      Console.WriteLine("> BestSoftware is unlicensed...");
      Console.WriteLine("> Please enter your name...");
      Console.Write("> ");
      string name = Console.ReadLine();
      Console.WriteLine("> Please enter your email...");
      Console.Write("> ");
      string email = Console.ReadLine();
      Console.WriteLine("> Please enter your license key...");
      Console.Write("> ");
      string licenseKey = Console.ReadLine();
      Console.WriteLine("> Activating BestSoftware license...");
      if (Program.CheckLicenseKey(name, email, licenseKey))
      {
        Console.WriteLine("> Activated BestSoftware license...");
        Console.WriteLine("> The flag is SEE{" + licenseKey + "}...");
      }
      else
        Console.WriteLine("> Activation failed, invalid license key...");
      Console.WriteLine("> Press any key to exit...");
      Console.ReadKey();
    }

    public static bool CheckLicenseKey(string name, string email, string licenseKey)
    {
      string shA256 = Program.CalculateSHA256(name + "1_l0v3_CSh4rp" + email);
      return licenseKey.Equals(shA256);
    }

    public static string CalculateSHA256(string inputString)
    {
      using (SHA256 shA256 = SHA256.Create())
      {
        byte[] hash = shA256.ComputeHash(Encoding.UTF8.GetBytes(inputString));
        StringBuilder stringBuilder = new StringBuilder();
        for (int index = 0; index < hash.Length; ++index)
          stringBuilder.Append(hash[index].ToString("X2"));
        return stringBuilder.ToString();
      }
    }
  }
}
```

The flag is printed from the following line:

```
Console.WriteLine("> The flag is SEE{" + licenseKey + "}...");
```

We need to identify the license key.

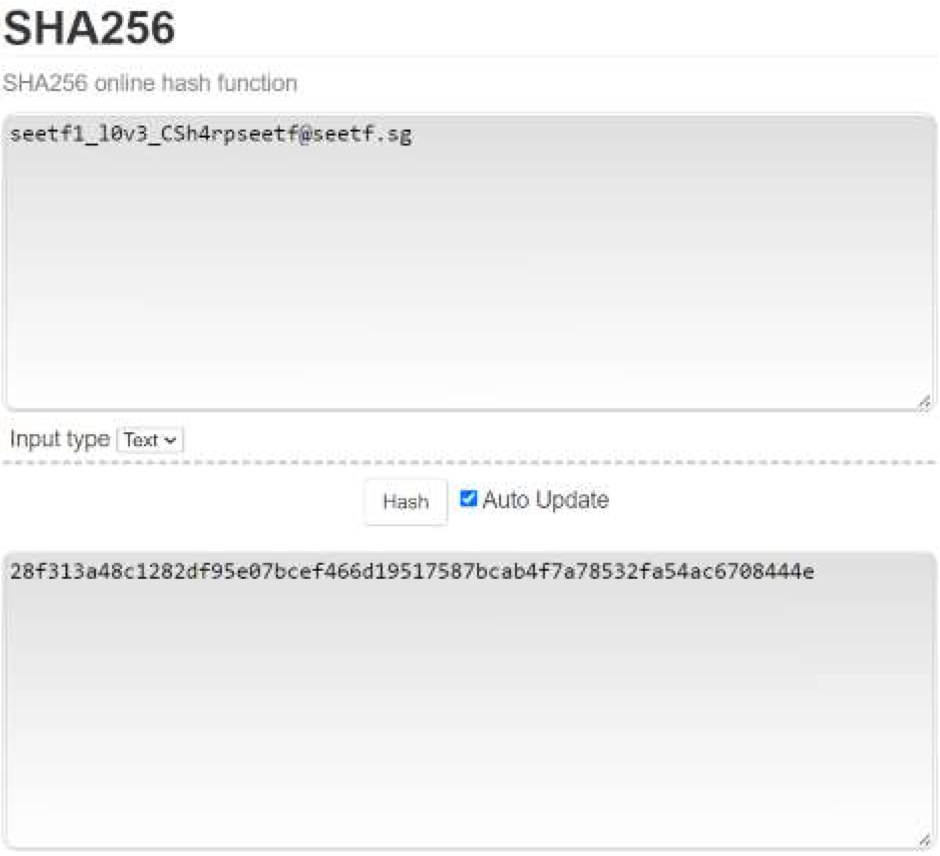
From the code, we can identify 2 key functions: (1) `CheckLicenseKey()` and (2) `CalculateSHA256` .

(1) The license key is calculated based on the user's name, email, and the keyword `"1_l0v3_CSh4rp"` using function (2). This license key is then checked against the user's input to verify that the correct key has been used.

(2) The license key is calculated here. The string is encoded using the SHA256 standard. A simple Google search will show that SHA256 is a cryptographic hash function.

A cryptographic hash function is a one-way function that is practically irreversible. In our challenge, we can easily identify the key as `name + "1_l0v3_CSh4rp" + email` , or `"seetf1_l0v4_CSh4rpseetf@seetf.sg"` . We now require the output of the SHA256 hashing.

For the encryption, there exists SHA256 encryptors online. I used the encryptor provided in this website: [https://emn178.github.io/online-tools/sha256.html (https://emn178.github.io/online-tools/sha256.html)](https://emn178.github.io/online-tools/sha256.html).

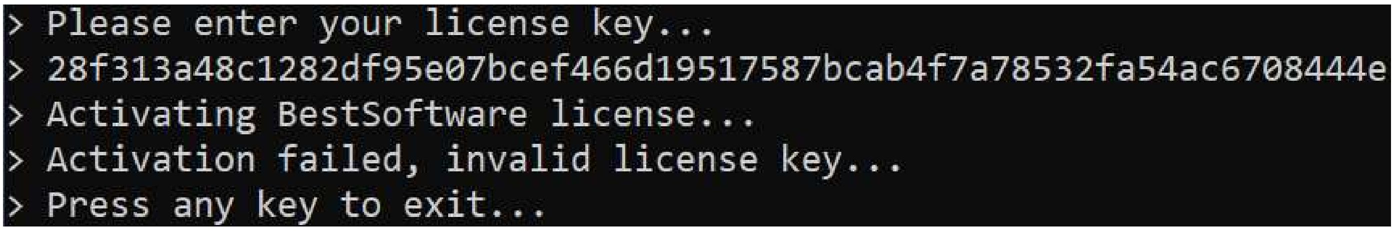


