

## Question 1

### Analysis of Screenshots

#### Get Request (with Headers) [rest of screenshots attached]

The first screenshot shows the 'Query' tab of a REST client. The URL is `https://api.openai.com/v1/models?model=gpt-3.5-turbo`. The 'Query Parameters' section has a checked 'model' parameter with the value 'gpt-3.5-turbo'.

The second screenshot shows the 'Headers' tab. The 'HTTP Headers' section has two checked headers: 'Accept' with value '\*'/\*' and 'User-Agent' with value 'Thunder Client (https://www.thunderclient.com)'. There is also an unchecked 'header' field with a 'value' placeholder.

The third screenshot shows the 'Auth' tab. The 'Auth' dropdown is set to 'Bearer'. The 'Bearer Token' field contains the API key: `sk-XxT6fxvDEA1SKHrB4tFT3BlbkFjpVfLXhArfxraLK1aKR`.

This get request has headers specifying the client from which the request is made, and query which includes the parameters about which model data is to be received. Here we are requesting data about GPT-3.5-turbo models. The authorization includes the API key.

#### Get Response

The screenshot shows the 'Response' tab of the REST client. The status is '200 OK', the size is '42.01 KB', and the time is '1.27 s'. The response body is a JSON object:

```
1 {
2   "object": "list",
3   "data": [
4     {
5       "id": "babbage",
6       "object": "model",
7       "created": 1649358449,
8       "owned_by": "openai",
9       "permission": [
10        {
11          "id": "modelperm-49FUp5v084tBB49tC4z8LPH5",
12          "object": "model_permission",
13          "created": 1669085501,
14          "allow_create_engine": false,
```

The status is 200 OK which refers to a success message along with the size of the response message and the time to get this response.

Retrieves a model instance in the query, here we requested instance of GPT-3.5 turbo, so information about the model such as the owner and permissioning is returned in the answer.

## Post Request

The screenshot shows a Thunder Client interface for a POST request to `https://api.openai.com/v1/chat/completions`. The 'Headers' tab is active, showing the following headers:

| Header       | Value  |
|--------------|--|
| Accept       | */*  |
| User-Agent   | Thunder Client (https://www.thunderclient.com) |
| Content-Type | application/json                               |
| header       | value  |

The 'Body' tab is also visible, showing a JSON payload:

```

1 {
2   "model": "gpt-3.5-turbo",
3   "messages": [{"role": "user", "content": "Hello!"}]
4 }
  
```

At the bottom, the status is `200 OK`, size is `323 Bytes`, and time is `1.41 s`.

Creates a model response for the given chat conversation. Here we are posting for GPT-3.5-turbo model, and the body includes the message we are trying to post.

## Post Response

The screenshot shows the JSON response received from the OpenAI API. The status is `200 OK`, size is `323 Bytes`, and time is `1.41 s`. The response is as follows:

```

1 {
2   "id": "chatcmpl-77VE50FuUwLIDThv1NuzHYtg71yYR",
3   "object": "chat.completion",
4   "created": 1682022685,
5   "model": "gpt-3.5-turbo-0301",
6   "usage": {
7     "prompt_tokens": 10,
8     "completion_tokens": 10,
9     "total_tokens": 20
10  },
11  "choices": [
12    {
13      "message": {
14        "role": "assistant",
15        "content": "Hi there! How can I assist you today?"
16      },
17      "finish_reason": "stop"
18    }
19  ]
20 }
  
```

The status 200 OK refers to a successful message and the size is of the entire message and time in which the response was received.

The response includes basic information of the model and also includes the choices array in which the conversation so far is recorded, so here it is "Hey there, how can I assist you today".

## Benefits and Purposes of OpenAI API

OpenAI has developed several innovative AI models, including GPT-3, DALL-E, and CLIP, which have been used in many applications, from chatbots and virtual assistants to image recognition and content creation. OpenAI has also partnered with several leading organizations, including Microsoft, to advance the development and deployment of AI technologies.

One way to integrate the functionalities provided by OpenAI in any apps we are trying to develop is through the use of APIs provided. The main reasons why these APIs are used today are described below.

### 1. Integration of Advanced AI Features

The use of models in applications today can enhance user experience through the use of advanced AI functionality. Currently, OpenAI's algorithms are used by some of the world's largest companies, including Google, Facebook, and Amazon.

For instance, DALL-E, an OpenAI creation, can be trained to generate images from a text description input, creating fascinating AI Art. Another OpenAI creation, CLIP can predict the most relevant text description for that image, which can be used for image recognition, visual search, and personalized recommendations. The third and most famous AI creation is GPT-3. GPT-3 can be used in app development to automate tasks such as customer support, develop personalized recommendations, and create content for the app. All three models support user personalization which significantly increases user retention.

