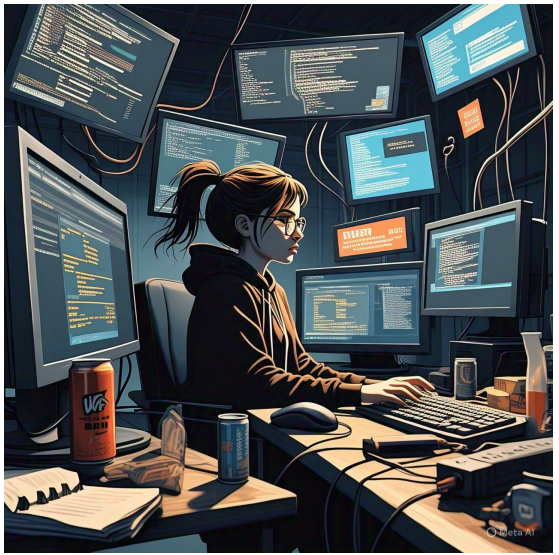
Teste para padrões técnicos modernos de Internet e segurança: IPv6, DNSSEC, TLS, HTTPS e HSTS

Alguns conceitos e dados importantes antes de começarmos:

- Internet não é um lugar, não é um site, nem tão pouco uma pessoa ou instituição. Então o que é a Internet?

- Existem duas tecnologias muito importantes e que sem elas, a Internet não existiria. Você sabe quais são elas?



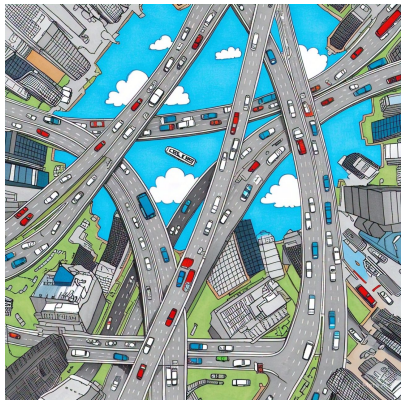
- Internet não é um lugar, não é um site, nem tão pouco uma pessoa ou instituição. Uma analogia: a Internet é como as rodovias que te levam aos lugares que deseja ir e o destino pode ser sua casa, um mercado, hospital, por exemplo. Neste caso os destinos seriam: os sites/serviços que você está acessando.

- A Internet conecta coisas, pessoas a sistemas, serviços e a outras pessoas. Ela é o caminho entre todos.

- A Internet possibilita acesso ilimitado a conhecimento espalhado pelo mundo.

- A **Internet é a maior obra da humanidade**. Palavras do meu amigo Thiago Ayub da Sage Networks.

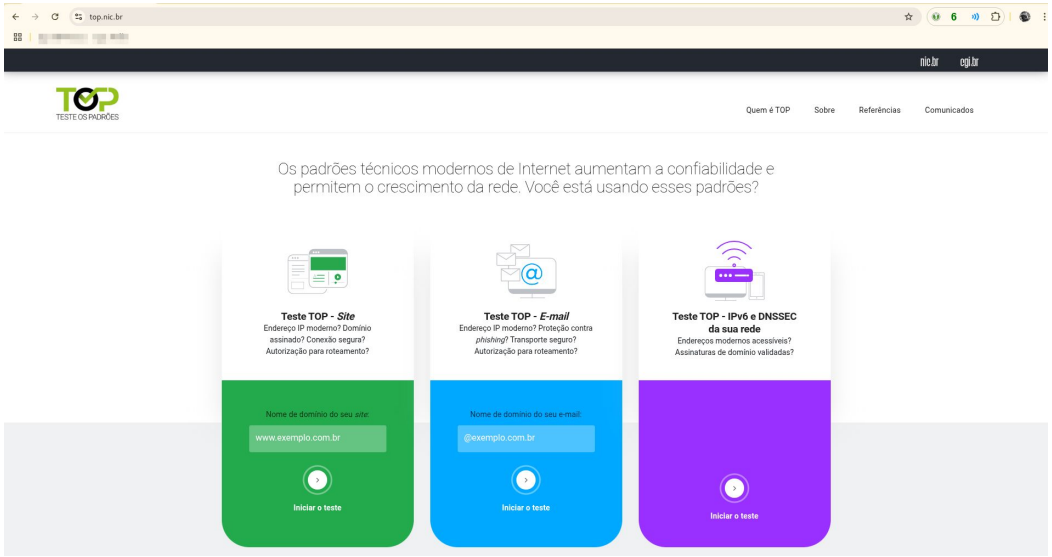
- Se pegar um caminho duvidoso na Internet, pode acabar em um site malicioso e ser vítima de um golpe digital.



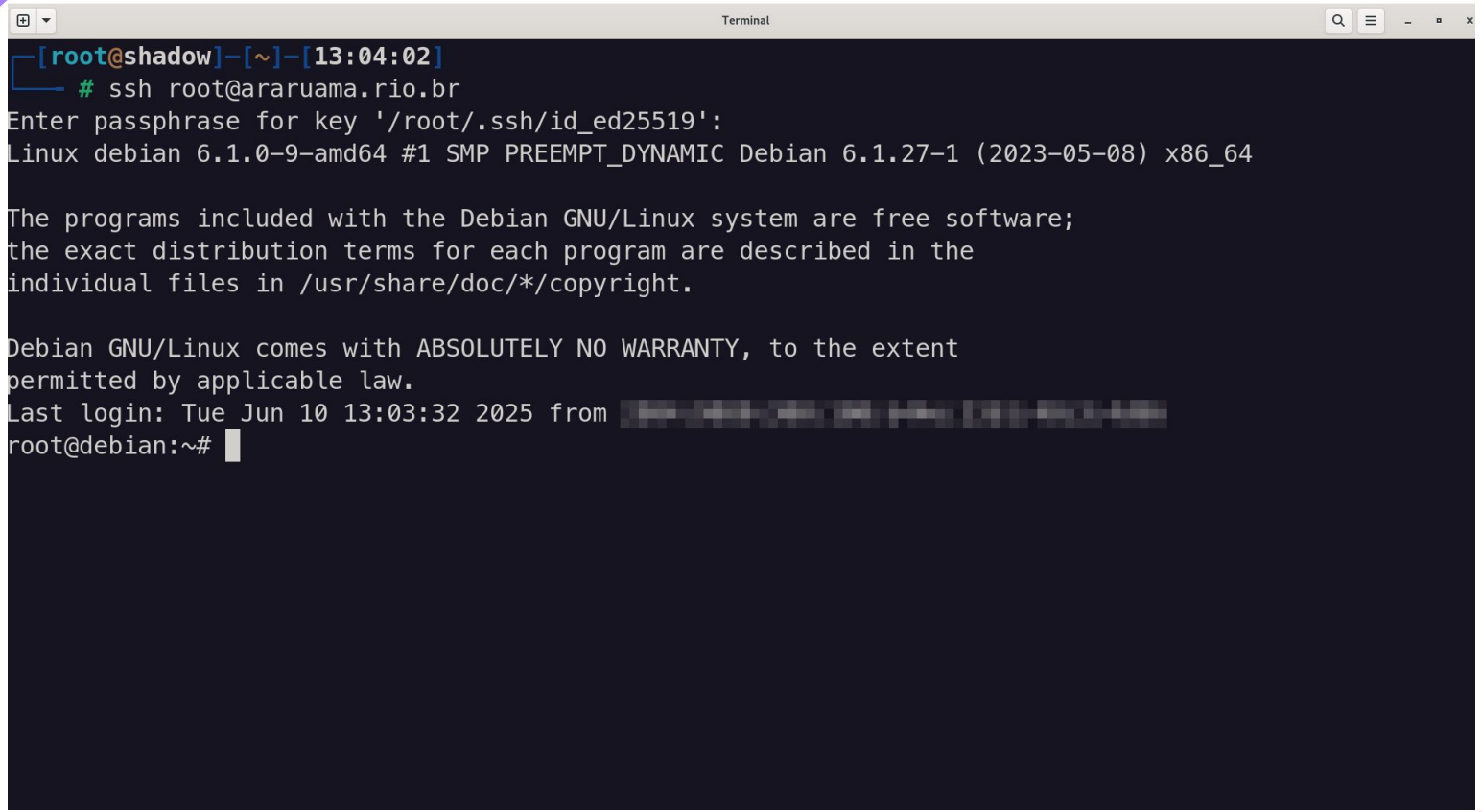
- Se você pensou: **BGP (Border Gateway Protocol)** e **DNS (Domain Name System)**. Acertou!
- O BGP é o protocolo que interliga todos os **Sistemas Autônomos** no mundo, formando assim a **Internet**.
- DNS faz a tradução de nomes em IPs e vice e versa. Bem, isso era suficiente no passado, hoje ele agrega outras funções extremamente importantes para a segurança dos serviços na Internet.

Vamos falar agora do cartão de visitas digital do seu negócio, o seu site. Não basta o seu site ser bem atrativo e informativo, ele precisa ser alcançável em toda a Internet (IPv4 e IPv6), precisa ser seguro pois ali podem passar dados pessoais e sigilosos dos visitantes.

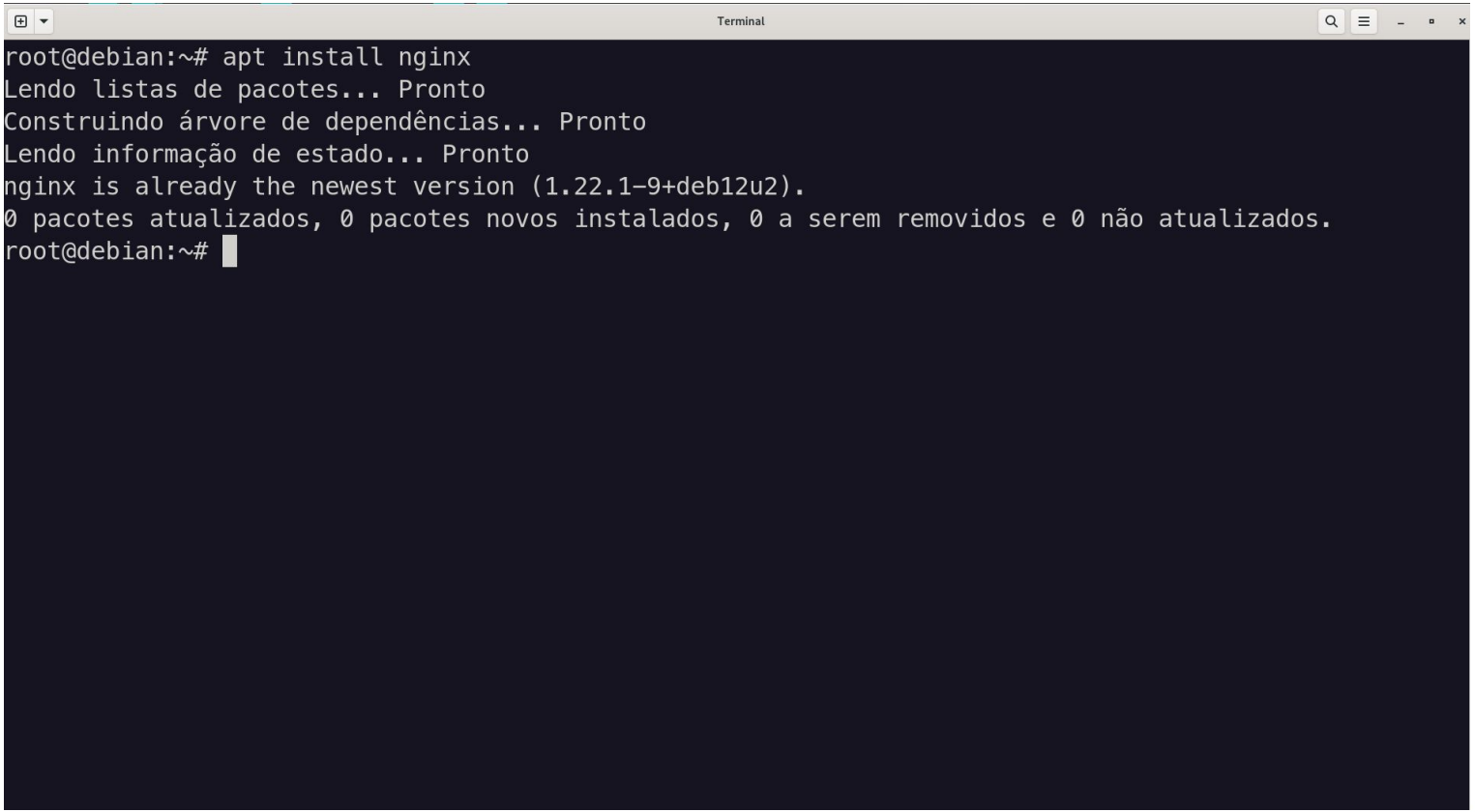
Para vermos na prática todos os problemas e melhorias, pegamos uma **VPS (Virtual Private Server)** e apontamos o domínio **araruama.rio.br** e o subdomínio **www.araruama.rio.br**, no DNS Autoritativo responsável, para a VPS. Para testarmos os padrões técnicos modernos e a segurança do nosso site de exemplo, usaremos a ferramenta **TOP** do **NIC.br** em https://top.nic.br.



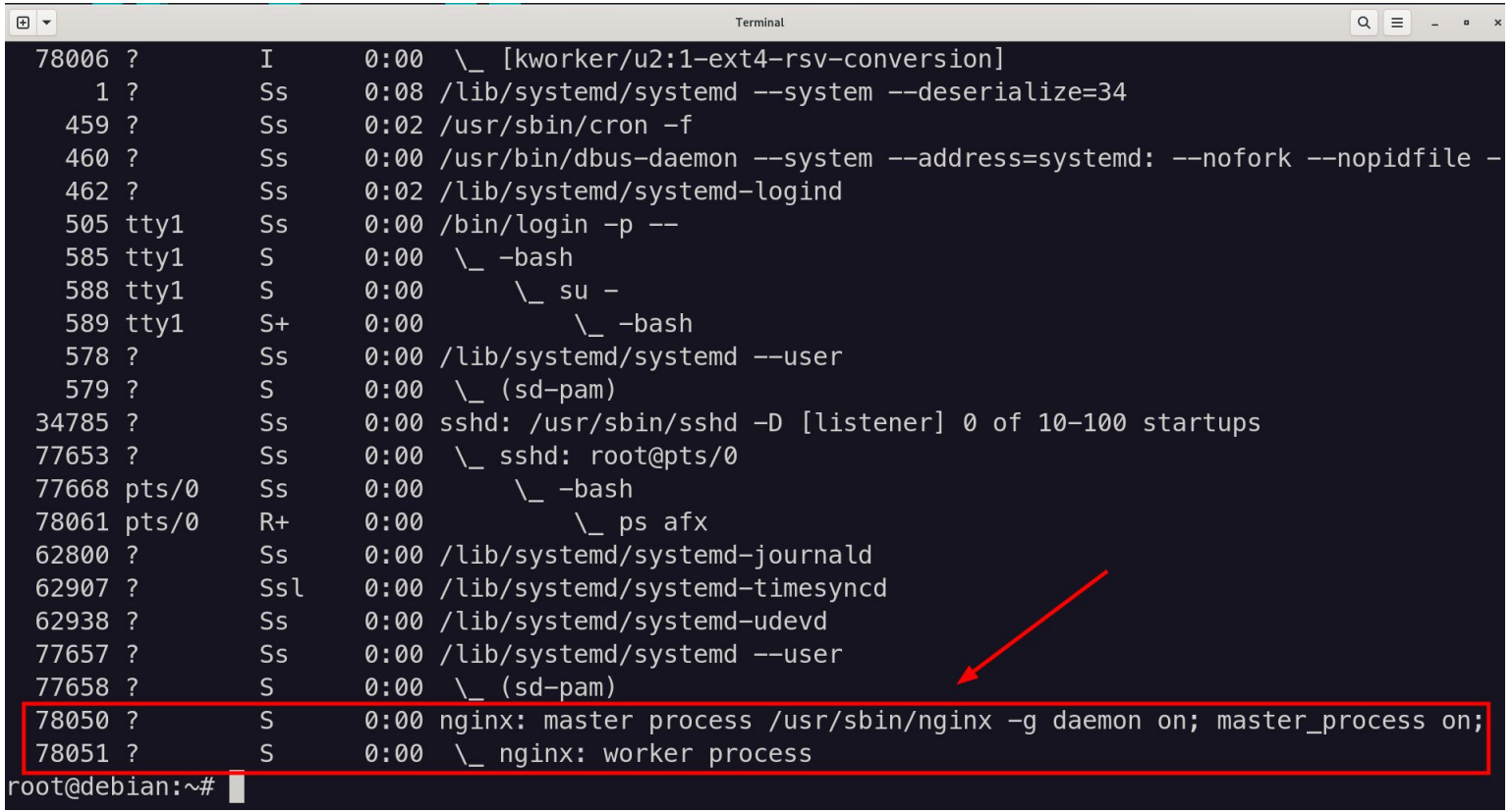
Acessando nosso VPS Debian GNU/Linux Bookworm:



Instalando o Nginx que será nosso primeiro servidor Web de testes:



Após a instalação vamos checar se o nginx está rodando com o comando: **ps afx**:



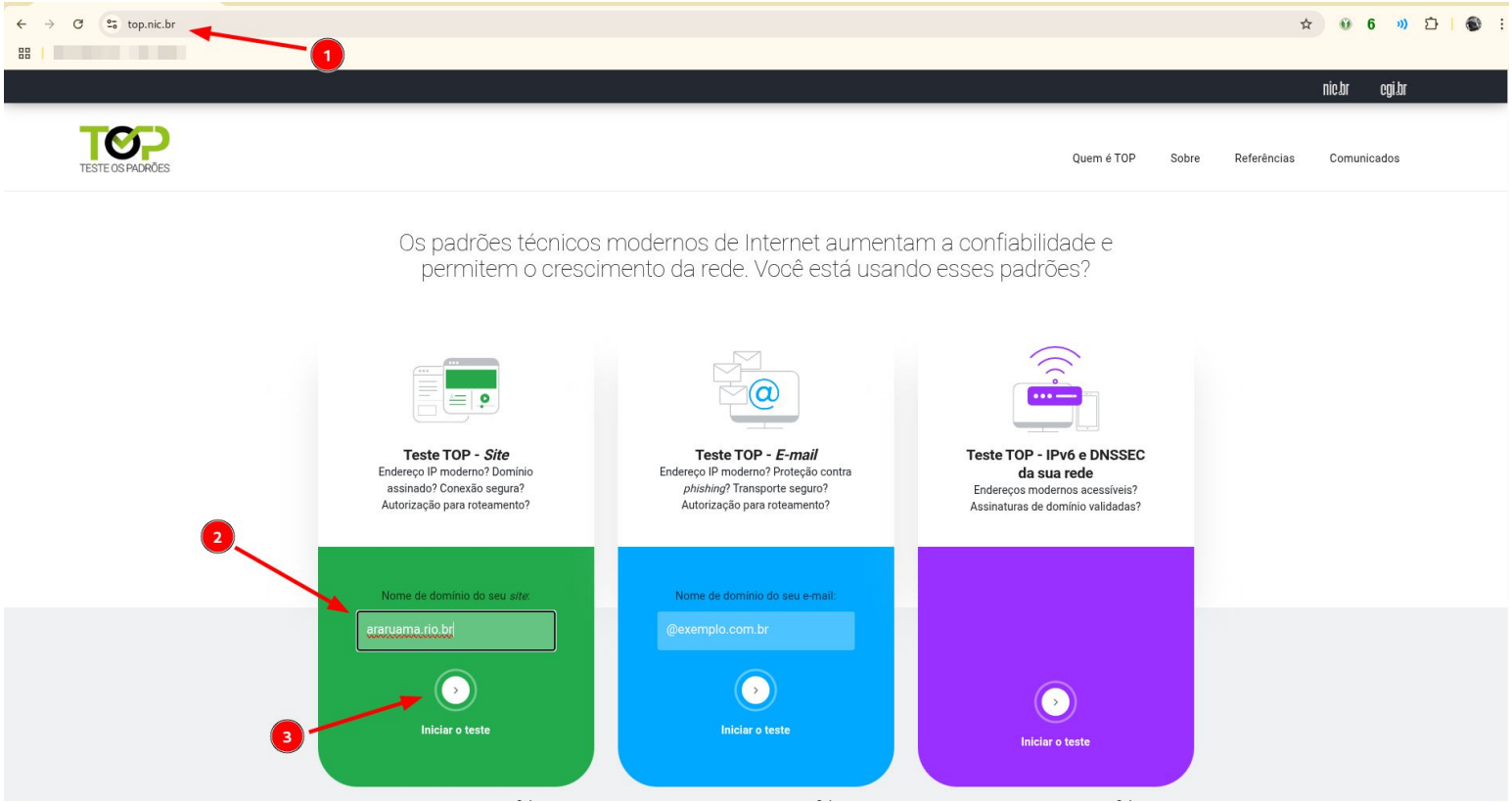
Será que está carregando algum site? Vamos checar em nosso navegador.

Vamos criar um **/var/www/html/index.html** para carregar uma outra página no navegador:

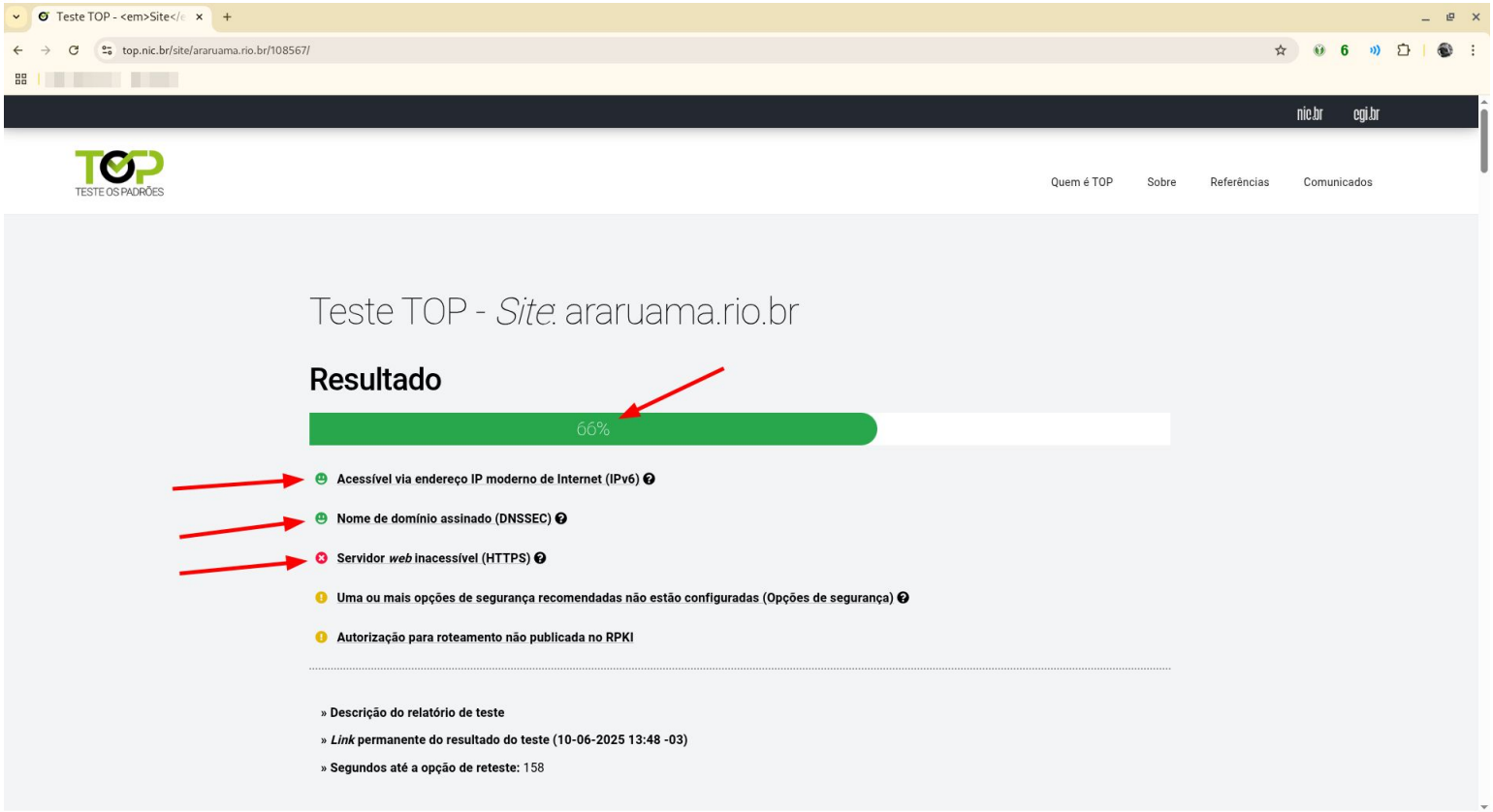
```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Semana de Capacitação 10</title>
  <style>
    body {
      margin: 0;
      height: 100vh;
      background-color: #f4f4f4;
      font-family: Arial, sans-serif;
      display: flex;
      align-items: center;
      justify-content: center;
      text-align: center;
      flex-direction: column;
      padding: 20px;
    }
    h1 {
      font-size: 2.5em;
      margin-bottom: 0.5em;
      color: #333;
    }
    p {
      font-size: 1.2em;
      color: #555;
      max-width: 600px;
    }
  </style>
</head>
<body>
  <h1>Semana de Capacitação 10</h1>
  <p>
    Teste para padrões técnicos modernos de Internet e segurança: IPv6, DNSSEC, TLS, HTTPS e HSTS
  </p>
</body>
</html>
```

Como podemos observar, o conteúdo da página mudou mas ainda estamos inseguros. Será?

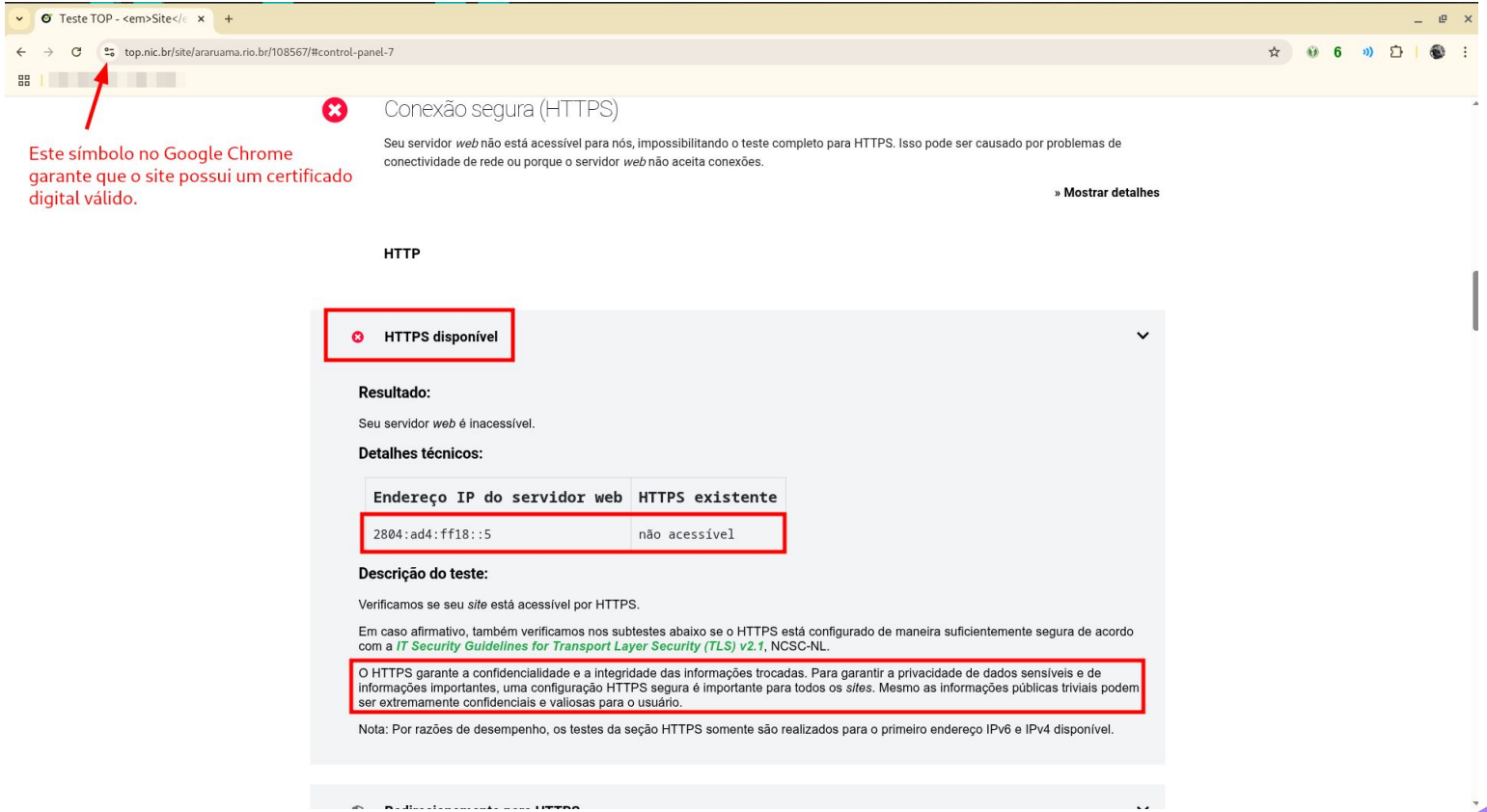
Vamos agora utilizar a ferramenta **TOP** do **NIC.br**, para avaliar nosso site de testes:



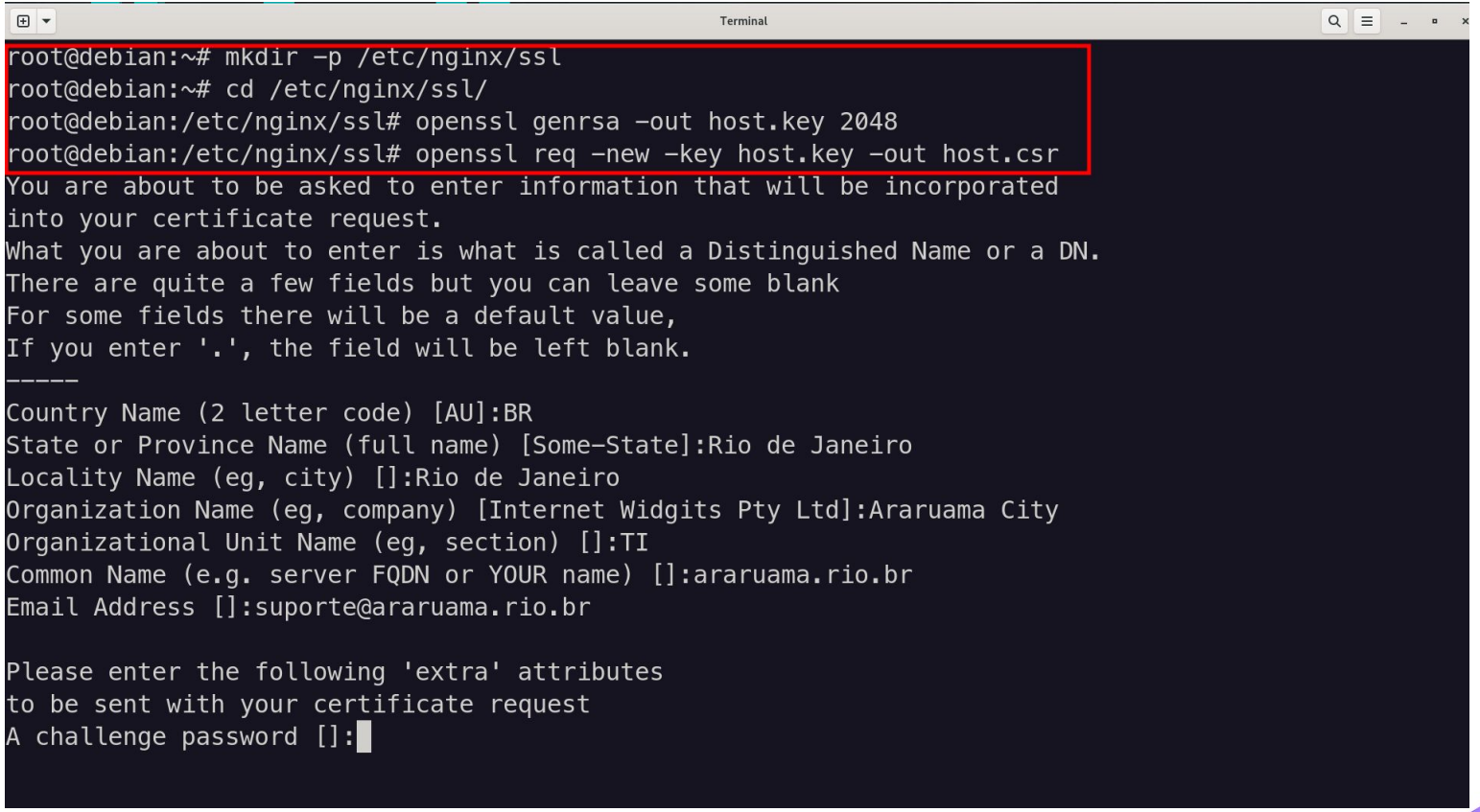
Nosso objetivo é alcançar os 100% e nesse caso chegamos em 66%. Poderia ser bem pior o resultado se o site não tivesse endereçamento **IPv6** e se **araruama.rio.br** não tivesse **DNSSEC**.



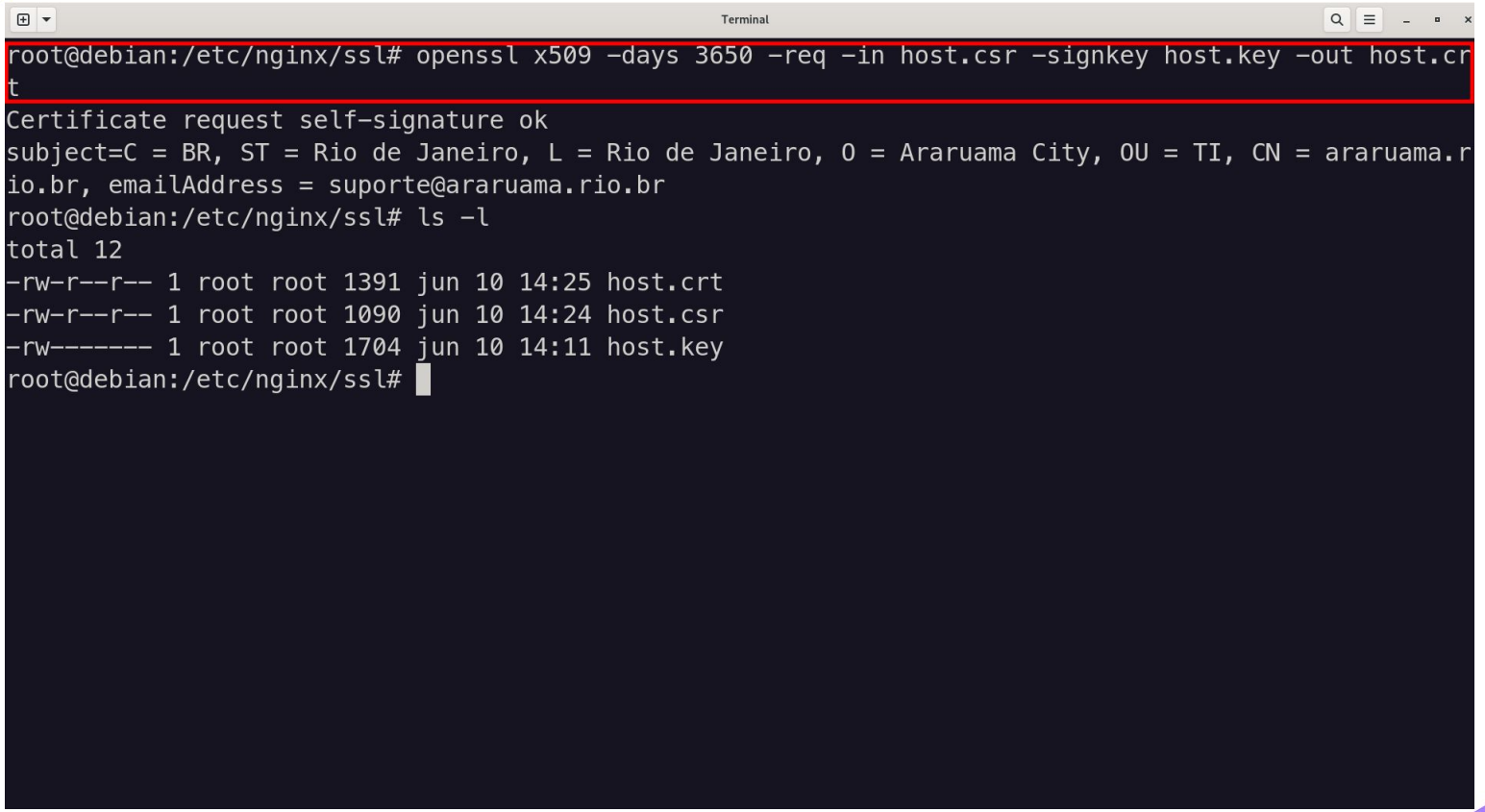
Para alcançarmos os **100%** de pontuação, precisamos resolver todos os problemas com **X**.



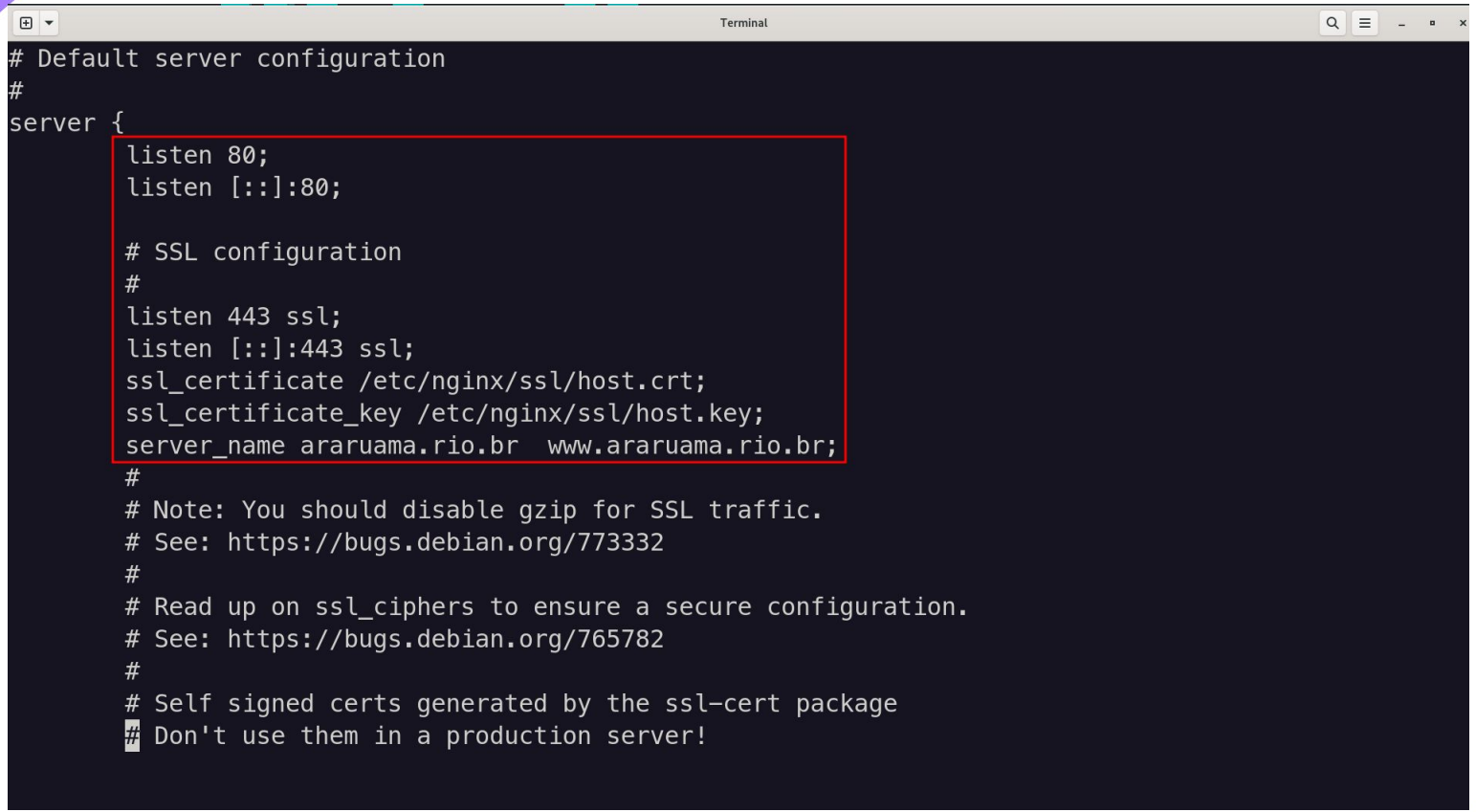
Vamos criar agora um certificado digital auto-assinado:



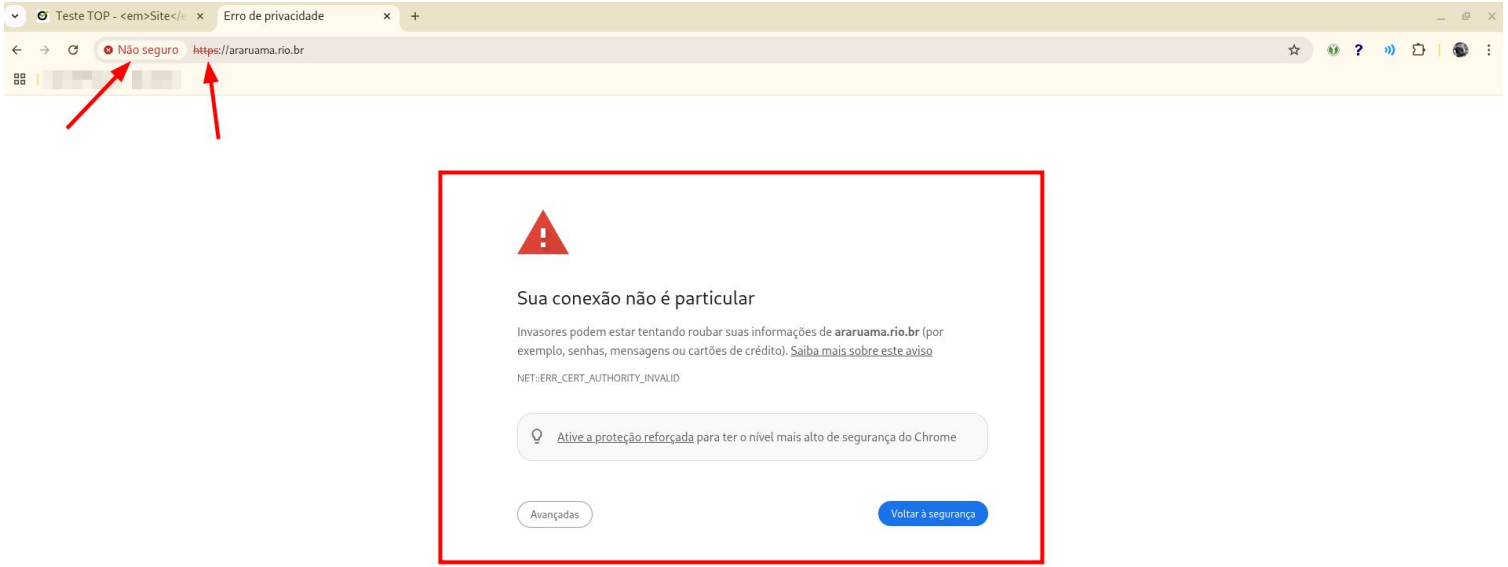
Vamos criar agora um certificado digital auto-assinado:



Agora que temos o certificado auto-assinado, vamos configurar o Nginx editando o arquivo **/etc/nginx/sites-available/default**:

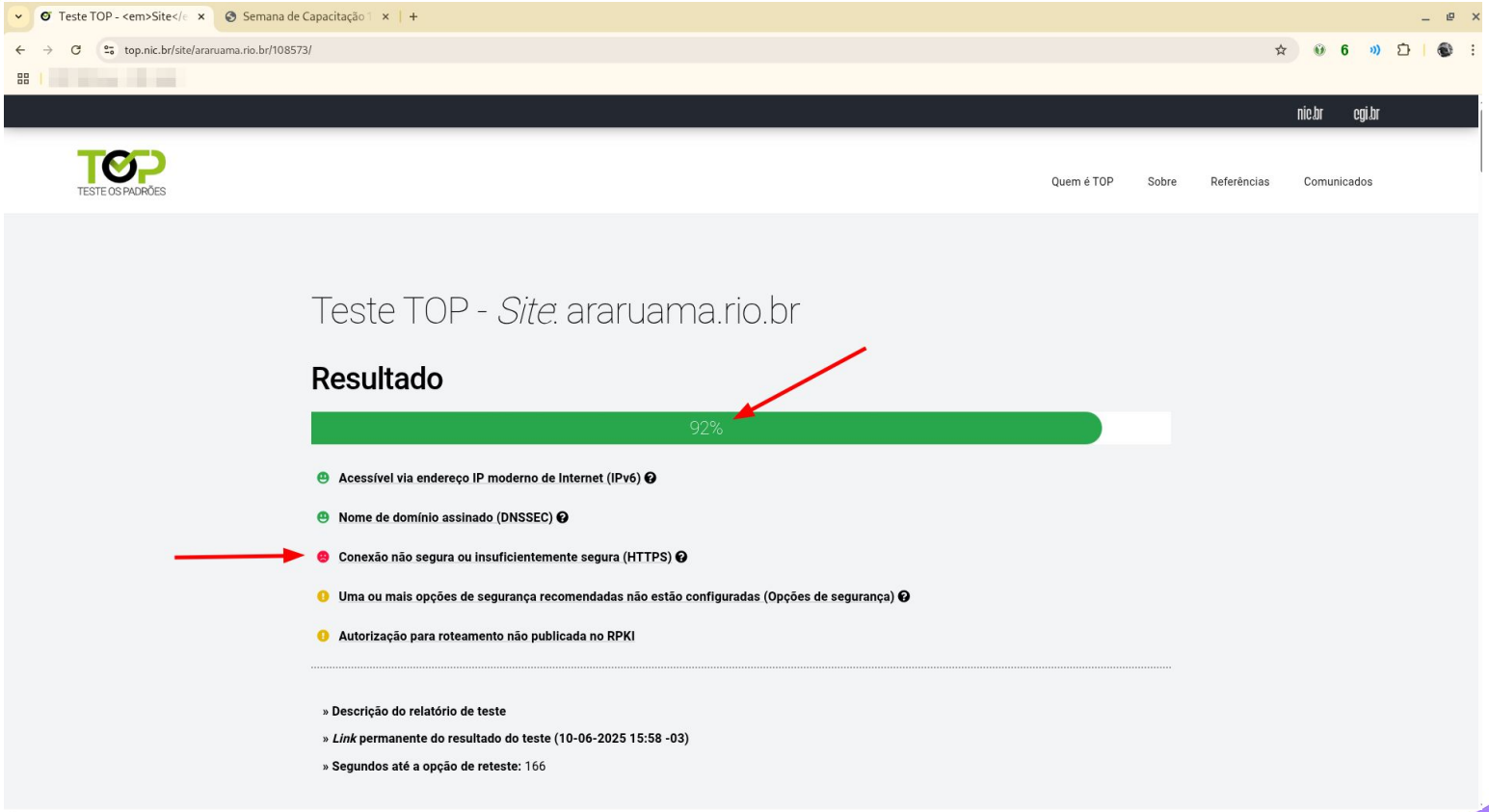


Após fazer um restart no nginx vamos acessar **https://araruama.rio.br**:

