2. What are nginx header security and its uses. And also implement in the test.conf file

Security Headers :

HTTP header fields are a list of linefeed-separated HTTP data being sent and received by both the client program and server on every HTTP request.

They define how information sent/received through the connection are encoded (as in Accept-Encoding),

- the session verification and identification of the client (as in browser cookies, IP address, user-agent) or their anonymity thereof (VPN or proxy masking, user-agent spoofing),
- how the server should handle data (as in Do-Not-Track),
- the age of the document being downloaded, amongst others.

HTTP security headers are a very important part of website security as they protect you against different types of attacks including, XSS, SQL injection, clickjacking, etc.

Some important security headers are

1. HSTS - HTTP Strict Transport Security

This header instructs a user agent to only use HTTPs connections and it also declared by Strict-Transport-Security

This will prevent web browsers from accessing web servers over non-HTTPS connections. Currently all major web browsers support HTTP strict transport security.

We can implement HSTS in Nginx by adding the following entry in /etc/nginx/sites-available/test.conf file:

```
add_header Strict-Transport-Security 'max-age=31536000; includeSubDomains; preload';
```

It caches this information for the max-age period (typically 31,536,000 seconds, equal to about 1 year). The optional includeSubDomains parameter tells the browser that the HSTS policy also applies to all subdomains of the current domain.

2. X-Frame-Options

The X-Frame-Options header can help prevent click-jacking attacks.

This header indicates whether the web page can be rendered in a frame.

X-Frame-Options headers allow you to tell the web browser not to allow embedding of your web pages in a frame.

Can be implemented as:

```
add_header X-Frame-Options "SAMEORIGIN" always;
```

- **SAMEORIGIN** – allow your webpages to be displayed in an iframe on the same website
- **ALLOW-FROM uri** – allow your webpages to embedded in only specific domains/websites
- **DENY** – do not allow any website to embed your webpages in an iframe

3. Content Security Policy

The Content-Security-Policy header is an improved version of the X-XSS-Protection header and provides an additional layer of security.

It is a very powerful header aimed to prevent XSS and data injection attacks.

The CSP header is a way of whitelisting the things that your site is allowed to run. This includes images, stylesheets, javascript, inline styles, frames.

Content-Security-Policy: policy

Can be implemented as:

```
add_header Content-Security-Policy: "default-src 'self'; font-src *;img-src * data:; script-src
*; style-src *; *.youtube.com; *.facebook.com";
```

4. X-XSS-Protection

X-XSS also known as Cross Site Scripting header is used to defend against Cross-Site Scripting attacks.

Can be implemented as:

```
add_header X-XSS-Protection "1; mode=block" always;
```

- **X-XSS-Protection: 0** : This will disable the filter entirely.
- **X-XSS-Protection: 1** : This will enable the filter but only sanitizes potentially malicious scripts.
- **X-XSS-Protection: 1; mode=block** : This will enable the filter and completely block the page.

5. X-Content-Type-Options

The x-content-type header is also called "Browser Sniffing Protection" to tell the browser to follow the MIME types indicated in the header.

It is used to prevent web browsers such as Internet Explorer and Google Chrome from sniffing a response away from the declared Content-Type.

Can be implemented as:

add_header X-Content-Type-Options “nosniff” always;

6. Feature-Policy

The HTTP Feature-Policy header provides a mechanism to allow and deny the use of browser features in its own frame, and in content within any iframe elements in the document.

Syntax:

Feature-Policy: <directive> <allowlist>

Can be implemented as:

```
add_header Feature-Policy " microphone 'none' ; geolocation 'none' " ;
```