### 2. Improved Customer Experience

As explained above, personalizing content for users is an extremely useful feature provided by OpenAI APIs. OpenAI can be used to create chatbots and virtual assistants that can assist customers and answer their queries in real-time. This can result in better customer experiences and faster response times.

### 3. Efficiency

Using OpenAI APIs in apps today can automate redundant tasks such as creating visual content again and again, or by answering customer support queries repetitively. Both these tasks can be accomplished using AI models, GPT-3 for customer support queries and text-to-visual content generation through DALL-E.

### 4. Include Predictive Analytics Features for Better App Monetization

Predictive analytics is an interesting feature of OpenAI that enables you to use machine learning techniques for analyzing data and making predictions. Predictive analytics is implemented to improve the performance of AI-based systems, such as chatbots.

Using various algorithms and models, such as linear regression, decision trees, and neural networks, you can perform predictive analysis and know forecasting sales, identify potential customer churn, or detect fraud. The goal of using OpenAI is also to understand your customers so that you focus on personalized marketing and advertising to reduce churn rate.

For example,

**In-app Purchase Prediction:** By analyzing user behavior, a model could predict which users are most likely to make an in-app purchase, and target them with personalized promotions or offers.

**Ad Targeting:** A model could predict which users are most likely to click on ads, and target them with ads that are more likely to be relevant to their interests

## 5. Ability to Program in Natural Language

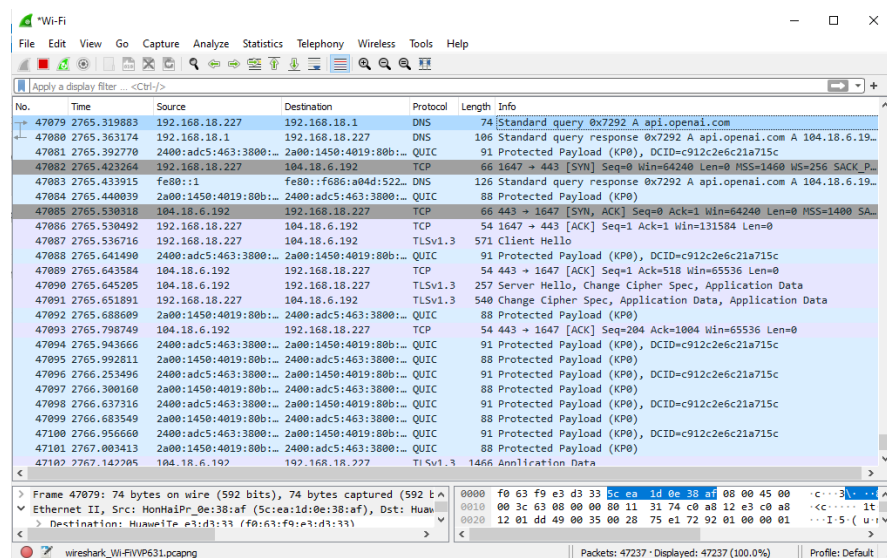
Apart from benefits for end users of intelligent applications built using AI, there are several benefits for the developers as well. The data resources available to developers can be accessed in plain English. Previously, programmers used to analyze databases and other datasets in their native language what they want to know about the dataset. But now they don't need to know SQL or the Python Pandas Library to access content.

Moreover, in order to extract a piece of information from a dataset, GPT-3 is extremely useful. There is no longer a need to analyze large datasets using Advanced SQL or Python. For Big Data projects, this is going to be vital.

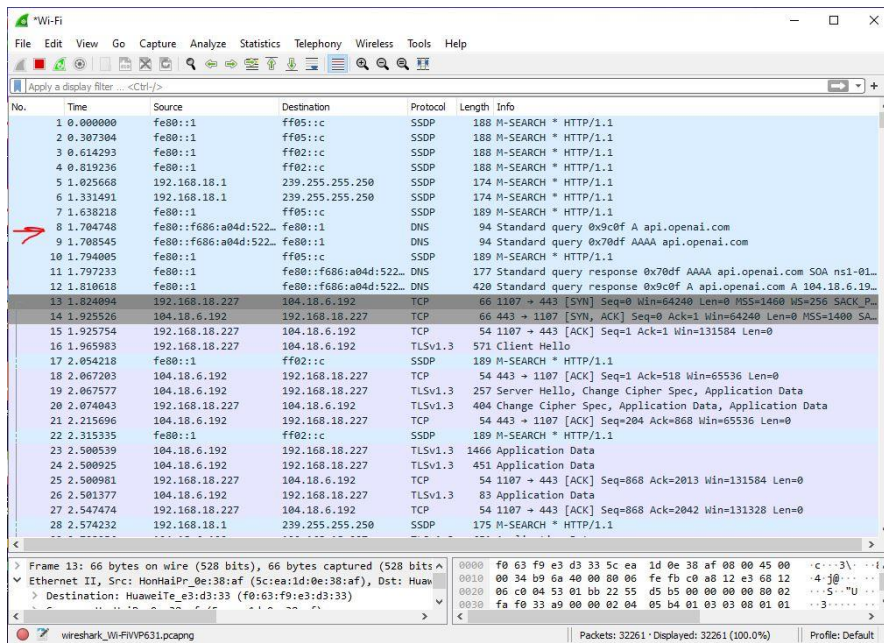
In addition to the fact that GPT-3 models do not just take input in plain English but they can also deliver output responses in various languages and formats.