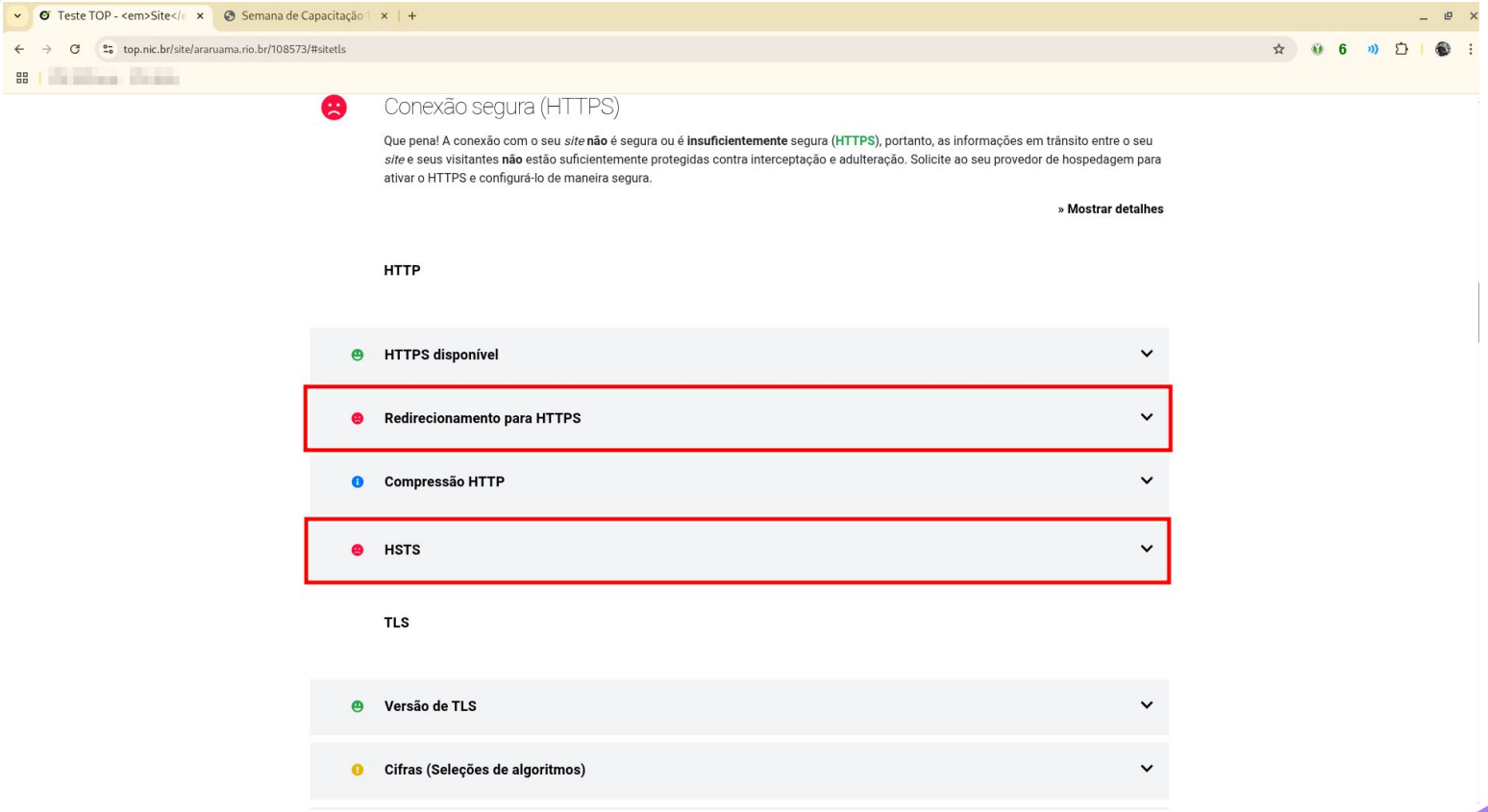


Mesmo estando **criptografado** (**HTTPS**), ainda não está seguro. Vamos olhar o TOP?

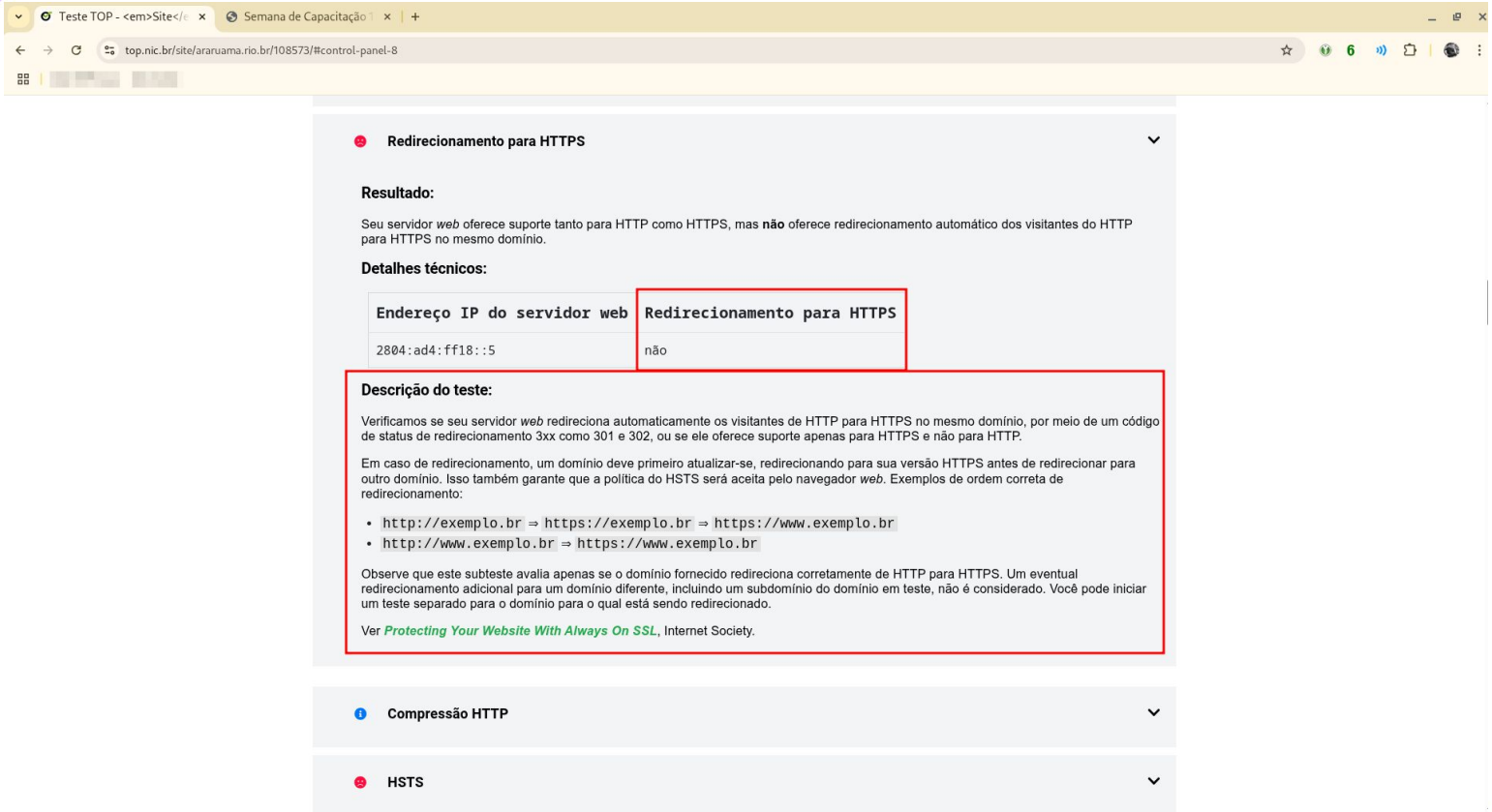
Agora alcançamos a marca de 92%. Queremos os 100% mas agora temos uma carinha vermelha.



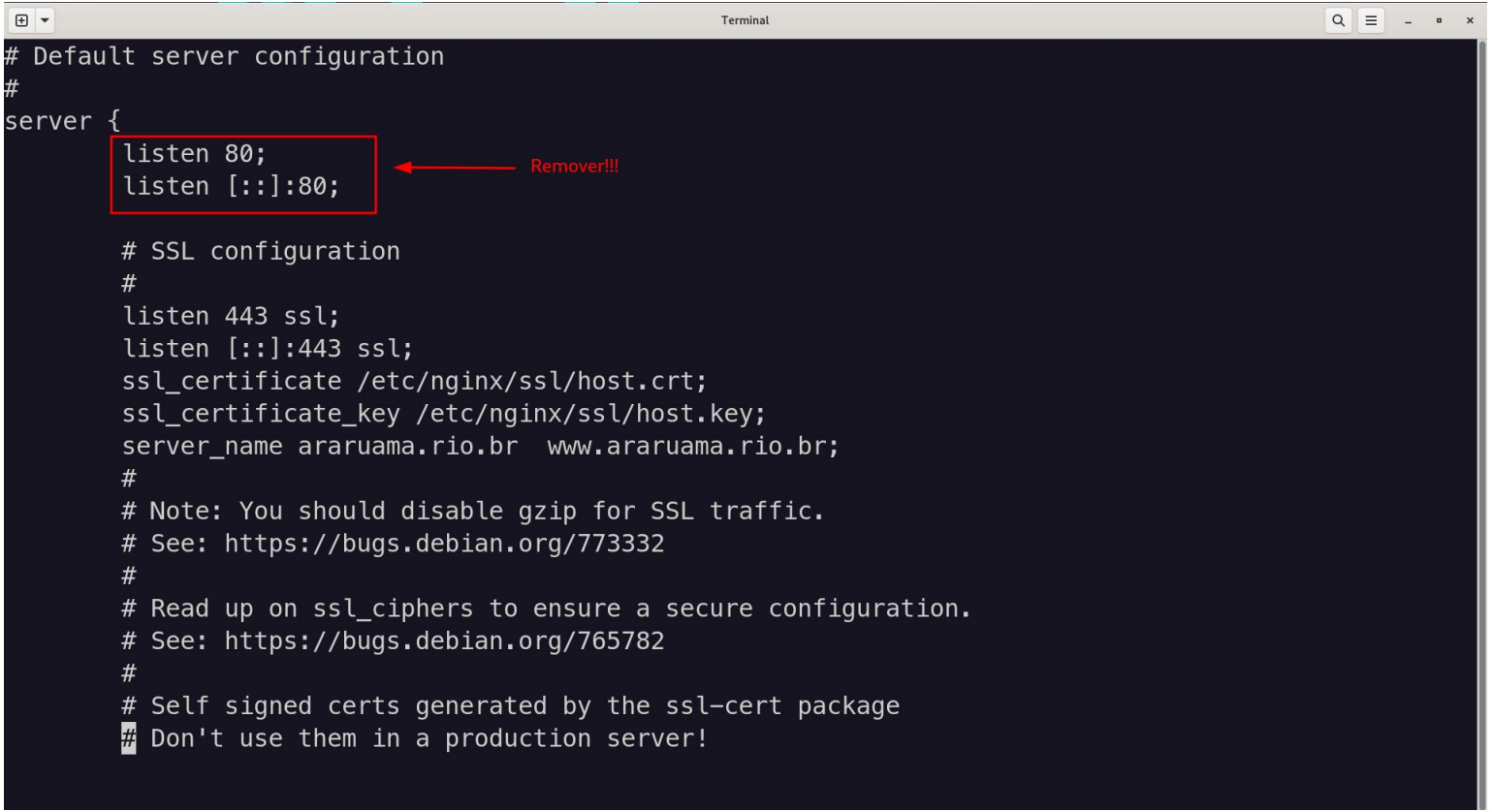
Mesmo usando HTTPS, ainda temos 2 penalidades. Vamos ver e resolver uma por uma.



Precisamos fazer um redirecionamento automático de **HTTP (80/TCP)** para **HTTPS (443/TCP):**



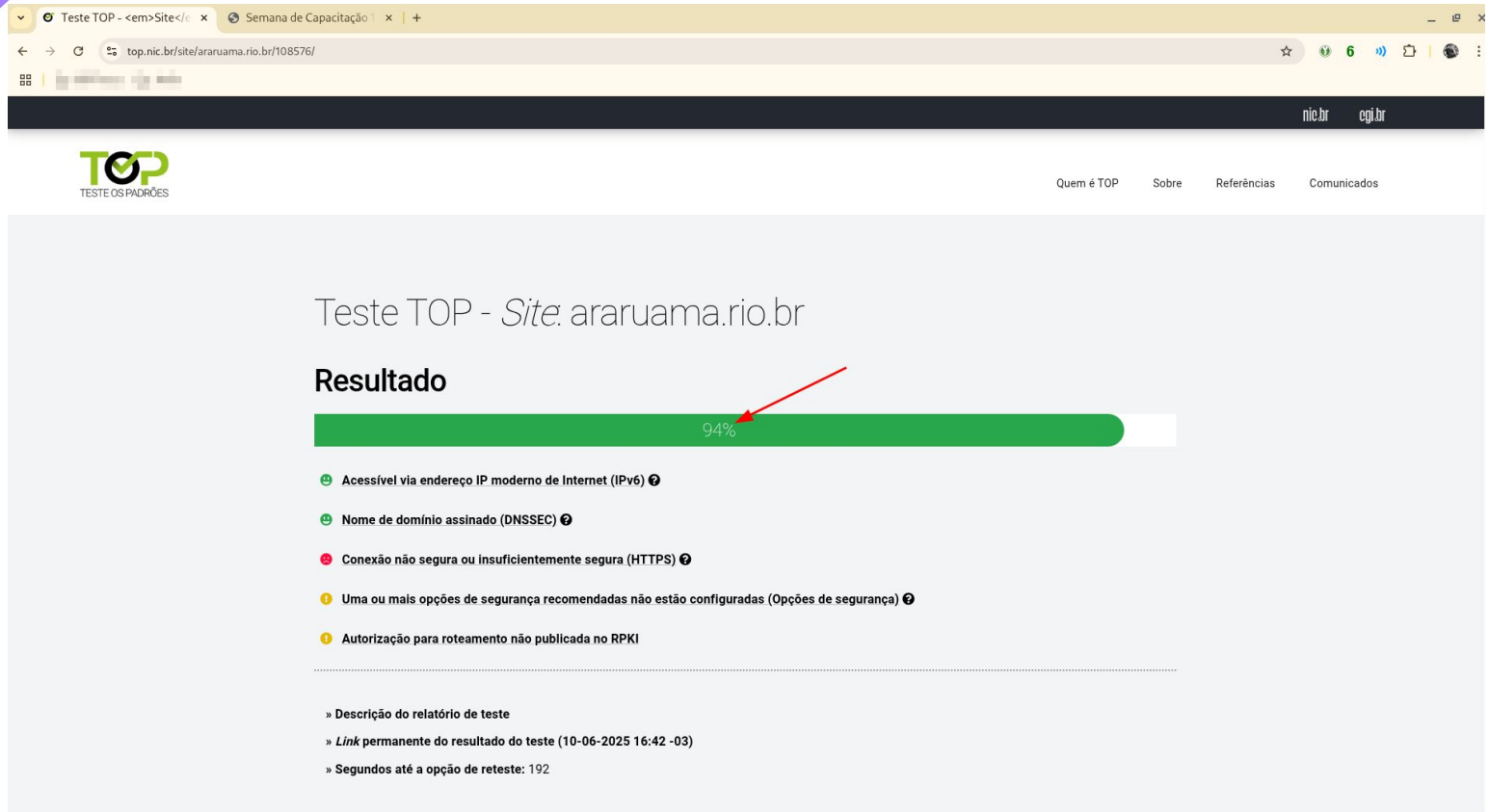
No arquivo **/etc/nginx/sites-available/default** teremos que remover no topo da nossa configuração e deixar somente o ssl:



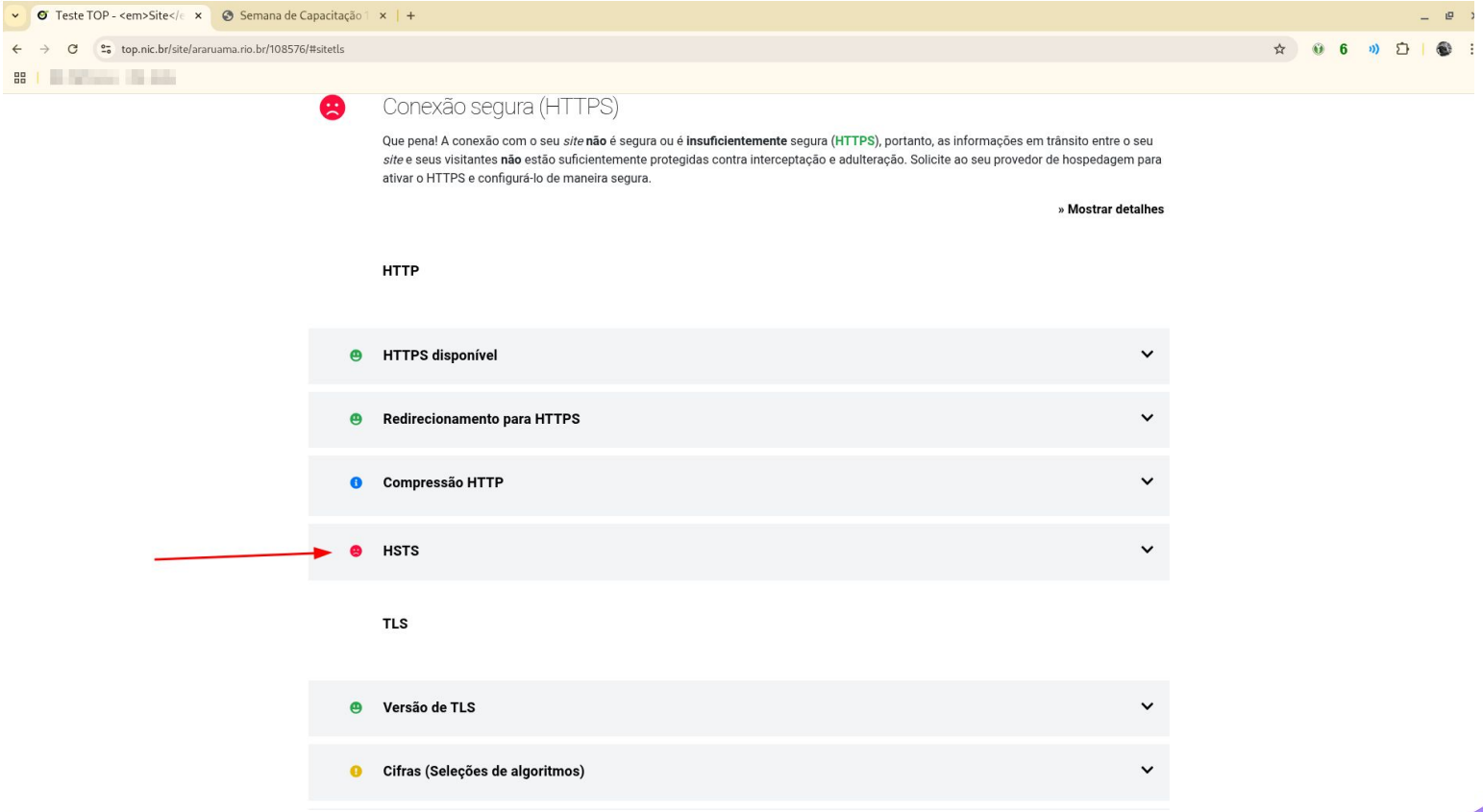
No final dele vamos incluir o seguinte código abaixo. vai fazer com que tentativas como: **http://araruama.rio.br** ou **http://www.araruama.rio.br** se tornem **HTTPS**.

```
#       root /var/www/example.com;
#       index index.html;
#
#       location / {
#               try_files $uri $uri/ =404;
#       }
#}
server {
    if ($host = www.araruama.rio.br) {
        return 301 https://$host$request_uri;
    }
    if ($host = araruama.rio.br) {
        return 301 https://$host$request_uri;
    }

    server_name www.araruama.rio.br araruama.rio.br;

    listen 80;
    listen [::]:80;
    return 404; # managed by Certbot


}
```

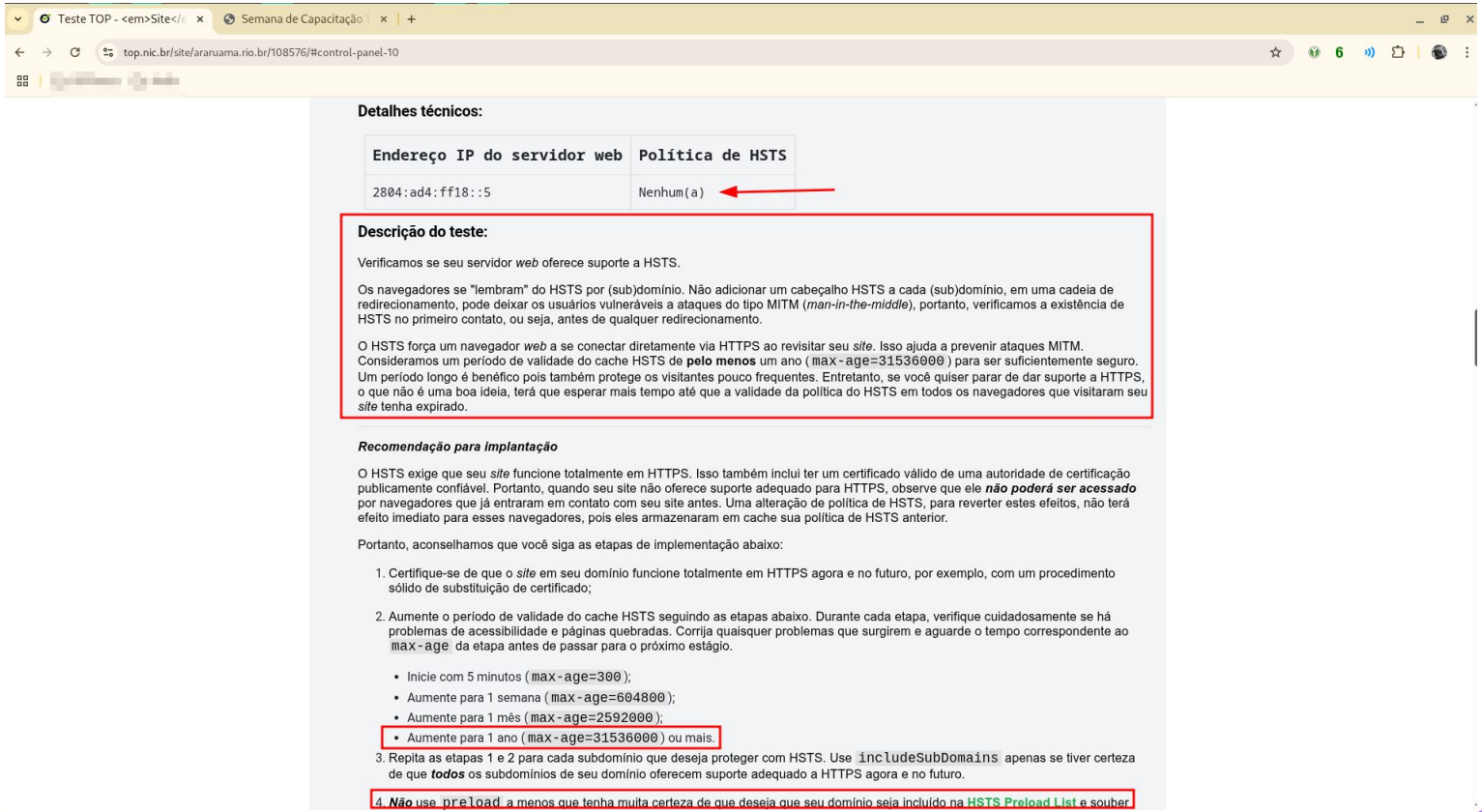
Opa! Saímos da casa dos 92% e atingimos 94%. Vamos checar o que falta agora.



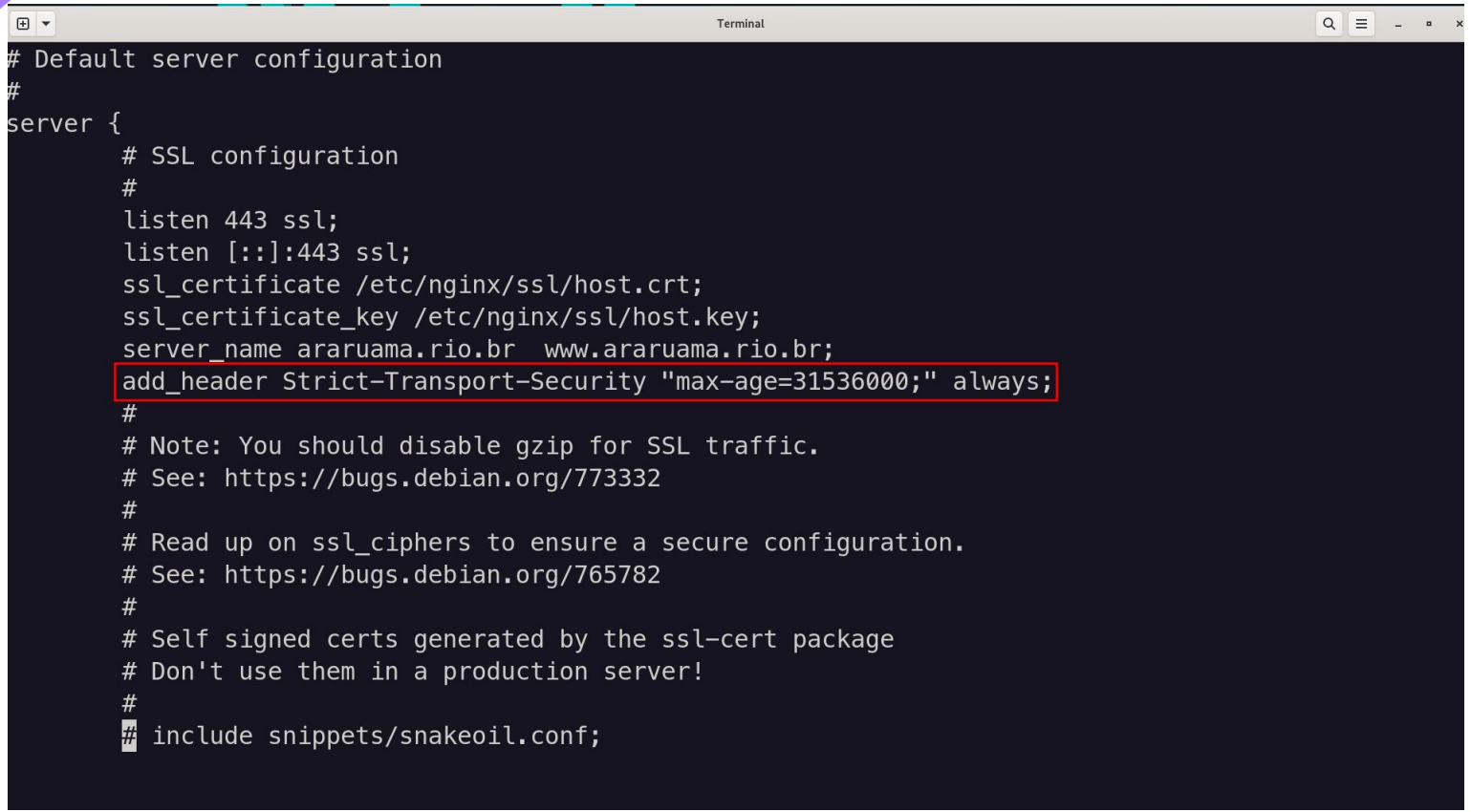
Resolvemos o redirecionamento e agora teremos que acertar o **HSTS**. Para isso precisaremos adicionar um cabeçalho na configuração do nosso site.



