1. Utilize HTTPS: Configurar o Nginx para usar HTTPS em vez de HTTP. Isso garante que os dados sejam criptografados durante a transmissão.

```
server {
        listen       8080;
        server_name  localhost;
        # Redireciona todas as requisições HTTP para HTTPS
        return 301 https://$server_name$request_uri;
```

2. HSTS: É uma política de segurança que instrui os navegadores a sempre usar HTTPS ao se comunicarem com o servidor

```
# Ativação do HSTS
add_header Strict-Transport-Security "max-age=31536000;
includeSubDomains" always;
```

max-age especifica por quanto tempo o navegador deve lembrar-se de usar HTTPS (no exemplo, é definido como 1 ano, em segundos). includeSubDomains é uma opção opcional que instrui os navegadores a aplicar a política também aos subdomínios.

3. CSP: Implementar uma Política de Segurança de Conteúdo (CSP) para mitigar ataques de XSS (Cross-Site Scripting) e outros ataques baseados em conteúdo.

```
 location / {
            root   html;
            index  index.html index.htm;
          #CSP
          add_header Content-Security-Policy "default-src 'self';
script-src 'self' 'unsafe-inline' 'unsafe-eval'; style-src 'self'
'unsafe-inline'; img-src 'self'; font-src 'self'; frame-src 'none';
object-src 'none'";
        }
```

- default-src 'self': Define a origem padrão para carregar todos os tipos de recursos, como imagens, fontes, scripts, etc., do próprio domínio.
- script-src 'self' https://dominio-permitido.com: Permite a execução de scripts apenas do próprio domínio e do domínio https://dominio-permitido.com.
- style-src 'self' 'unsafe-inline': Permite apenas estilos do próprio domínio e também permite estilos inline (geralmente não recomendado devido a vulnerabilidades potenciais de XSS).

4. Limitação de taxas de solicitação: Configurar limites de taxa para solicitações HTTP para proteger contra ataques de negação de serviço (DDoS).

```
    limit_req_zone $binary_remote_addr zone=one:10m rate=1r/s;
    server {
        listen       8080;
        server_name  localhost;
        location / {
            root   html;
            index  index.html index.htm;
            # Aplica o rate limiting
            limit_req zone=one burst=5;
        }
```

limit_req_zone: Define a zona de limitação de solicitação.
limit_req: Aplica a limitação de taxa de solicitação na localização especificada.

5. Filtragem de IP: Implementar listas de permissão e listas negras de IP para permitir ou bloquear acessos com base nos endereços IP dos clientes.

```
#filtragem de ip
allow 192.168.1.1;
allow 10.0.0.0/24;
deny all;
```

6. Ocultar informações de versão: Configure o Nginx para não divulgar informações de versão na resposta do servidor, o que pode ajudar a proteger contra ataques direcionados a versões específicas.

```
    #remover o cabeçalho com a versão do nginx
    server_tokens off;
    #customizar a tela de erro do nginx, pois a padrão tem a versão do
nginx
    error_page 404 /404.html;
    location = /404.html {
        internal;
    }
```

7. Monitoramento e registros: Configure logs detalhados para monitorar o tráfego e as atividades do servidor

```
    #customizar um formato de log
    log_format meu_formato '$remote_addr - $remote_user [$time_local]
"$request" '
                           '$status $body_bytes_sent "$http_referer" '
                           '"$http_user_agent" "$http_x_forwarded_for"';
    #onde os logs vão ser registrados.
    access_log /var/log/nginx/access.log meu_formato;
    error_log /var/log/nginx/error.log;
```

8. Reverse Proxy: Configure um reverse proxy. O reverse proxy pode ser utilizado por várias razões, e uma delas é a Ocultação de Servidores de Destino e Filtragem e Inspeção de Tráfego.

```
    server {
        listen 80;
        server_name seu_dominio.com;

        location / {
            proxy_pass http://localhost:8080;
            proxy_set_header Host $host;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_set_header X-Forwarded-Proto $scheme;
        }
    }
```

ele vai escutar as solicitações na porta 80, e o nginx vai encaminhar as solicitações para http://localhost:8080.

9. Evitar injeção de SQL:

```
location / {
            root   html;
            index  index.html index.htm;
          # Aplica o rate limiting
          limit_req zone=one burst=5;
          # verificar injeção de sql
          if ($query_string ~* "(<|%3C).*script.*(>|%3E)|sql") {
                return 403;
            }
        }
```

$query_string (que contém a string de consulta da URL).
Expressão regular que procura >, %3E, <, %3C, ou sql estão na string de consulta, caso sim, retornar 403 que significa proibido.

10. Limitação de tamanho de solicitação e corpo de resposta.

```
http {
    client_max_body_size 10m; # Limita o tamanho máximo do corpo da
solicitação para 10 MB
    client_body_buffer_size 128k; # Define o tamanho máximo do buffer de
corpo da solicitação
    client_header_buffer_size 1k; # Define o tamanho máximo do buffer de
cabeçalho da solicitação
    large_client_header_buffers 4 4k; # Define o número e o tamanho
máximo dos buffers de cabeçalho da solicitação
```