This article will go over using a **Python** script to check **SSL/TLS certifications** expiration date. It will then send a notification email via **AWS Simple Email Service**. This is helpful in production environments to ensure that your certificates do not expire and cause issues with your websites security.

Prerequisites

AWS account
VSCode or preferred IDE
AWS CLI installed and configured

Step 1

First we will add an email account to Amazon SES. In the AWS console navigate to *SES*. Click on *Verified identities* > *Create identity*. Here you will add your preferred email address for sending the email notifications. Click on *Create identity* when finished.

Once that is complete you will receive an email to verify. Now you are able to send emails through SES.

Step 2

Next step is to get the python script set up. In VSCode make sure you have the AWS CLI installed and configured with your credentials. Then you will want to download the boto3 extension for python. Use `pip install boto3`.

Now for the code piece you can use the script I created below.

```
import ssl
import socket
import datetime
import boto3

client = boto3.client("ses", region_name="us-east-1")

print(f"Program to check SSL certificate validity and 
expiration date\n")

##opening file
with open("server_ip.txt") as ip_file:

    ##check  certificate expiration
    for ip in ip_file:

        try:
            host, port = ip.strip().split(":")
            print(f"\nChecking certifcate for server 
{host}")
            context = ssl.create_default_context()
            with socket.create_connection((host, port)) 
as sock:
                with context.wrap_socket(sock, 
server_hostname=host) as ssock:
                    certificate = ssock.getpeercert()
                certExpires = 
datetime.datetime.strptime(
                    certificate["notAfter"], "%b %d %H:
%M:%S %Y %Z"
                )
                daysToExpiration = (certExpires - 
datetime.datetime.now()).days
                print(f"Expires on: {certExpires} in 
{daysToExpiration} days")
                ##preparing mailbody
                mailbody = (
                    "Server name: "
                    + host
                    + ", expires in "
                    + str(daysToExpiration)
                    + " days."
                )

        except:
            print(f"error on connection to Server, {host}")

        ##sending ses email
        if daysToExpiration < 45:
            response = client.send_email(
                Destination={
                    "ToAddresses": ["user@gmail.com"],
                },
                Message={
                    "Body": {
                        "Text": {
                            "Charset": "UTF-8",
                            "Data": "The following requires attention; "
                            + mailbody
                            + "\nThank you.",
                        },
                    },
                    "Subject": {
                        "Charset": "UTF-8",
                        "Data": "Certificate Expiring Soon",
                    },
                },
                Source="user@gmail.com",
            )

print(f"\nCert check complete!")
```

This script will do all the work for you! It takes the websites you would like to be verified and checks when the certificate for that website expires. Then it will send an email through SES based on how many days from expiration you would like to be notified to renew. You will need to input your AWS region and the source email as well as the email you would like to receive the notifications.

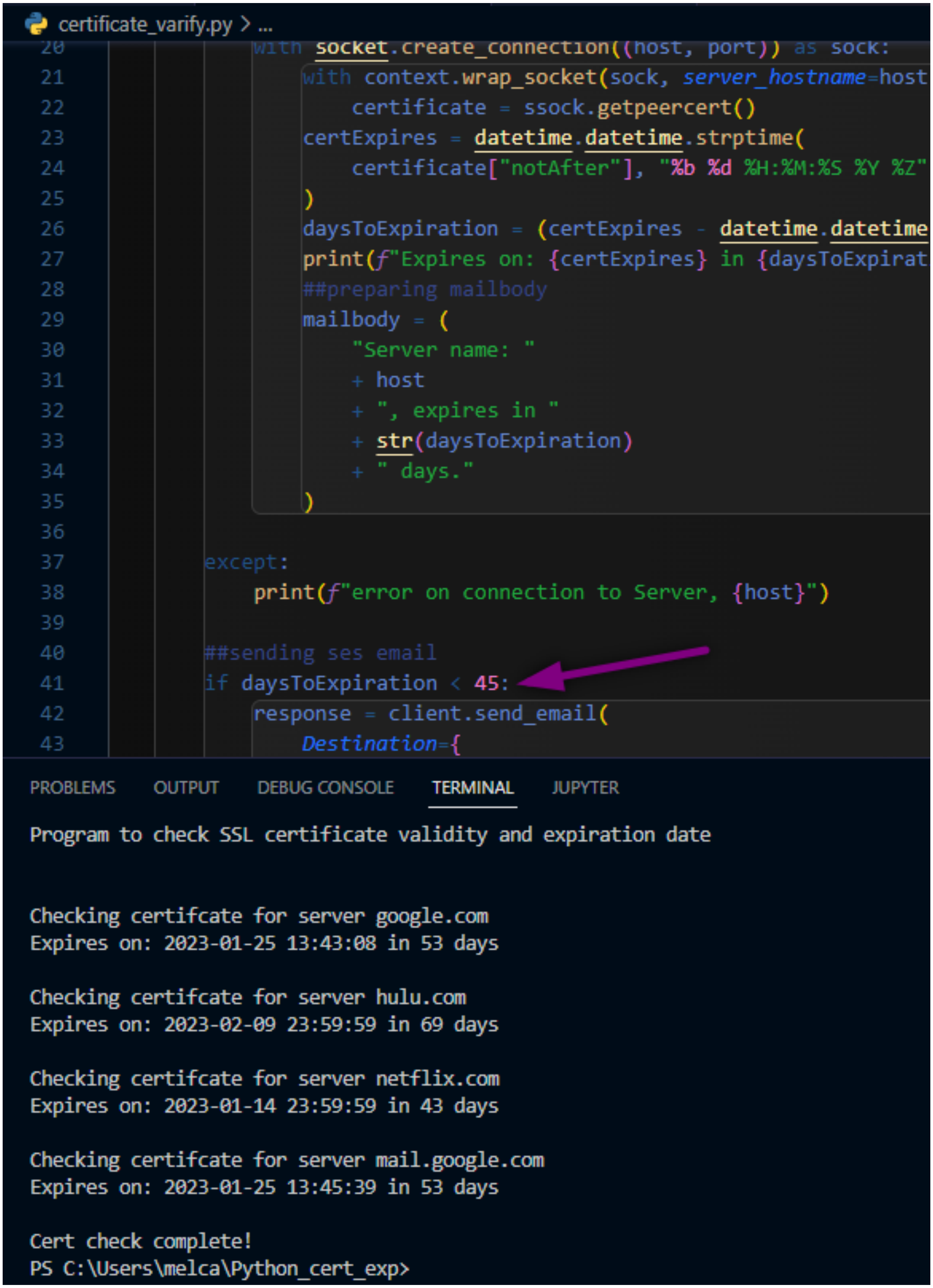
How I have the code set up is to open a text file with the websites I want checked as well as the port they are on. I will show an example below.

```
google.com:443
hulu.com:443
netflix.com:443
mail.google.com:443
```