## Question 2

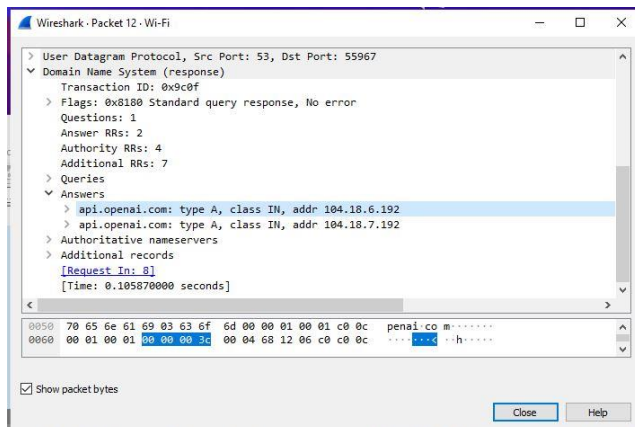
### Post Request [without filter]



### Get Request [without filter]

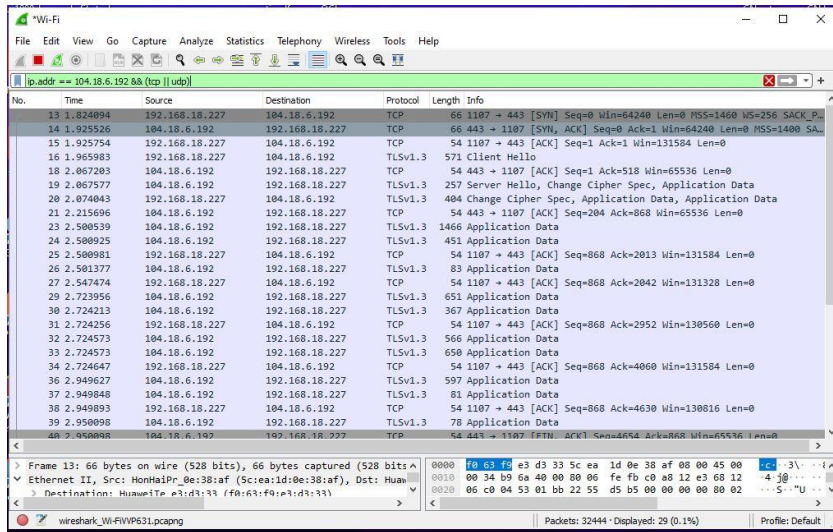


As it can be analyzed here, from Packet 8 the DNS request is sent to obtain the IP Address of api.openai.com. In Packet 12, this IP Address is returned.



This is Packet 8 and this return the IP Address of OPEN AI API, which is 104.18.6.192.

Now, this IP Address will be used by the next TCP Requests. Hence a filter was applied:



In the filter we restricted the IP Address and the protocol to be UDP or TCP and we obtained the following packets.

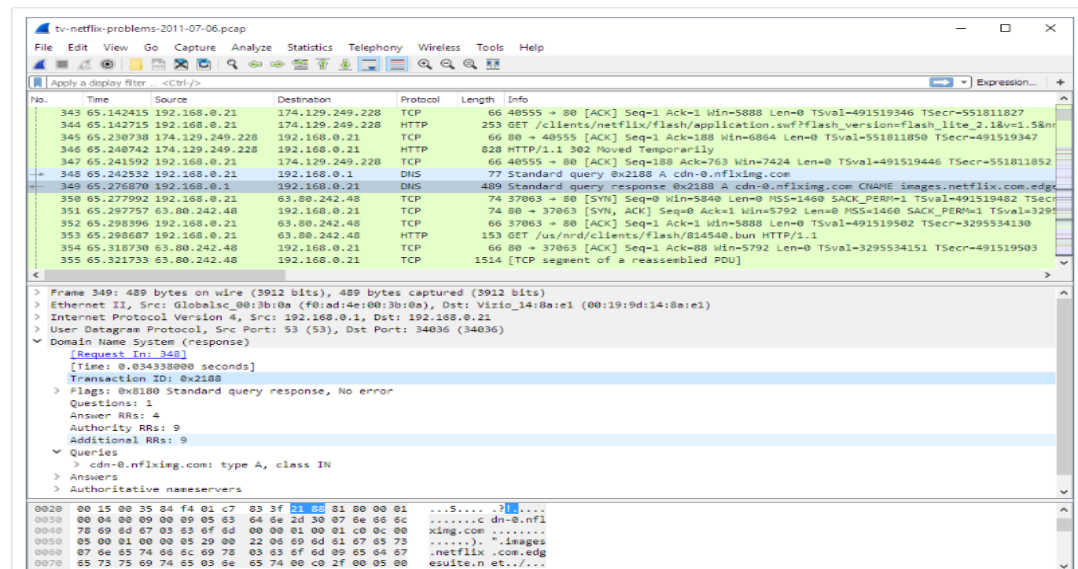
In the TCP Packets we can see a 3-way handshake in packets 16,19,20. In packet 20, some payload is also sent. SYN packets in Packet 13,14 are used to establish the connection and similarly at the end, the FIN packets 40,43 are used to tear down the connection.