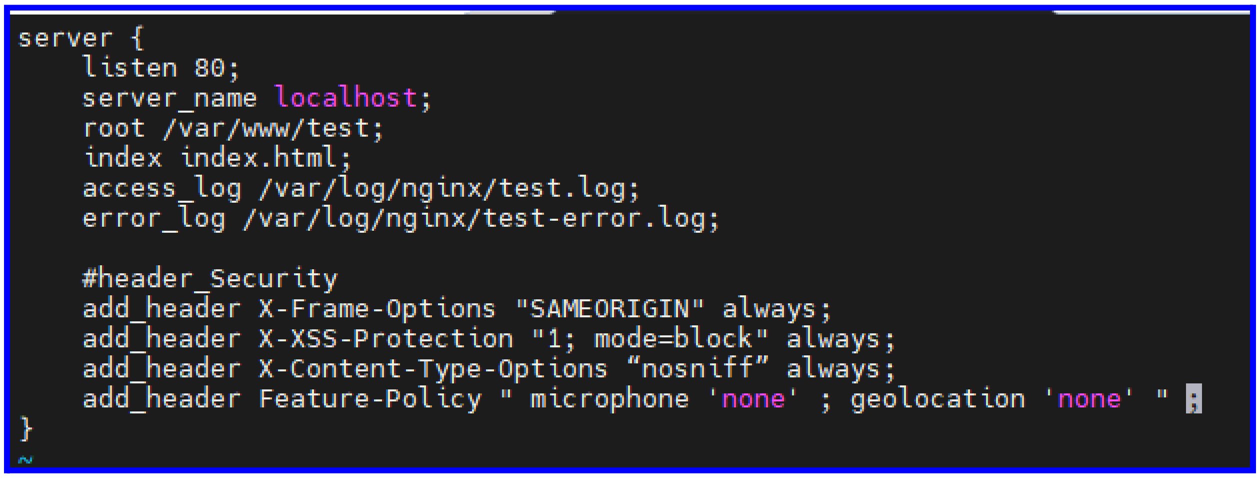
Some of Directives are:

Camera, midi, accelerometer, fullscreen,etc

Allowlist values

- ***:** The feature will be allowed in this document, and all nested browsing contexts (iframes) regardless of their origin.
- **'self':** The feature will be allowed in this document, and in all nested browsing contexts (iframes) in the same origin. The feature is not allowed in cross-origin documents in nested browsing contexts.
- **'src':** (In an iframe allow attribute only) The feature will be allowed in this iframe, as long as the document loaded into it comes from the same origin as the URL in the iframe's src attribute. Note: The 'src' origin is used in the iframe allow attribute only, and is the default allowlist value.
- **'none':** The feature is disabled in top-level and nested browsing contexts.
- **<origin(s)>**: The feature is allowed for specific origins (for example, https://example.com). Origins should be separated by a space.

We add some of these headers into our **test.conf file** to implement those security



```
server {
    listen 80;
    server_name localhost;
    root /var/www/test;
    index index.html;
    access_log /var/log/nginx/test.log;
    error_log /var/log/nginx/test-error.log;

    #header_Security
    add_header X-Frame-Options "SAMEORIGIN" always;
    add_header X-XSS-Protection "1; mode=block" always;
    add_header X-Content-Type-Options “nosniff” always;
    add_header Feature-Policy " microphone 'none' ; geolocation 'none' " ;
}
~
```