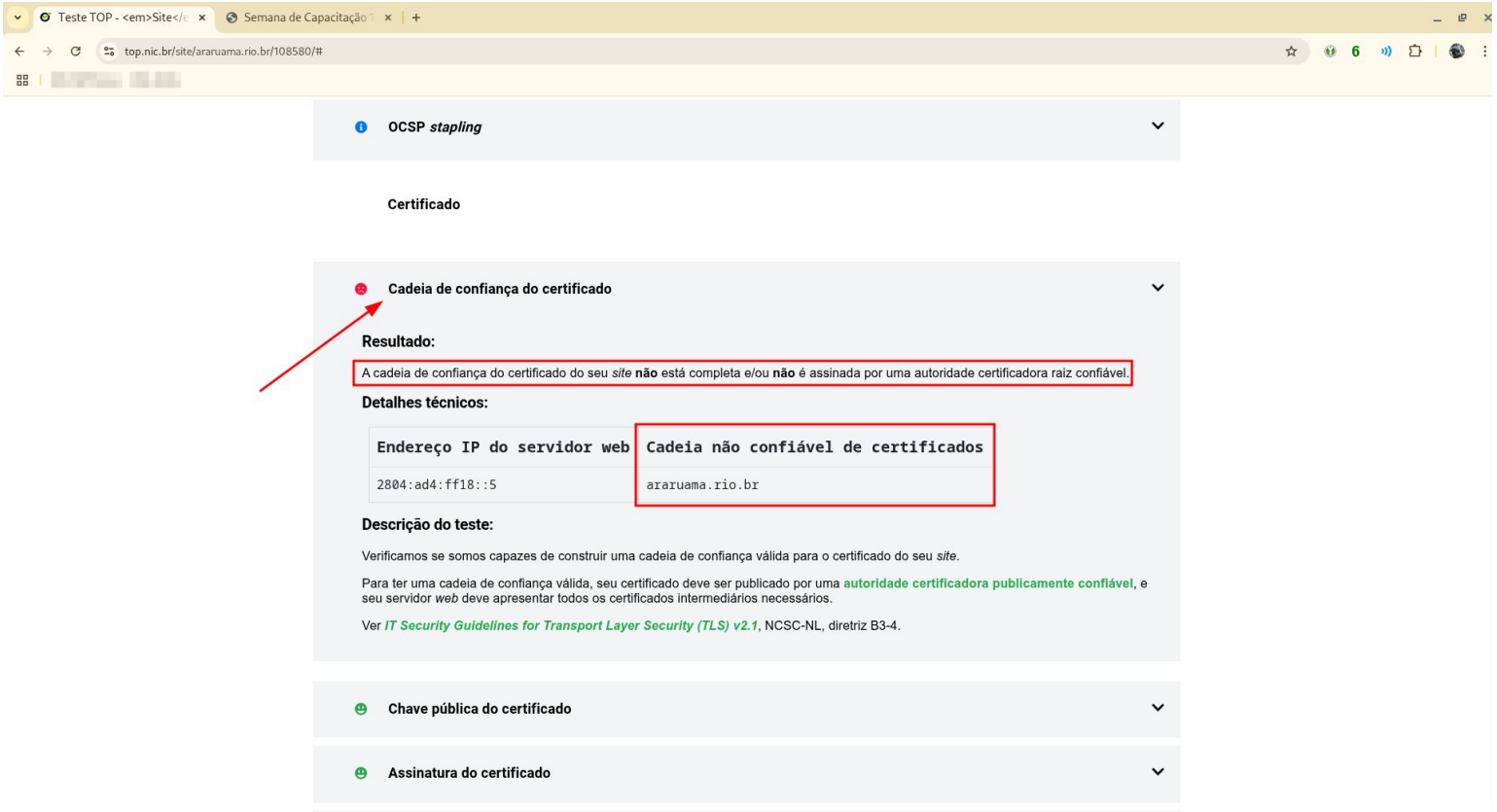
O **HSTS** serve para forçar o navegador do visitante a acessar o (sub)domínio somente se ele estiver em **HTTPS** e isso **definido por um período**, que ficará guardado no navegador até expirar. Se definido **1 ano por exemplo**, todos os navegadores que revisitarem seu site, exigirão **HTTPS** até que expire esse prazo.



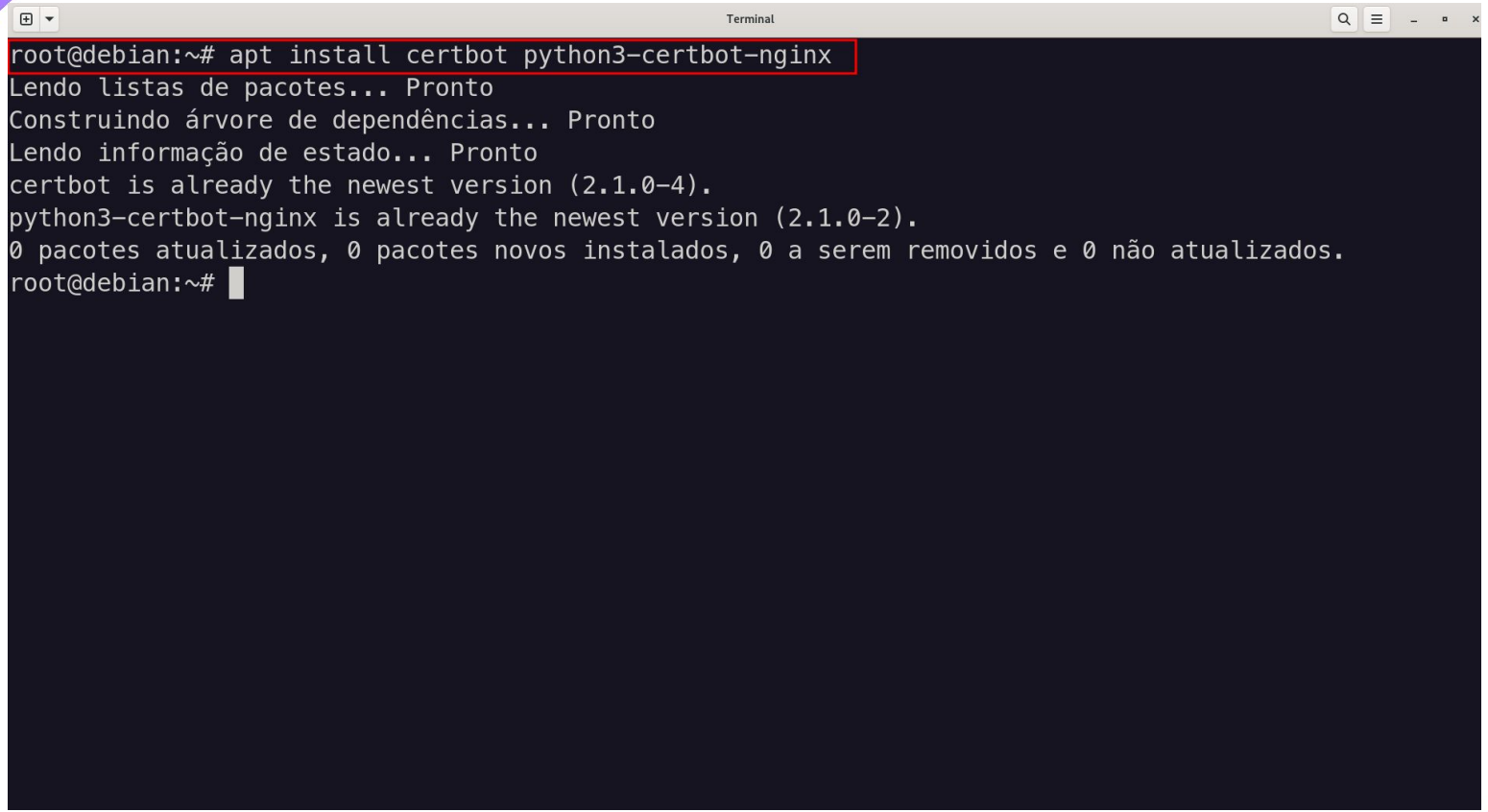
Vamos adicionar em **/etc/nginx/sites-available/default** a seguinte linha abaixo. Para passar no TOP o mínimo a ser definido é de **pelo menos 1 ano**.



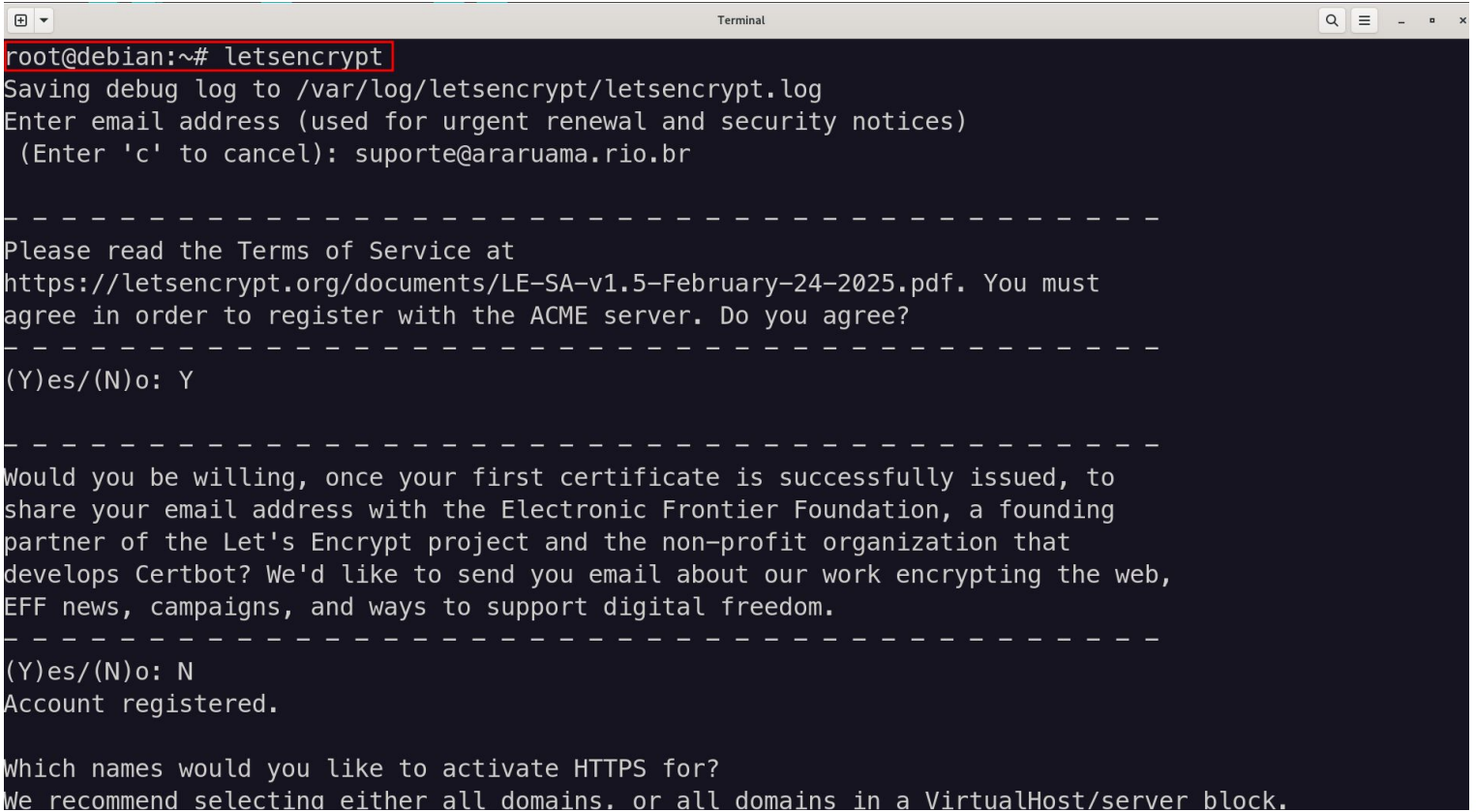
Um outro detalhe. Lembram do **certificado auto-assinado**?



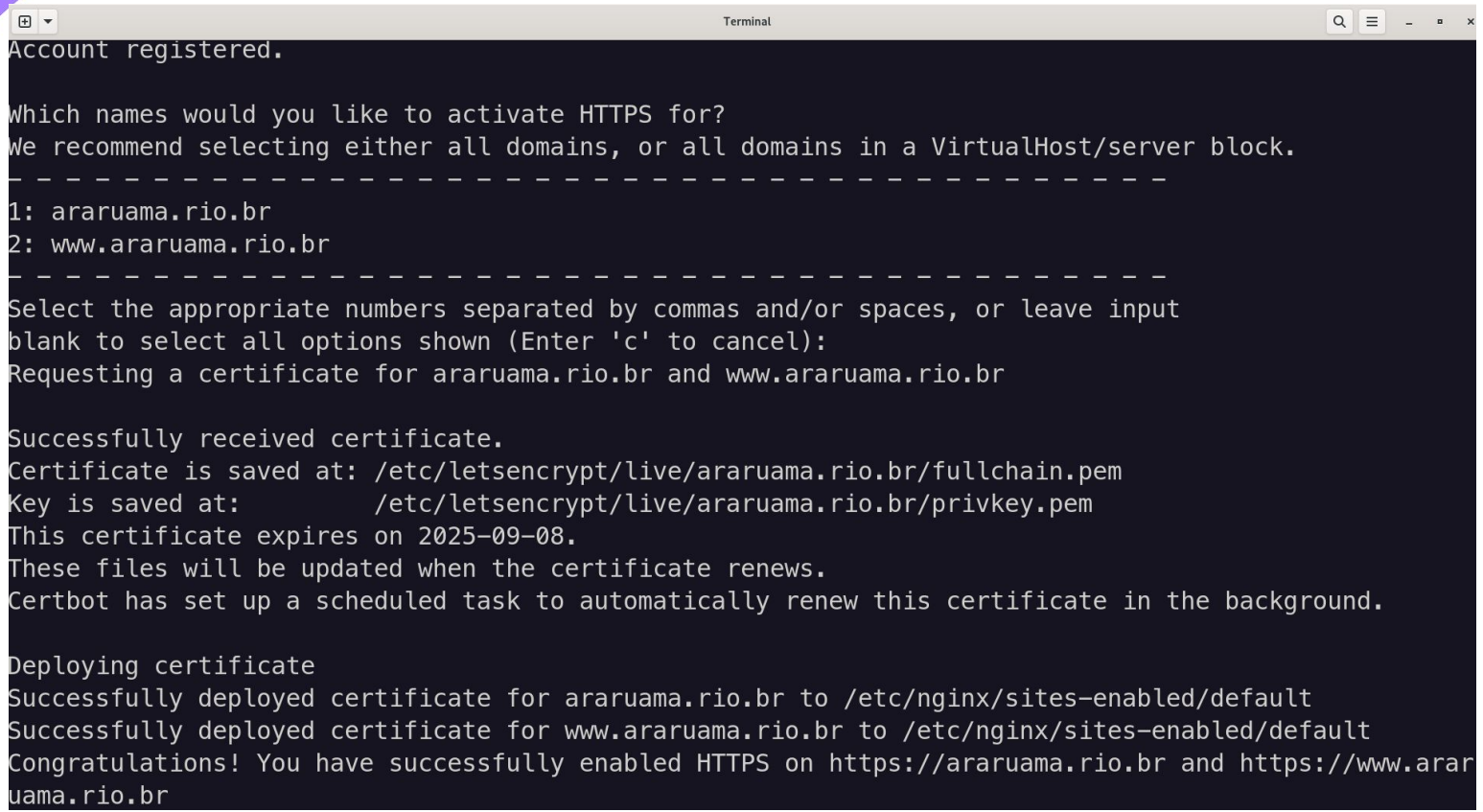
Vamos criar um certificado válido o nosso servidor com o **Let's Encrypt**. Para isso vamos instalar alguns pacotes no nosso Debian.