## Benefits and Purposes of Wireshark for network traffic analysis

Wireshark is a type of network packet sniffer. More formally, it is a network protocol analyzer, or an application that captures packets from a network connection. It does 3 main things:

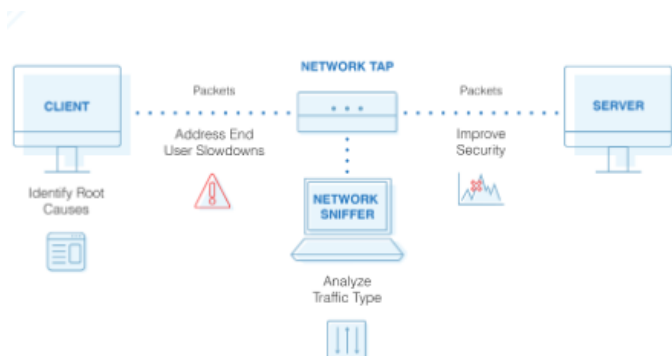
- Packet capture: Wireshark sniffs packets over a particular connection and grabs entire traffic captures.
- Filtering. By using various filters on the type of protocols, or ports or IP Addresses, Wireshark is able to filter out packets.
- Visualization. Wireshark displays the entire information of a packet by visually displaying the entire conversation and network streams.

Figure 1.1. Wireshark captures packets and lets you examine their contents.



As it can be seen here, several packets are captured, and their source and destination IP Addresses, what protocol they work on and further detail by clicking on them is available to visualize them thoroughly.

Briefly, packet sniffing can be described in the following image.



Wireshark can be used to for network security purposes.

1. To determine the root cause of network problems

Packet Sniffers such as Wireshark collect all the information from across your traffic and evaluate your network paths. Therefore, it will pinpoint the network traffic bottlenecks. It measures the network response time, which is also known as network latency. Thus, it will determine the time required for specific information to travel from the sender to the receiver. Packet sniffing can identify the affected applications so that administrators can fix the problem.

2. Analyze traffic by type

As mentioned above, Wireshark contains filtering capabilities which can be used to segregate content flowing over different categories such as on types of protocols or on IP Addresses. This



in turn is useful for capacity planning for network administrators, for future network growth and optimize network resources.

### 3. Enhance security

A packet sniffer can identify if there is an unusual spike in your network traffic. It can indicate if some intruder is trying to apply for illegal communication or transfer a large amount of data. Wireshark can be used to detect potential security breaches and suspicious network activity. It allows users to identify malicious traffic, such as viruses, malware, and unauthorized access attempts. Thus, it provides you with network security and minimizes cybercriminal work.

### 4. Improve bandwidth

To understand where your network bandwidth is used, Wireshark can help visualize data packets consuming this bandwidth. A Wi-Fi packet sniffer can retrieve performance and monitor your network security. Hence, you can detect potential issues and resolve the downtime.

## Question 3

### Concept Of SSL and Its Importance in Website Security

SSL stands for Secure Socket Layer. Basically, while a client exchanges data with a server over the internet, that data stands at a risk of being intercepted and misused by various third parties. That is where SSL comes in. In short, it's a security protocol for keeping an internet connection secure and safeguarding any sensitive data that is being sent between two systems.

Websites that employ an https in their url (an extra s at the end which stands for secured) make use of SSL by obtaining an SLL certificate. This means that any data transferred between users and sites, or between two systems remain impossible to read. It uses encryption algorithms to scramble data in transit, preventing hackers from reading it as it is sent over the connection. This information could be anything sensitive or personal which can include credit card numbers and other financial information, names and addresses.

### The Different Types of SSL Certificates

There are multiple types of SSL Certificates available with different features.

#### 1. Single Domain SSL Certificates

A single-domain SSL certificate applies to one domain and one domain only. It cannot be used to authenticate any other domain, not even subdomains of the domain it is issued for. However, all pages on ta domain employing this type are covered by this certificate, for instance, if cloudflare.com has a single-domain certificate, then cloudflare.com/learning (the Learning Center main page) is also covered by that certificate.