You can also fork the code from my repo here.

Step 3

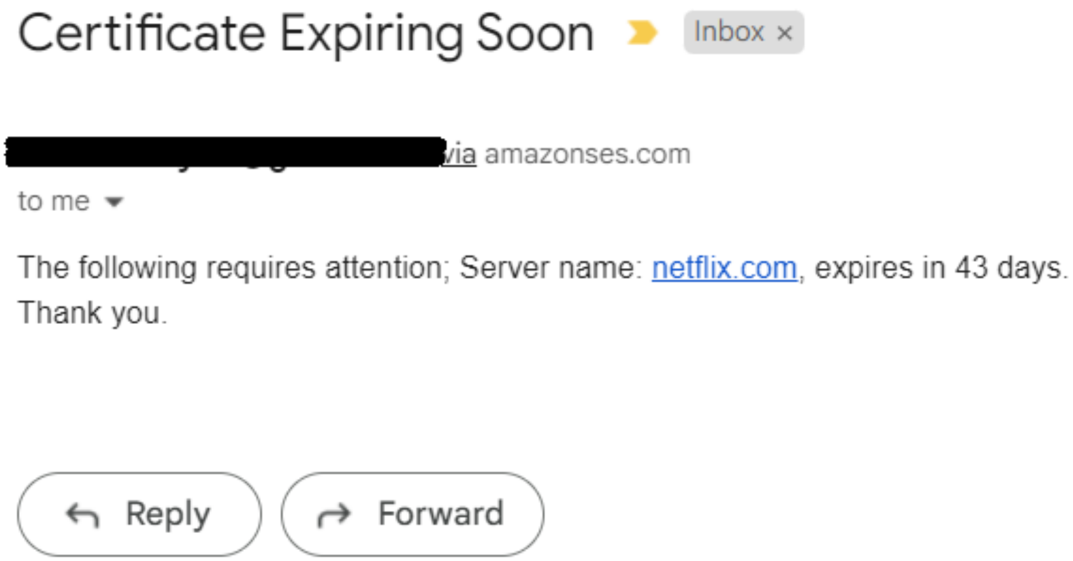
Let’s see how the code runs! In VSCode click on the play button and watch the magic. Be sure to command into the correct folder where your python code is. The output should look as follows.



```python
            with socket.create_connection((host, port)) as sock:
                with context.wrap_socket(sock, server_hostname=host
                    certificate = ssock.getpeercert()
                certExpires = datetime.datetime.strptime(
                    certificate["notAfter"], "%b %d %H:%M:%S %Y %Z"
                )
                daysToExpiration = (certExpires - datetime.datetime
                print(f"Expires on: {certExpires} in {daysToExpirat
                ##preparing mailbody
                mailbody = (
                    "Server name: "
                    + host
                    + ", expires in "
                    + str(daysToExpiration)
                    + " days."
                )

        except:
            print(f"error on connection to Server, {host}")

        ##sending ses email
        if daysToExpiration < 45:
            response = client.send_email(
                Destination={
```

```
Program to check SSL certificate validity and expiration date


Checking certifcate for server google.com
Expires on: 2023-01-25 13:43:08 in 53 days

Checking certifcate for server hulu.com
Expires on: 2023-02-09 23:59:59 in 69 days

Checking certifcate for server netflix.com
Expires on: 2023-01-14 23:59:59 in 43 days

Checking certifcate for server mail.google.com
Expires on: 2023-01-25 13:45:39 in 53 days

Cert check complete!
PS C:\Users\melca\Python_cert_exp>
```

It prints out the server being checked, the date it expires, and in how many days it will expire. Note the < *45*, this is the number of days until expiration that I want to be notified about. So only certificates that expire in less than 45 days will I receive an email notification about. You can change this number to whatever works best for your needs. I will verify that this worked by checking if I received an email regarding *netflix.com* server certificate.



Certificate Expiring Soon

[redacted] via amazonses.com

to me

The following requires attention; Server name: netflix.com, expires in 43 days.
Thank you.

It worked! In conclusion we used a python script to check servers TLS/SSL certificates expiration date and sent a notification through SES. To further this project you could add a cron job to check daily automatically. I hope you found this helpful!