Executamos o comando: **letsencrypt**



Continuando…



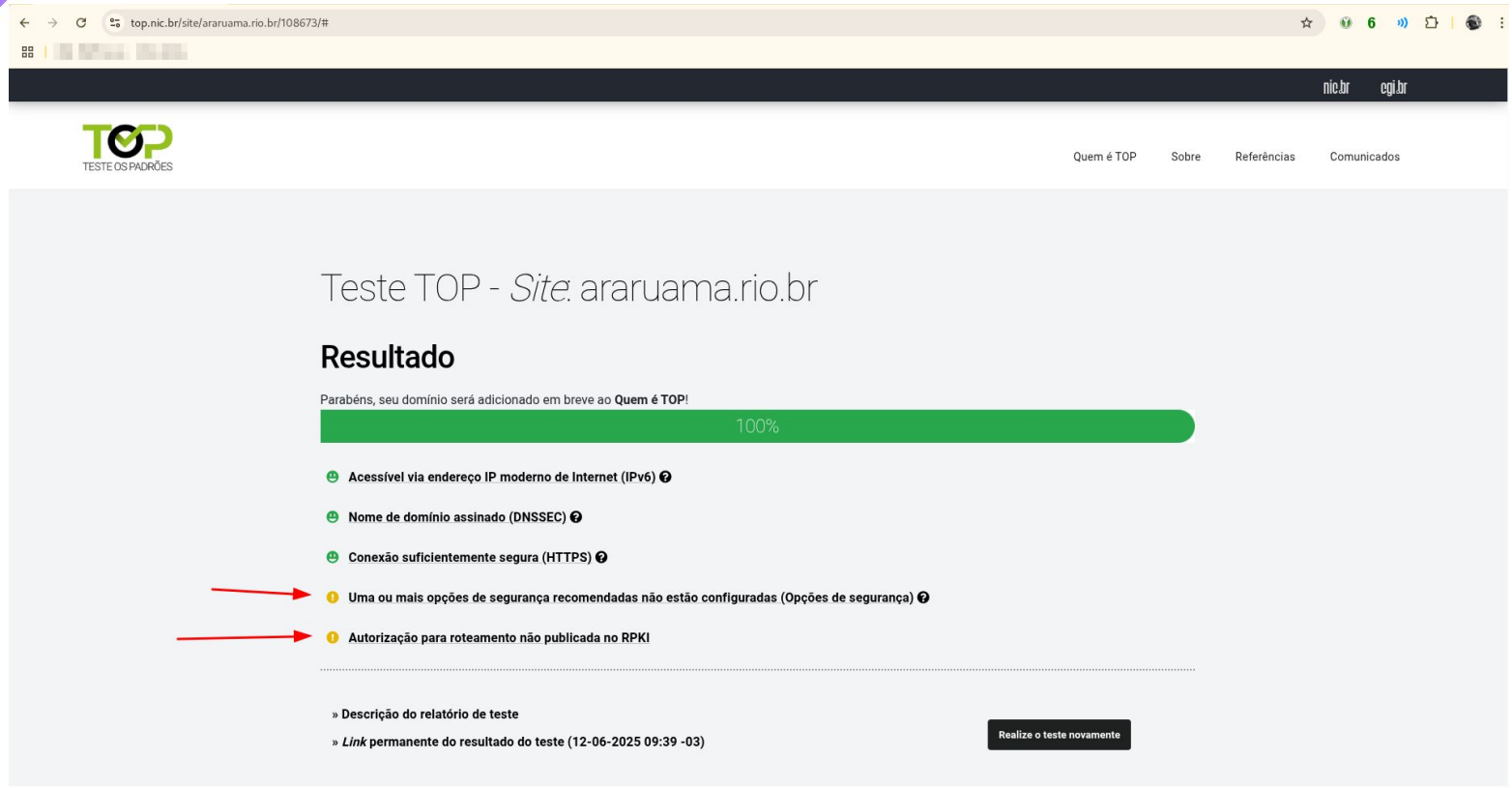
Vamos deixar nosso arquivo **/etc/nginx/sites-available/default** assim:

```
server {
        listen [::]:443 ssl; # managed by Certbot
        listen 443 ssl; # managed by Certbot

        ssl_certificate /etc/letsencrypt/live/araruama.rio.br/fullchain.pem; # managed by Certbot
        ssl_certificate_key /etc/letsencrypt/live/araruama.rio.br/privkey.pem; # managed by Certbot
        add_header Strict-Transport-Security "max-age=31536000;" always;

        include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
        ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

       root /var/www/html;

       index index.html index.htm index.nginx-debian.html;

       server_name araruama.rio.br www.araruama.rio.br;

       location / {
              try_files $uri $uri/ =404;
       }

}

server {
    if ($host = www.araruama.rio.br) {
        return 301 https://$host$request_uri;
    }
    if ($host = araruama.rio.br) {
        return 301 https://$host$request_uri;
    }

    server_name araruama.rio.br www.araruama.rio.br;
    listen 80;
    listen [::]:80;
    return 404; # managed by Certbot
}
```

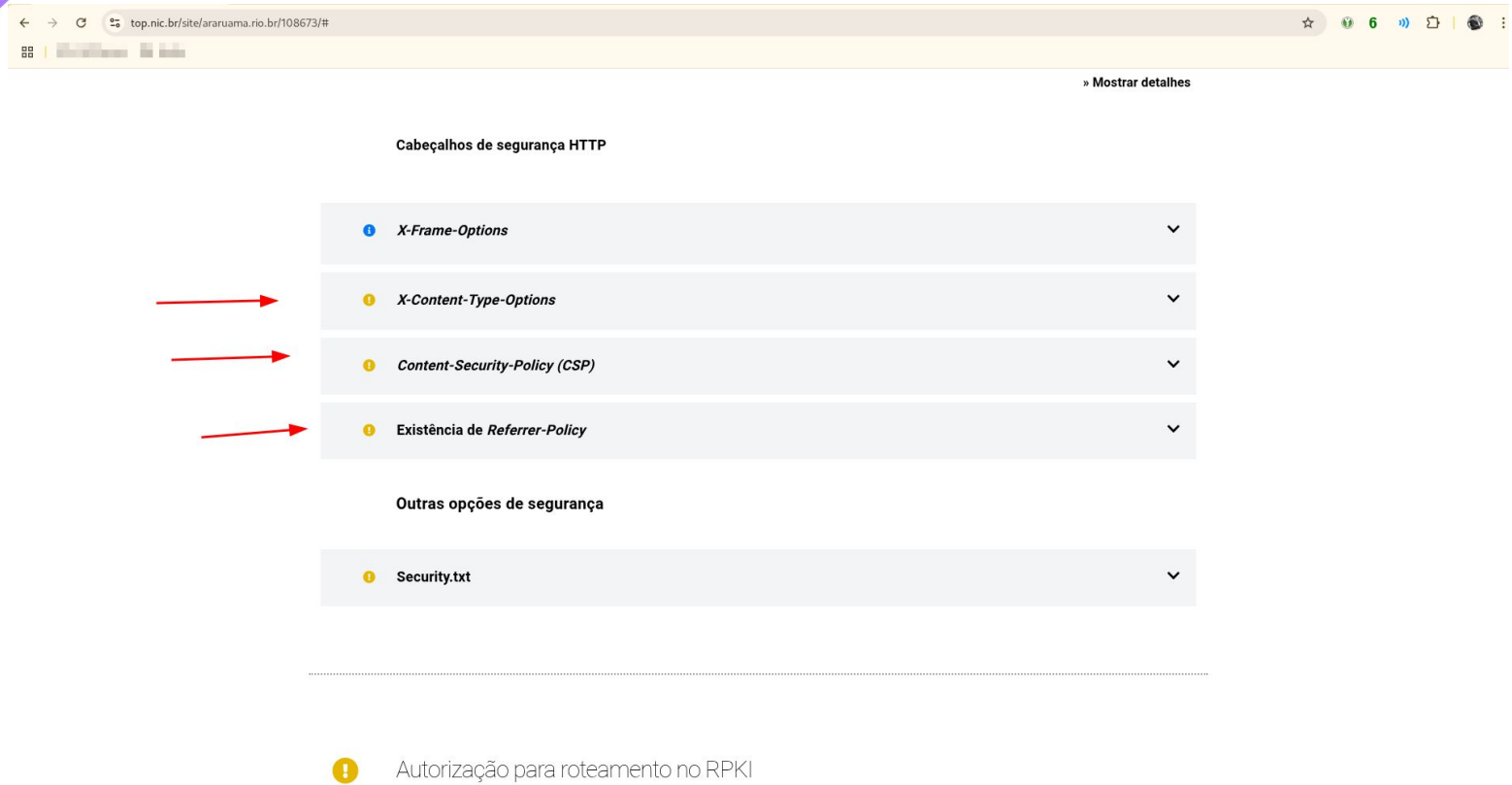
Para deixarmos o sistema atualizar o certificado digital automaticamente colocamos a seguinte linha no **/etc/crontab** executando o comando abaixo:

```
echo '00 00   1 * *   root    /usr/bin/certbot renew --post-hook="/usr/bin/systemctl
reload nginx"' >> /etc/crontab
```

Chegamos aos 100% e agora vamos ver sobre:



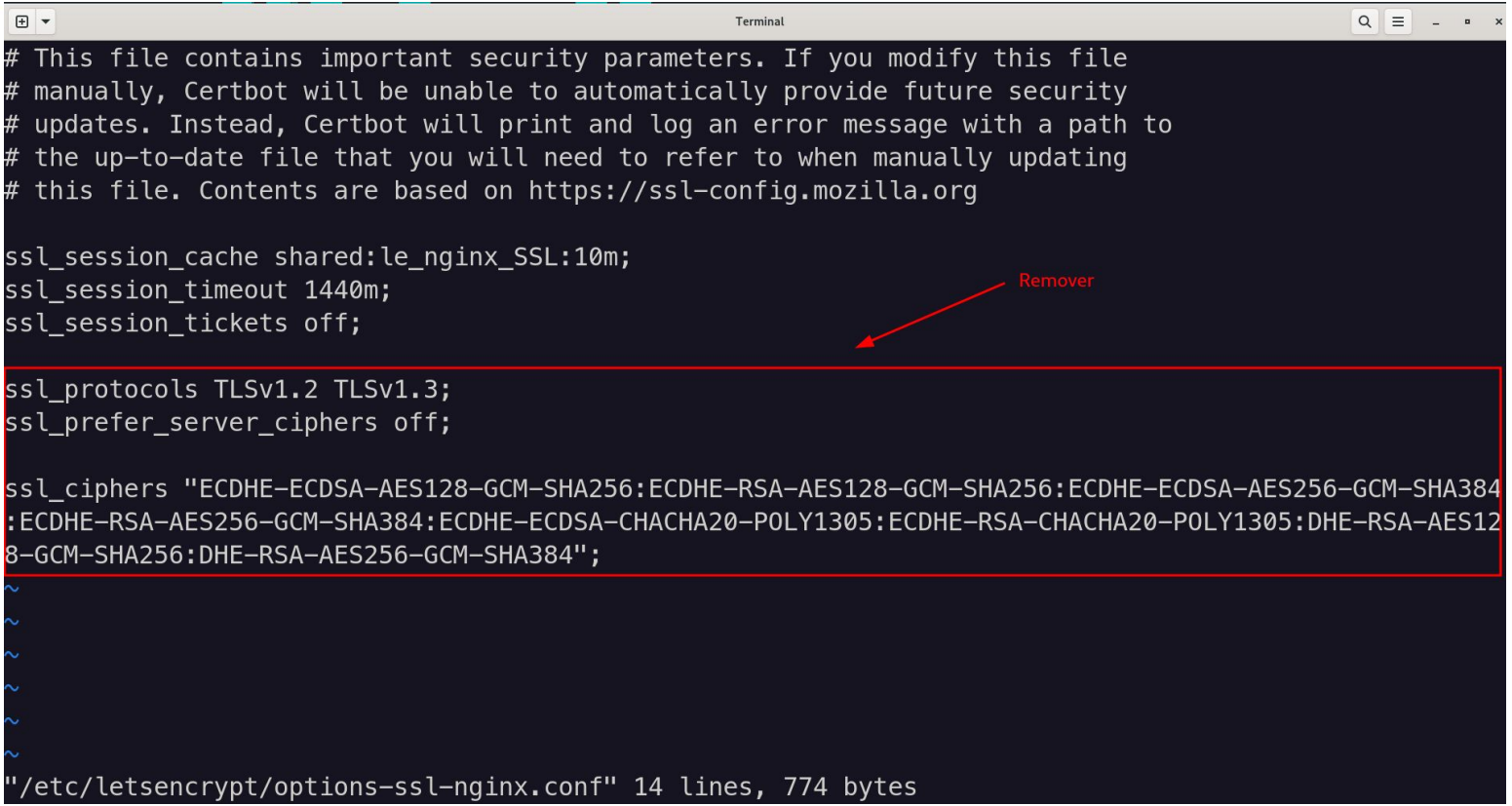
Dos 3 opcionais abaixo faremos apenas o **X-Content-Type-Options**:



Vamos deixar nosso arquivo **/etc/nginx/sites-available/default** - versão com opcionais:

```
server {
        listen [::]:443 ssl; # managed by Certbot
        listen 443 ssl; # managed by Certbot

        ssl_certificate /etc/letsencrypt/live/araruama.rio.br/fullchain.pem; # managed by Certbot
        ssl_certificate_key /etc/letsencrypt/live/araruama.rio.br/privkey.pem; # managed by Certbot
        ssl_protocols TLSv1.2 TLSv1.3;
        ssl_prefer_server_ciphers on;
        ssl_ciphers
ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-CHAC
HA20-POLY1305:ECDHE-RSA-AES128-GCM-SHA256;
       ssl_conf_command Ciphersuites TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256;
        add_header Strict-Transport-Security "max-age=31536000;" always;
        gzip off;
        add_header X-Frame-Options SAMEORIGIN;
        add_header X-Content-Type-Options nosniff;
        add_header Referrer-Policy "strict-origin-when-cross-origin";

        include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
        ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

         root /var/www/html;
         index index.html index.htm index.nginx-debian.html;
         server_name araruama.rio.br www.araruama.rio.br;

         location / {
                 try_files $uri $uri/ =404;
         }
}

server {
    if ($host = www.araruama.rio.br) {
        return 301 https://$host$request_uri;
    }
    if ($host = araruama.rio.br) {
        return 301 https://$host$request_uri;
    }

    server_name araruama.rio.br www.araruama.rio.br;
    listen 80;
    listen [::]:80;
    return 404; # managed by Certbot
}
```

Precisamos remover a configuração abaixo em **/etc/letsencrypt/options-ssl-nginx.conf** para não conflitar com a nossa:



Para o **Apache2** podemos fazer o seguinte, modifique **/etc/apache2/mods-available/ssl.conf** deixando-o assim:

```
SSLEngine on

SSLProtocol TLSv1.2 TLSv1.3
SSLCipherSuite
ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM
-SHA256:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-CHACHA20-POLY1305:ECDHE-RSA-AES128-G
CM-SHA256
SSLOpenSSLConfCmd Ciphersuites
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
SSLHonorCipherOrder on
SSLCompression off
SSLSessionTickets off

LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-agent}i\""
vhost_combined
LogFormat "%v %h %l %u %t \"%r\" %>s %b" vhost_common
```

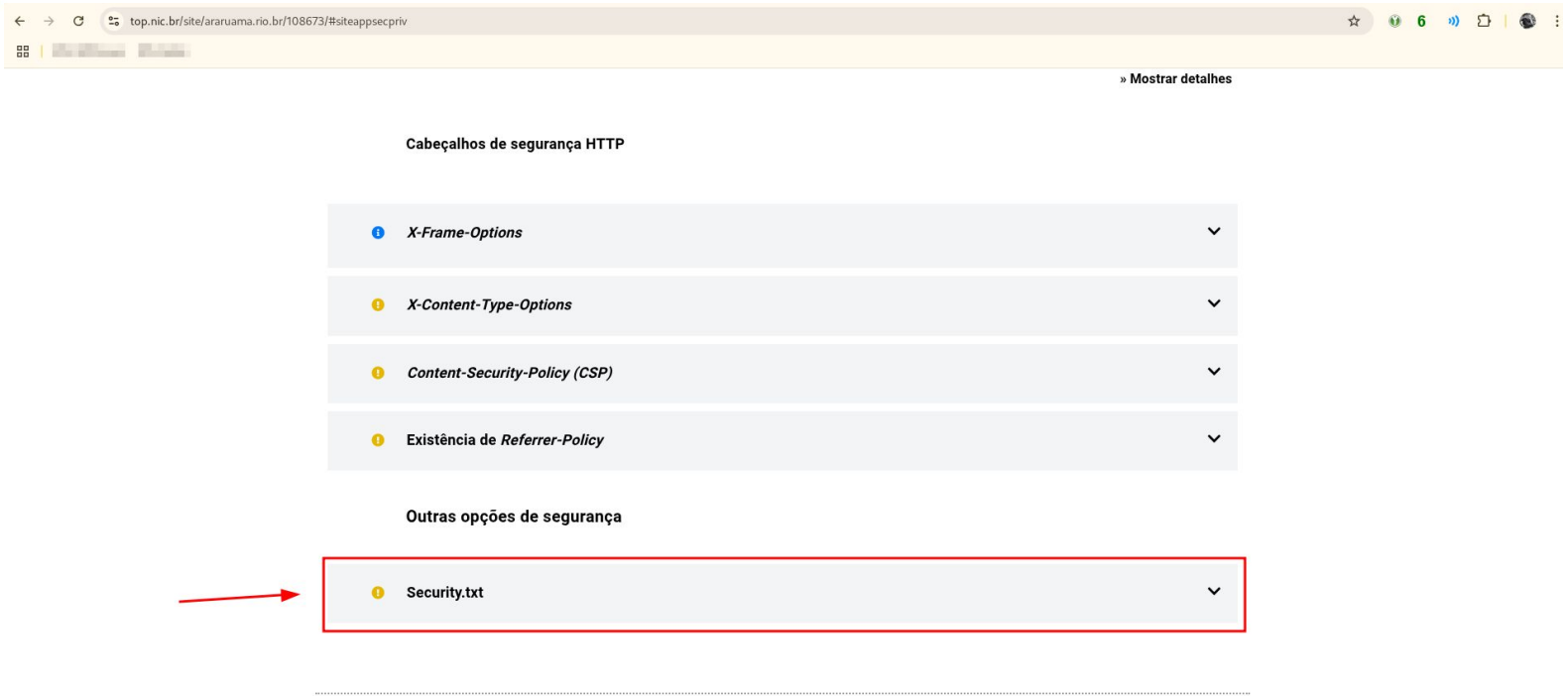
Habilite o módulo headers:

```
# a2enmod headers
# a2enmod ssl
```