#### 2. Wildcard SSL Certificates



Wildcard SSL certificates are for a single domain and all its subdomains. A subdomain is under the umbrella of the main domain. Usually subdomains will have an address that begins with something other than 'www.' For example, www.cloudflare.com has a number of subdomains, including blog.cloudflare.com, support.cloudflare.com, and developers.cloudflare.com. Each is a subdomain under the main cloudflare.com domain.

### 3. Multi-Domain SSL Certificates (MDC)

A multi-domain SSL certificate, or MDC, lists multiple distinct domains on one certificate. With an MDC, domains that are not subdomains of each other can share a certificate.

## Process Of Generating a Certificate Signing Request (CSR)

A CSR is necessary to secure web traffic. A certificate signing request (CSR) is an encoded file containing information about your website, service, organization, and domain name. This information is used by a Certificate Authority (CA) to create an SSL/TLS certificate for your website to encrypt traffic to your site.

Creating an SSL/TLS certificate with a CSR is a two-step process:

1. Generate a private key and public key pair. This can be done using a variety of tools, such as OpenSSL.

A public key is a mathematical value that is used to encrypt data. The CSR contains your public key, which the CA uses to create your certificate and verify the signature on your CSR. Public keys are generated using a cryptographic algorithm, such as RSA.

A public key is mathematically derived from a private key. Together, these are referred to as a “key pair.” Private keys can be used to decrypt data that was encrypted by the corresponding public key, and public keys can be used to verify digital signatures created by the corresponding private key.

Many web servers and runtime environments, such as Internet Information Services (IIS), have CSR generation capabilities built in. Another way to generate a private key is to use the OpenSSL command line tool. This will generate a private key file called example.com.key that is 2048 bits long.

2. Create a CSR using the private key. This step will generate a CSR file, which you must submit to a CA.

Once you have generated a private key, you can use it to create a CSR file. This file will contain the above information and is typically encoded using Base-64. The CSR file can also be generated using the OpenSSL command line tool.

This CSR file must now be submitted to a Certificate Authority to generate the SSL certificate.

## Identify The Different SSL Certificate Providers and Their Pricing Plans.

Multiple types of Certificate Providers are in the market. Some of these include Symantec, GeoTrust, DigiCert and many more.

### Symantec

- Symantec Secure Site

This starts from \$272.41 a year and secures up to 24 Domains.

- Symantec Secure Site Pro

This starts from \$661 a year.

Symantec Secure Site Pro comes with strong 256-bit encryption to protect users' data as well free vulnerability assessment and daily malware scan features will continuously examine your web pages for malicious activities.

- Symantec Secure Site Pro EV

This starts from \$991 a year.

Symantec Secure Site Pro with EV SSL removes security weakness by providing robust 256-bit encryption along with green bar. The company name indicates on address bar itself ensures that the website has higher validation.

Thawte

- SSL 123 (\$149/yr)
- SSL Web Server (\$268/yr)
- SSL Webserver with EV (\$398/yr)

DigiCert

- Basic SSL (\$867.00 USD)/yr.
- Secure Site SSL (\$1,452.00 USD)/yr.
- Secure Site Pro SSL (\$2,414.00 USD) a year

## Process Of Installing and Configuring SSL In Different Types of Web Servers

Before installing/configuring the SSL Certificate, it needs to be ensured that the CSR is saved along with the private key, and all relevant SSL files are downloaded.

Now for different Web Servers, different processes must be followed.

- Apache

First the certificate files need to be copied to the server. Next, Find the Apache configuration file (httpd.conf) you need to edit. Apache's main configuration file is typically named httpd.conf or apache2.conf. Identify the SSL <Virtual Host> block you need to configure. If your site needs to be accessible through both secure (https) and non-secure (http) connections, you need a virtual host for each type of connection. Configure the <Virtual Host> block for the SSL-enabled site.

This is how it will look like:

```
<VirtualHost 192.168.0.1:443>
    DocumentRoot /var/www/html2
    ServerName www.yourdomain.com
    SSLEngine on
    SSLCertificateFile /path/to/your_domain_name.crt
    SSLCertificateKeyFile /path/to/your_private.key
    SSLCertificateChainFile /path/to/DigiCertCA.crt
</VirtualHost>
```

Test your Apache configuration file before restarting.

- Nginx

The primary and intermediate SSL certificate will have been emailed after the CSR certificate was issued. These two files will have to be concatenated after they are downloaded to the server. Next, edit the Nginx virtual hosts files and add the lines below. Make a copy of the existing non-secure server module and paste it below the original.

