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

Answer:

Security Headers:
HTTP security headers are a subset of HTTP headers and are exchanged between a web client (usually a browser) and a server to specify the security-related details of HTTP communication. Some HTTP headers that are indirectly related to privacy and security can also be considered HTTP security headers. By enabling suitable headers in web applications and web server settings, one can improve the resilience of your web application against many common attacks, including cross-site scripting (XSS) and clickjacking.

Some of the important HTTP Security Headers are as follows:

a) **HTTP Strict Transport Security(HSTS):**
When enabled on the server, HTTP Strict Transport Security (HSTS) enforces the use of encrypted HTTPS connections instead of plain-text HTTP communication. A typical HSTS header might be:
Strict-Transport-Security: max-age=63072000; includeSubDomains; preload

b) **Content-Security Policy(CSP):**
The Content Security Policy (CSP) header is the recommended way to protect your websites and applications against XSS attacks. It allows you to precisely control permitted content sources and many other parameters. A basic CSP header to allow only assets from the local origin is:
Content-Security-Policy: default-src 'self'

Other directives include **script-src**, **style-src**, and **img-src** to specify permitted sources for scripts, CSS stylesheets, and images

c) **X-Frame Options:**
This header is used to provide protection against cross-site scripting attacks involving HTML iframes. To prevent the current page from being loaded into any iframes, we would use:
X-Frame-Options: deny

Other supported values are **sameorigin,** to allow loading into iframes with the same origin and **allow-from** to indicate specific URLs.

d) **X-Content-Type-Options:**
This header forces web browsers to strictly follow the MIME types specified in Content-Type headers. This protects websites from cross-site scripting attacks that abuse MIME sniffing capabilities to supply malicious code covering as a non-executable MIME type. The header has just one directive:
X-Content-Type-Options: nosniff

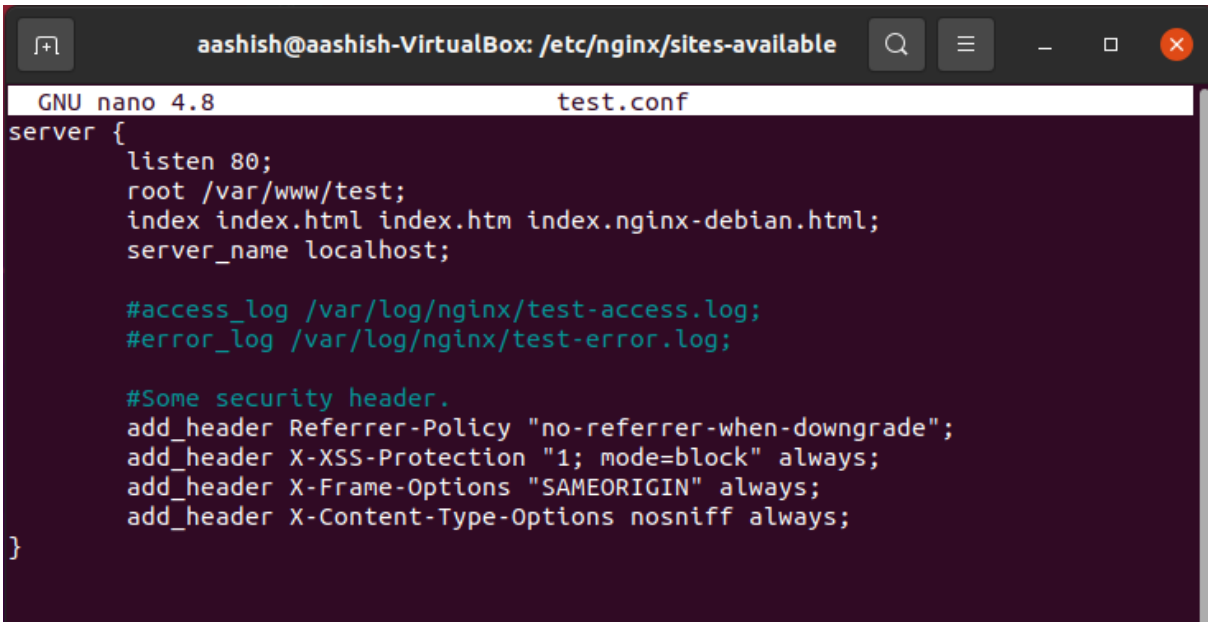
e) **Referrer-Policy:**
It controls if and how much referrer information should be revealed to the web server. Typical usage would be:
Referrer-Policy: origin-when-cross-origin

With this header, the browser will only reveal complete referrer information (including the URL) for same-origin requests. For all other requests, only information about the origin is sent.

Now, to implement the nginx header security in test.conf, we add **test.conf** in **/etc/nginx/sites-available**. We add following things in test.conf file;

```
server {
        listen 80;
        root /var/www/test;
        index index.html index.htm index.nginx-debian.html;
        server_name localhost;

        add_header Referrer-Policy “no-referrer-when-downgrade”;
        add_header X-XSS-Protection “1; mode=block” always;
        add_header X-Frame-Options “SAMEORIGIN” always;
        add_header X-Content-Type-Options nosniff always;
}
```



Now, we check syntactical error and if there is no error, we restart the nginx server using following command;

- **sudo nginx -t**
- **sudo systemctl restart nginx**

```
aashish@aashish-VirtualBox: /etc/nginx/sites-available
aashish@aashish-VirtualBox:/etc/nginx/sites-available$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
aashish@aashish-VirtualBox:/etc/nginx/sites-available$
```

```
aashish@aashish-VirtualBox: /etc/nginx/sites-available
aashish@aashish-VirtualBox:/etc/nginx/sites-available$ sudo systemctl status ngi
nx
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset:>
     Active: active (running) since Wed 2021-11-17 00:09:34 +0545; 10s ago
       Docs: man:nginx(8)
    Process: 2547 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_proce>
    Process: 2548 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (c>
   Main PID: 2549 (nginx)
      Tasks: 2 (limit: 4638)
     Memory: 2.3M
     CGroup: /system.slice/nginx.service
             ├─2549 nginx: master process /usr/sbin/nginx -g daemon on; master_>
             └─2550 nginx: worker process

नवम्बर 17 00:09:34 aashish-VirtualBox systemd[1]: Starting A high performance we>
नवम्बर 17 00:09:34 aashish-VirtualBox systemd[1]: Started A high performance web>

aashish@aashish-VirtualBox:/etc/nginx/sites-available$ ^C
```

Lastly, we test it on the web browser and inspect the page. Then we click on the network and check the response header.