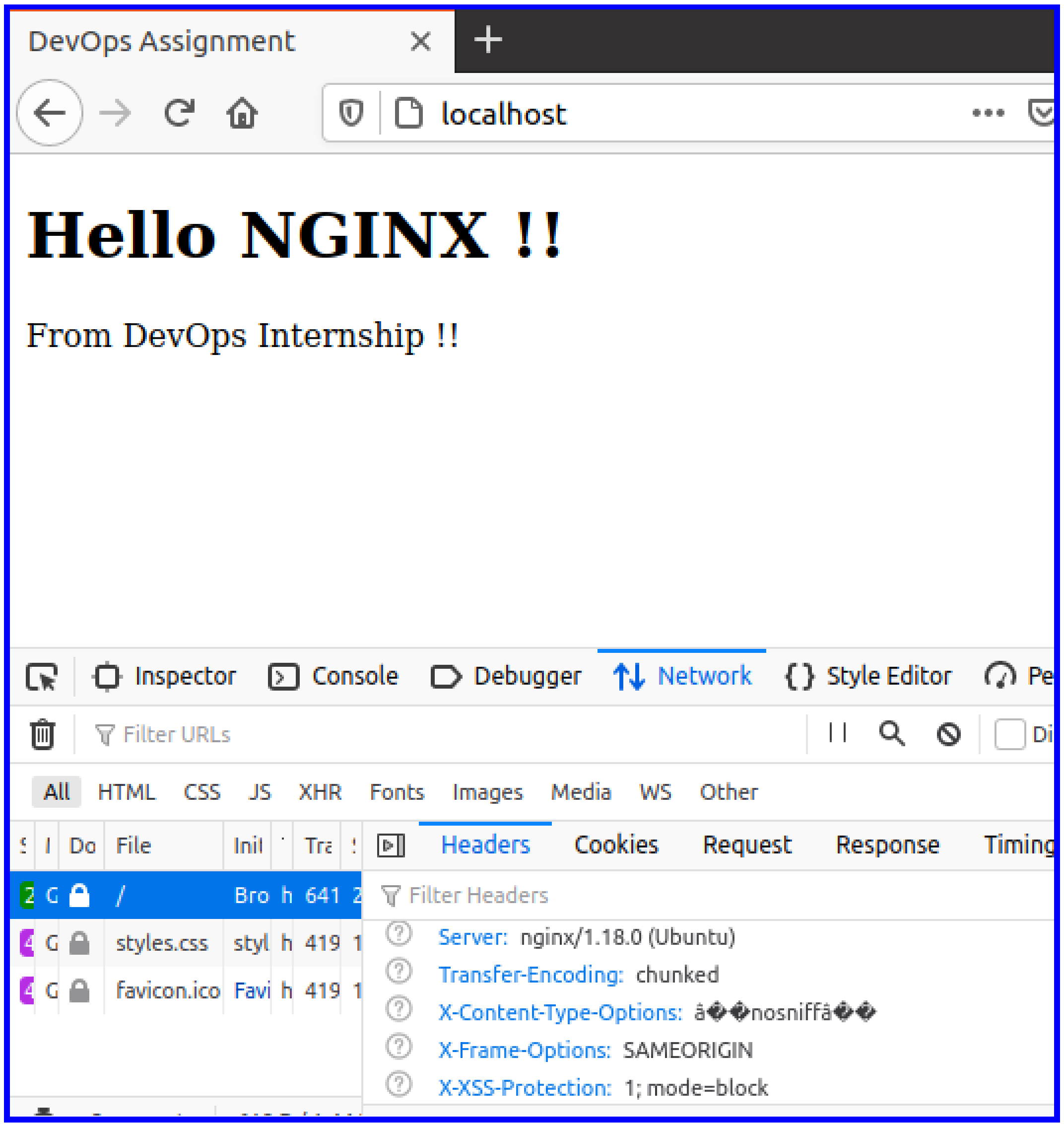
Checked the syntactical error

nginx -t

And restart the nginx service

systemctl restart nginx

And we see the output in browser with implemented header securities

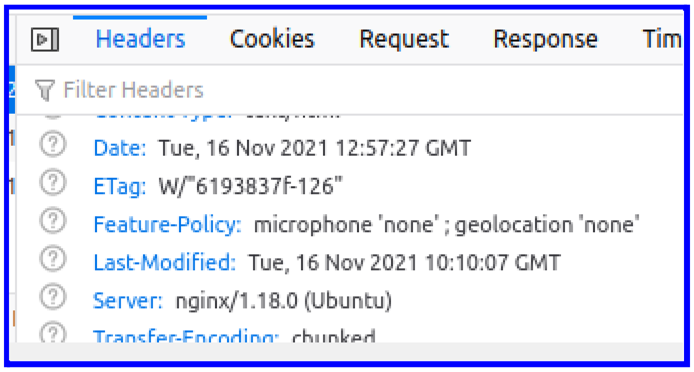


Hello NGINX !!

From DevOps Internship !!

			File			
2	G	/	Bro	h	641	2
4	G	styles.css	styl	h	419	1
4	G	favicon.ico	Favi	h	419	1

```
Server: nginx/1.18.0 (Ubuntu)
Transfer-Encoding: chunked
X-Content-Type-Options: â��nosniffâ��
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
```



```
Date: Tue, 16 Nov 2021 12:57:27 GMT
ETag: W/"6193837f-126"
Feature-Policy: microphone 'none' ; geolocation 'none'
Last-Modified: Tue, 16 Nov 2021 10:10:07 GMT
Server: nginx/1.18.0 (Ubuntu)
Transfer-Encoding: chunked
```