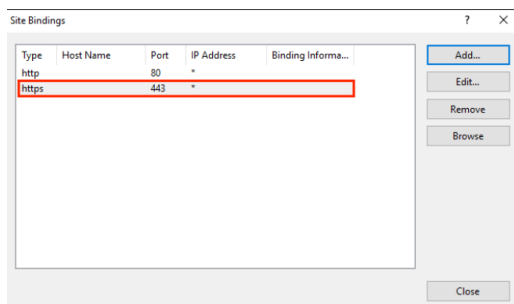
```
server {  
  
    listen 443;  
  
    ssl on;  
    ssl_certificate /etc/ssl/your_domain_name.pem; (or bundle.crt)  
    ssl_certificate_key /etc/ssl/your_domain_name.key;  
  
    server_name your.domain.com;  
    access_log /var/log/nginx/nginx.vhost.access.log;  
    error_log /var/log/nginx/nginx.vhost.error.log;  
    location / {  
        root /home/www/public_html/your.domain.com/public/;  
        index index.html;  
    }  
  
}
```

Restart the Nginx server.

## - IIS

On the server where you created the CSR, save the SSL certificate .cer file (e.g., your\_domain\_com.cer) that DigiCert sent to you. In the Windows start menu, type Internet Information Services (IIS) Manager and open it. In Internet Information Services (IIS) Manager, in the Connections menu tree (left pane), locate and click the server name. On the server name Home page (center pane), in the IIS section, double-click Server Certificates.

On the Server Certificates page (center pane), in the Actions menu (right pane), click the Complete Certificate Request... link. In the Complete Certificate Request wizard, on the Specify Certificate Authority Response page, adjust the file name, friendly name and configure to Web Hosting and then click OK. Then open Bindings, and add the SSL certificate.



Restart IIS to get started.

## Process Of Renewing SSL Certificates

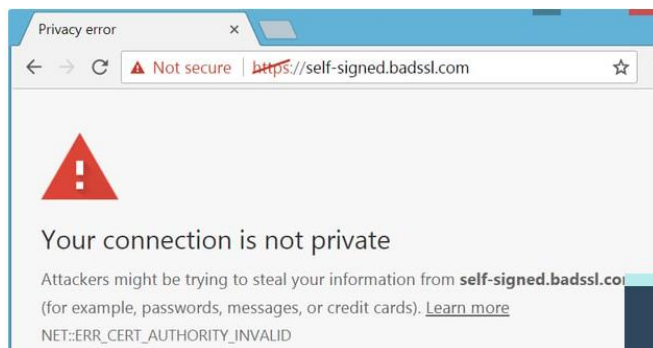
The process of renewing SSL Certificates is very similar to the process of installing and configuring them in the first place. A new CSR needs to be obtained for renewal and all the valid information needs to be obtained again. Once payment is made, the certificate is emailed.

## Common SSL Issues

Issues may arise with the issuance of the SSL Certificate. Some of these include:

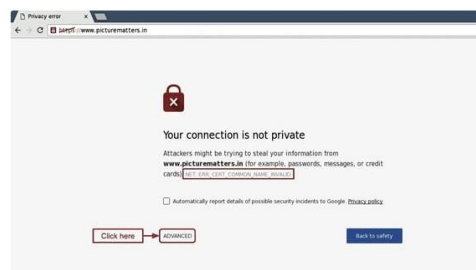
- SSL Certificate Not Trusted Error

The certificate authority (CA), is not on the browser's built-in list of trusted certificate providers or that the certificate was issued by the server itself. Either way the SSL certificate is not trusted by the browser.



- Name Mismatch Error

This error indicates that the domain name in the SSL certificate doesn't match the URL that was typed into the browser. This message can be caused by something as simple as "www." Say the certificate is registered for www.yoursite.com and you type in https://yoursite.com. Then you'll get an SSL certificate name error.



- Generic SSL Protocol Error

This error is particularly tricky to resolve because there are multiple potential causes, including: an improperly formatted SSL certificate that the browser cannot parse, a certificate that is not properly installed on the server, faulty, unverified, or lack of digital signature, the use of an outdated encryption algorithm, a firewall or other security software interfering with the SSL protection, a problem in the certificate's chain of trust, the series of certifications that make up your site's SSL encryption.

## How To Troubleshoot

Typically problems can be fixed with an online tool. To start, use an online tool to identify the problem causing the SSL certificate error on your site. You can use a tool like SSL Checker, SSL Certificate Checker, or SSL Server Test, which will verify that an SSL certificate is installed and not expired.

If the problem is that your CA is not trusted, then you may need to install at least one intermediate certificate on your web server. Intermediate certificates help browsers establish that the website's certificate was issued by a valid root certification authority.

If you're still getting a certificate not trusted error, then you could have installed the certificate incorrectly. In that case, you can generate a new CSR from your server and reissue it from your certificate provider.

If you're getting a name mismatch error, then the problem may be your IP address. When you type your domain name into your browser, it first connects to your site's IP address and then goes to your site. Usually, a website has its own IP address. But if you use a type of web hosting other than dedicated hosting, your site may be sharing an IP address with multiple sites. If one of those websites does not have an SSL certificate installed, then a browser might not know which site it's supposed to visit and display a mismatch name error message. To resolve the issue, you can upgrade to a dedicated IP address for your site.