```
seetf1_l0v3_CSh4rpseetf@seetf.sg
```

Input type Text

```
28f313a48c1282df95e07bcef466d19517587bcab4f7a78532fa54ac6708444e
```

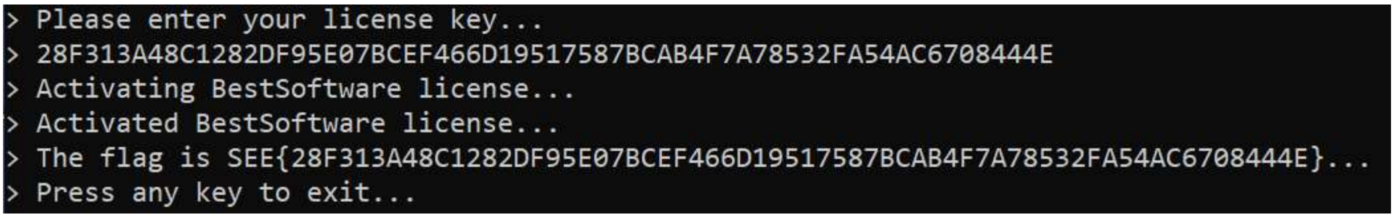
Run the application again and try inputting the following license key as obtained: `28f313a48c1282df95e07bcef466d19517587bcab4f7a78532fa54ac6708444e`



```
> Please enter your license key...
> 28f313a48c1282df95e07bcef466d19517587bcab4f7a78532fa54ac6708444e
> Activating BestSoftware license...
> Activation failed, invalid license key...
> Press any key to exit...
```

Activation still failed? But the decompiler clearly stated SHA256 encryption was used. Well, it happens that the encryption is not case sensitive. However, the challenge application might be.

Use Python's string method, `str.upper()` , to return the key in uppercase. Try again.



```
> Please enter your license key...
> 28F313A48C1282DF95E07BCEF466D19517587BCAB4F7A78532FA54AC6708444E
> Activating BestSoftware license...
> Activated BestSoftware license...
> The flag is SEE{28F313A48C1282DF95E07BCEF466D19517587BCAB4F7A78532FA54AC6708444E}...
> Press any key to exit...
```

As determined earlier, the flag is simply the license key formatted within the flag format `SEE{}` . Flag has been obtained. :)