Na sua configuração de virtual host adicione o cabeçalho do HSTS:

```
<VirtualHost *:443>
...
Header always set Strict-Transport-Security "max-age=31536000;"
Header always set X-Frame-Options "SAMEORIGIN"
Header always set X-Content-Type-Options "nosniff"
Header set Referrer-Policy "strict-origin-when-cross-origin"
...
</VirtualHost>
```

security.txt - **RFC 9116** (opcional mas importante):



Criando o arquivo **security.txt**:

Para assinarmos digitalmente o **security.txt** precisaremos de um par de chaves pública e privada. Se você não tiver, pode criar um instalando o **gnupg2**, por exemplo, em seu Debian GNU/Linux assim:

```
# apt install gnupg2 -y
$ gpg --full-generate-key
gpg (GnuPG) 2.2.40; Copyright (C) 2022 g10 Code GmbH
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Por favor selecione o tipo de chave desejado:
   (1) RSA and RSA (default)
   (2) DSA and Elgamal
   (3) DSA (apenas assinatura)
   (4) RSA (apenas assinatura)
  (14) Existing key from card
Opção? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (3072)
O tamanho de chave pedido é 3072 bits
Por favor especifique por quanto tempo a chave deve ser válida.
         0 = chave não expira
      <n>  = chave expira em n dias
      <n>w = chave expira em n semanas
      <n>m = chave expira em n meses
      <n>y = chave expira em n anos
A chave é valida por? (0) 0
Key does not expire at all
Is this correct? (y/N) y

GnuPG needs to construct a user ID to identify your key.

Nome completo: Marcelo Gondim da Cunha
Endereço de correio eletrónico: gondim@ispfocus.net.br
Comentário:
Você selecionou este identificador de utilizador:
    "Marcelo Gondim da Cunha <gondim@ispfocus.net.br>"

Mudar (N)ome, (C)omentário, (E)ndereço ou (O)k/(S)air? O
```

Vamos extrair nossa chave pública para copiar para o servidor:

```
$ gpg --armor --export gondim@ispfocus.net.br > pgp-key.txt
```

No servidor onde está o site faremos como exemplo:

```
# mkdir -p /var/www/html/.well-known/
```

Copie o arquivo **pgp-key.txt** para **/var/www/html/**.

Abaixo um exemplo do conteúdo do **security.txt**. Para entender melhor os parâmetros e até adicionar outros, consulte a RFC 9116.

```
Contact: mailto:gondim@ispfocus.net.br
Encryption: https://araruama.rio.br/pgp-key.txt
Preferred-Languages: pt, en
Expires: 2025-12-31T23:59:00z
Canonical: https://araruama.rio.br/.well-known/security.txt
```

Antes de usarmos nosso security.txt precisamos assinar ele com o comando:

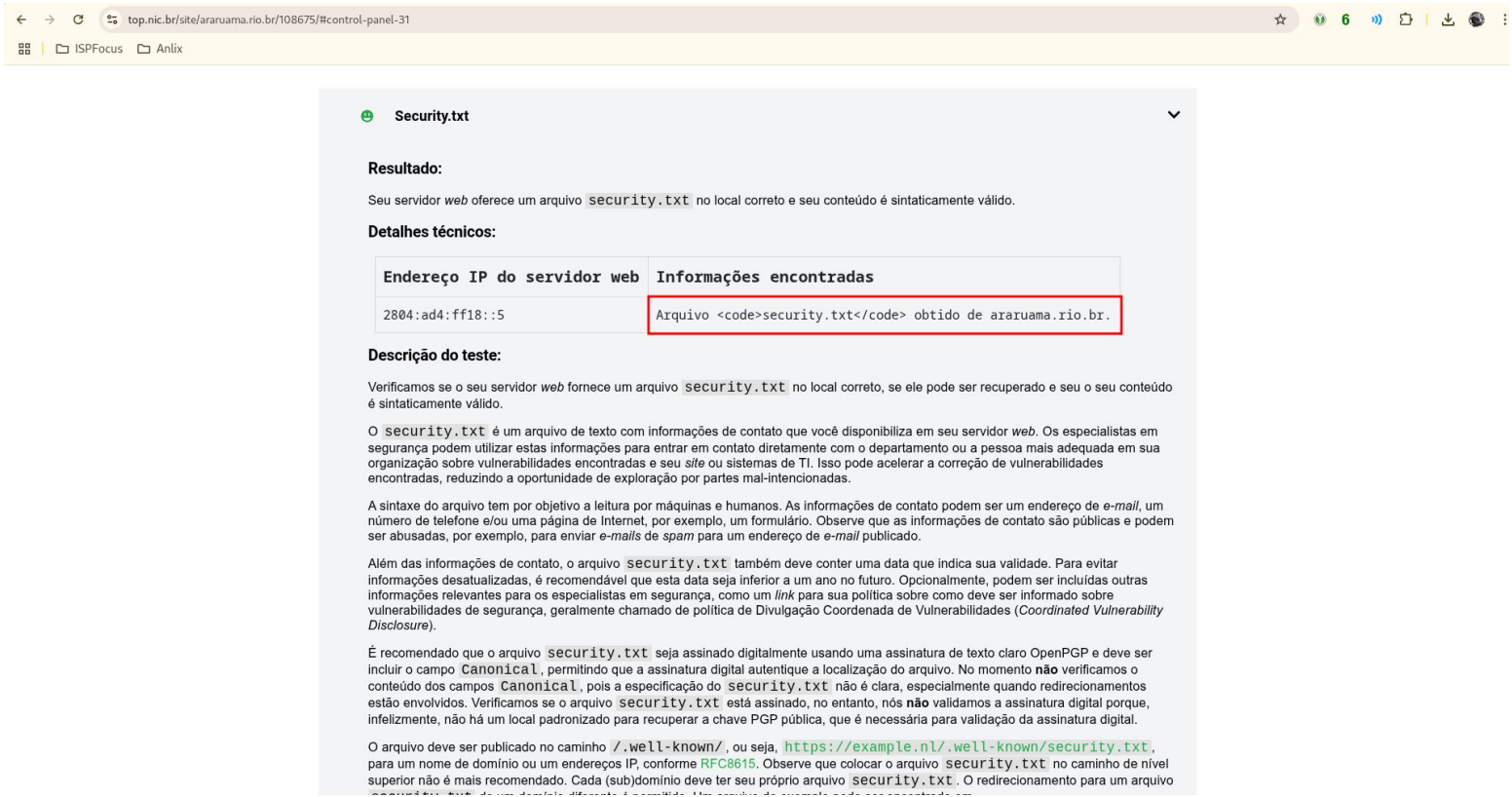
```
$ gpg --default-key gondim@ispfocus.net.br --armor --clearsign security.txt
$ mv security.txt.asc security.txt
```

Agora precisamos copiar esse arquivo **security.txt** para o servidor web do nosso lab em: **/var/www/html/.well-known/**

O conteúdo do arquivo **security.txt** assinado:

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA512

Contact: mailto:gondim@ispfocus.net.br
Encryption: https://araruama.rio.br/pgp-key.txt
Preferred-Languages: pt, en
Expires: 2025-12-31T23:59:00z
Canonical: https://araruama.rio.br/.well-known/security.txt
-----BEGIN PGP SIGNATURE-----

iQJLBAEBCgA1FiEEETLx9Tf3EoO627cgu0rIXDcmJnoFAmhKT6AXHGdvbmRpbUBp
c3Bmb2N1cy5uZXQuYnIACgkQu0rIXDcmJnq07A//X0XnvkfuhHHrs96DvBJMv3Hb
YtatxKQtSPX82srd0irFU9D9vhejlrmCk+2FzzKPdIDdCpoDj7C6PN4Db9w1MdbM
iqjswt+EnbdB6vTafbEuVFO6WicFCiVbBantyOxV4U3GcWvsSYyLkRbX0ME3OHQK
SFo7CDegSZn6aGnz8axA9YPRQldEbw9uv03ndrdRE8C8aua1wpLDLgjPbKNtO7Za
lYQ/UNwWYEQCx4meo8TWDF4++TNVLGqxV1HovJvuU4Gz/bjtC4S/7j0GqO6TyH7L
jiWBDFWnmgwLQ+cIOLdDaTNStXH/PYYvdRbJPM7CjcbiX3LwIy8yE64n9B+BL5UT
3hmLvELOEiMy1AEAqbJuHNtT6opMDNjAaKYbt2bU52axTPdYrH472OoUACtB/L+U
pGLypnA9HDwwDKjIjTowEnaoWPbEULkmkgZQy1NheGYc1kucrnAcvMNZK0Q+98SF
HA0+LkHvch2OqR9WGfESzLtKL3TaKdow6z2vqgmO5oyxwCoUoprGPIGjDtDiVAoW
4Mptvnqo/+poK7rYidB6prODXa484QmbJDMkYmTYR6jS4mB5YbOTahBfkoDgNLwT
QnNEwuyvDVDnFV8ThRQCsvq1t06i/zpCUB6Bh8mcivVXSh+pk87y2BsArgWqUDMG
cdGAN8IWNqImbqZ4Abk=
=TV8q
-----END PGP SIGNATURE-----
```

Vamos ao resultado final:



Cultura de Segurança da Informação

Algumas poucas dicas para melhorar a sua Cultura de Segurança da informação:

- Não acesse qualquer site recebido por links em e-mails ou enviados por estranhos, confira também se o certificado digital do site é válido e se o domínio está condizente com o que site legítimo.

- Não utilize a mesma senha em todos os lugares que você acessa. Crie uma senha para cada sistema. Lembre-se que se você utiliza a mesma senha em diversos lugares e esta vazar, o cibercriminoso terá acesso a todos os seus sistemas.

- Use senhas complexas e se possível de 20 caracteres pelo menos. Não é difícil, basta utilizar um Gerenciador de Senhas. Quer usar um bom e livre? KeepPassXC (https://keepassxc.org/) tem pra GNU/Linux, Windows e MacOS. Você só precisará guardar uma senha master e fazer backup do seu cofre com todas as suas senhas.

- Se disponível sempre habilite um MFA (Multi-factor authentication), isso aumenta a segurança em caso de vazamento da sua senha em algum sistema. Gere se possível e sempre guarde com segurança, seus códigos de recuperação caso perca seu MFA. Você pode guardá-los também no seu KeePassXC :)

- Somente cadastre seus dados pessoais em sistemas que realmente sejam necessários. Proteja seus dados.

- Cuidado com e-mails que contenham links para clicar ou anexos para abrir e executar. Desconfie sempre! Assim vai estar se protegendo de Ransomwares e outros malwares.

- Cuidado em salas de bate papo, não execute qualquer coisa que te enviem sem que você conheça a pessoa e confirme na procedência daquilo.

Algumas poucas dicas para melhorar a sua Cultura de Segurança da informação:

- Não use Wi-Fi público para seus acessos fora do seu ambiente de trabalho ou de casa, você pode se tornar vítima de um golpe digital. Habilite seu acesso através do seu smartphone e use a VPN para acessos corporativos.

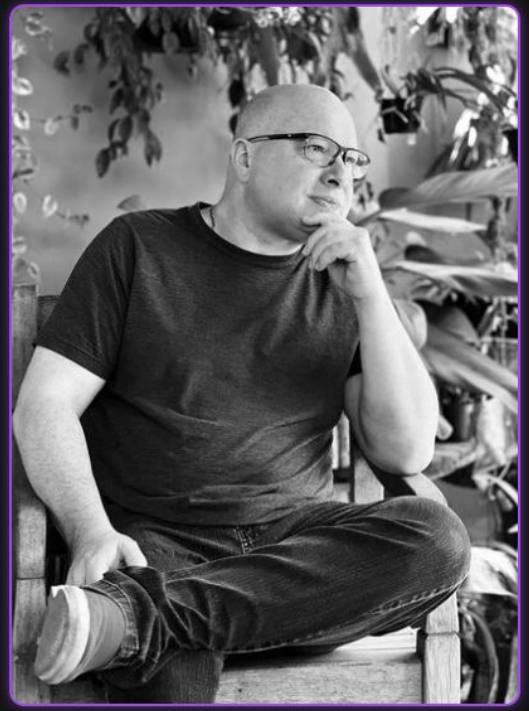
- Não instale qualquer software no seu Sistema Operacional, mantenha seu ambiente de trabalho mais seguro possível, ainda mais se costuma fazer compras e acessos a Bancos por ele. Confirme a procedência do Software.

- Se seu sistema tiver a possibilidade, mantenha um antivírus e anti-malware sempre atualizados.

- Não instale nada pirata ou crackeado no seu sistema. Não existe almoço grátis, algo ali poderá te prejudicar de alguma forma. Procure por Softwares Livres que te atenda.

- Não utilize ambientes de terceiros para se logar em suas contas, pode ser um ambiente hostil e nem deixe seu ambiente de trabalho aberto enquanto estiver ausente dele.

- O mais importante de tudo que foi comentado acima: tenha sempre um ou mais BACKUPs atualizados dos seus dados mais importantes e guarde-os com cuidado. Já dizia a frase: “Quem tem um, não tem nenhum. Quem tem dois, tem um”. Teste seus backups periodicamente para saber se estão recuperáveis. Não adianta ter um backup danificado.



Marcelo Gondim da Cunha

Especialista em redes e segurança, com experiência desde os anos 1990. Atuou como desenvolvedor, consultor de sistemas GNU/Linux e foi CTO da Nettel Telecom, onde implantou IPv6 em 2013. Contribui com o projeto MANRS. Também liderou o SOC da Brasil TecPar entre 2022 e 2025, focando em defesas contra DDoS, boas práticas e onde desenvolveu uma rede de DNS(s) Recursivos Anycast certificada pelo KINDNS.

- Administração de Sistemas Unix-Like desde 1996.
- Consultor na Conectiva S/A - Unidade Rio em 2000.
- Direção do AS53135 - Nettel Telecomunicações entre 2003 e 2021 atingindo a marca de 41.000 assinantes.
- Diversas palestras em eventos da área de Redes e Serviços e artigos técnicos publicados.