List of Resources Consulted:

- Problems and Solutions to Common SSL Issues  
<https://blog.hubspot.com/website/fix-ssl-certificate-error#:~:text=Generic%20SSL%20Protocol%20Error&text=a%20faulty%2C%20unverified%2C%20or%20lack,up%20your%20site's%20SSL%20encryption.>
- How to install SSL Certificates on different web serves  
<https://www.digicert.com/kb/csr-ssl-installation/apache-openssl.htm>
- Tutorial of installing SSL Certificate on Apache, IIS, Nginx  
<https://www.digicert.com/kb/ssl-certificate-installation.htm>
- Concept of SSL  
<https://www.youtube.com/watch?v=tKqSSOEjbgs>

## Question 4

### Part 1: Effect of HTTP Statelessness on Login

HTTP and HTTPS are stateless protocols, which means that each request and response pair is independent of any previous or future requests and responses. The web server does not remember if a user has login before, therefore every request that a user makes is susceptible to be interception by attackers/third-parties because login details will have to be sent again.

To solve this problem, session management can be used. Every login is treated as a separate session of which a session id is returned to the browser client, which would store that session id in a cookie. Next, for every subsequent login request, this session Id will be used.

In HTTP version 1.0, cookies were not supported natively, so websites used various workarounds such as URL rewriting to transmit session identifiers between the client and server. This approach was insecure because URLs could be intercepted and tampered with.

In HTTP version 1.1 and HTTPS, cookies are supported natively, and they are the preferred method for maintaining session state. Cookies are more secure than URL rewriting because they are stored on the client-side and can be encrypted and signed to prevent tampering.

Overall, while HTTP and HTTPS being stateless protocols can present challenges for maintaining the user's login state, websites use various techniques such as session management and cookies to address these challenges and provide a secure login experience for users.

## Part 2: Login on Different Devices

Login on different devices is dealt by session management. This is done to ensure that the user's login state remains consistent across all devices. Using a static variable instead could pose a lot of security issues because a user would use the same credentials across all devices leading to the risk of interception.

As described before, a unique session identifier is created and maintained for each login session and returned to the browser client for that particular session on different devices. This session identifier is then used to associate the user's login state with their session across different devices.

When the user logs in from a different device, the website creates a new session with a new session identifier and associates it with the user's account. The website then sends the new session identifier to the device, and the device stores it for future requests. This way, each device has a unique session identifier that is associated with the user's account, allowing them to access their account and maintain their login state across different devices.

## Part 3: IPv4 vs IPv6

IPv4 and IPv6 are both Internet Protocol versions used to identify devices on a network. IPv4 is the older protocol, and it uses 32-bit addresses to identify devices on a network. IPv6 is the newer protocol, and it uses 128-bit addresses to identify devices on a network.

Differences between IPv4 and IPv6:

- Address length: IPv4 uses 32-bit addresses, while IPv6 uses 128-bit addresses.
- Number of addresses: IPv4 provides approximately 4.3 billion unique IP addresses, while IPv6 provides an almost unlimited number of unique IP addresses.
- Address format: IPv4 addresses are written in decimal format separated by dots, while IPv6 addresses are written in hexadecimal format separated by colons.
- Address configuration: IPv4 addresses are typically statically or dynamically assigned using DHCP, while IPv6 addresses are typically dynamically assigned using DHCPv6 or autoconfigured using the Neighbor Discovery Protocol (NDP).
- Header format: IPv4 and IPv6 have different header formats, with IPv6 including additional fields to support features such as flow labeling and extension headers.

To configure IPv4, you typically need to set the IP address, subnet mask, default gateway, and DNS servers on the device. This can be done using the device's network configuration settings or by using command-line tools such as `ipconfig` on Windows or `ifconfig` on Linux.

To troubleshoot IPv4, common issues include incorrect IP configuration settings, network connectivity problems, and DNS resolution issues. You can troubleshoot these issues by checking the device's network configuration settings, checking for connectivity using tools such as `ping` or `tracert`, and checking DNS resolution using tools such as `nslookup`.

To configure IPv6, you typically need to enable IPv6 on the device and configure the device to obtain an IPv6 address using DHCPv6 or autoconfiguration using NDP. This can be done using the device's network configuration settings or by using command-line tools such as `ipconfig` or `ifconfig`.

To troubleshoot IPv6, common issues include incorrect IPv6 configuration settings, network connectivity problems, and DNS resolution issues. You can troubleshoot these issues by checking the device's IPv6

configuration settings, checking for connectivity using tools such as ping6 or traceroute6, and checking DNS resolution using tools such as nslookup with the -query=AAAA option.

In summary, IPv4 and IPv6 are different protocols used to identify devices on a network, with IPv6 providing a larger address space and additional features. Configuring and troubleshooting each protocol involves setting appropriate network configuration settings and using network diagnostic tools to identify and resolve issues.

## References

Q1

<https://www.spaceotechnologies.com/blog/advantages-disadvantages-using-openai-app-development/>

<https://platform.openai.com/docs/api-reference/models/retrieve>

Q2

<https://aardwolfsecurity.com/the-advantages-of-packet-sniffing/>

Q3

<https://www.websecurity.digicert.com/security-topics/what-is-ssl-tls-https>

<https://www.cloudflare.com/learning/ssl/types-of-ssl-certificates/>

<https://www.keyfactor.com/blog/what-is-a-certificate-signing-request-csr/>

<https://www.clickssl.net/platinum-partner-of-symantec-ssl-certificates>

<https://www.digicert.com/kb/csr-ssl-installation/apache-openssl.htm>

<https://www.digicert.com/kb/csr-ssl-installation/nginx-openssl.htm>

<https://www.digicert.com/kb/csr-creation-ssl-installation-iis-10.htm>

<https://blog.hubspot.com/website/fix-ssl-certificate-error#:~:text=Generic%20SSL%20Protocol%20Error&text=a%20faulty%2C%20unverified%2C%20or%20lack,up%20your%20site's%20SSL%20encryption.>

## Screenshots

Get Request



GET

https://api.openai.com/v1/models?model=gpt-3.5-turbo

Send

Query Headers 2 Auth 1 Body Tests Pre Run

Query Parameters

☒

model

gpt-3.5-turbo

☐

parameter

value

GET

https://api.openai.com/v1/models?model=gpt-3.5-turbo

Send

Query Headers 2 Auth 1 Body Tests Pre Run

HTTP Headers

☒

Accept

\*/\*

☒

User-Agent

Thunder Client (https://www.thunderclient.com)

☐

header

value

GET

https://api.openai.com/v1/models?model=gpt-3.5-turbo

Send

Query Headers 2 Auth 1 Body Tests Pre Run

None Basic Bearer OAuth 2 NTLM AWS

Bearer Token

sk-XxT6fXvDEA1SKHrB4tFIT3BlbkFjpVifLXhArfrxraLK1aKR

## Get Response

Status: 200 OK Size: 42.01 KB Time: 1.27 s Response

```
1 {
2   "object": "list",
3   "data": [
4     {
5       "id": "babbage",
6       "object": "model",
7       "created": 1649358449,
8       "owned_by": "openai",
9       "permission": [
10        {
11          "id": "modelperm-49FUp5v084tBB49tC4z8LPH5",
12          "object": "model_permission",
13          "created": 1669085501,
14          "allow_create_engine": false,
```


Status: **200 OK** Size: **42.01 KB** Time: **1.27 s**

Headers 

#### Response Headers

| Header            | Value                         |
|-------------------|-------------------------------|
| date              | Thu, 20 Apr 2023 20:10:14 GMT |
| content-type      | application/json              |
| transfer-encoding | chunked                       |
| connection        | close                         |
| openai-version    | 2020-10-01                    |

#### Post Request

POST 

https://api.openai.com/v1/chat/completions

Send

Query

Headers <sup>3</sup>

Auth <sup>1</sup>

Body <sup>1</sup>

Tests

Pre Run

Query Parameters

☐

parameter

value

Status: 200 OK

Size: 323 Bytes

Time: 1.41 s

Response 

1 {

POST 

https://api.openai.com/v1/chat/completions

Send

Query

Headers <sup>3</sup>

Auth <sup>1</sup>

Body <sup>1</sup>

Tests

Pre Run

HTTP Headers  Raw

☒

Accept

\*/\*

☒

User-Agent

Thunder Client (https://www.thunderclient.com)

☒

Content-Type

application/json

☐

header

value

POST 

https://api.openai.com/v1/chat/completions

Send

Query

Headers <sup>3</sup>

Auth <sup>1</sup>

Body <sup>1</sup>

Tests

Pre Run

JSON

XML

Text

Form

Form-encode

GraphQL

Binary

1 {

2 "model": "gpt-3.5-turbo",

3 "messages": [{"role": "user", "content": "Hello!"}]

4 }

Status: 200 OK

Size: 323 Bytes

Time: 1.41 s

Response 

Post Response

```
Status: 200 OK   Size: 323 Bytes   Time: 1.41 s   Response ▾

1  {
2    "id": "chatcmpl-77VE50fUuWLIDThvINuzHYtq71yYR",
3    "object": "chat.completion",
4    "created": 1682022685,
5    "model": "gpt-3.5-turbo-0301",
6    "usage": {
7      "prompt_tokens": 10,
8      "completion_tokens": 10,
9      "total_tokens": 20
10   },
11   "choices": [
12     {
13       "message": {
14         "role": "assistant",
15         "content": "Hi there! How can I assist you today?"
16       },
17       "finish_reason": "stop"
```