

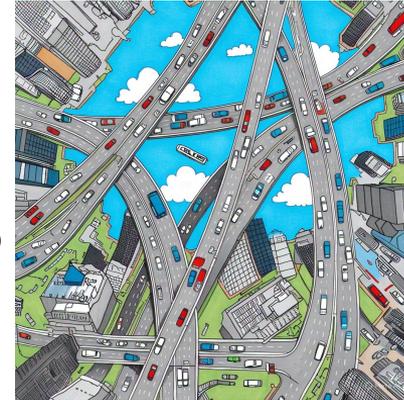
Teste para padrões técnicos modernos de Internet e segurança: IPv6, DNSSEC, TLS, HTTPS e HSTS

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.

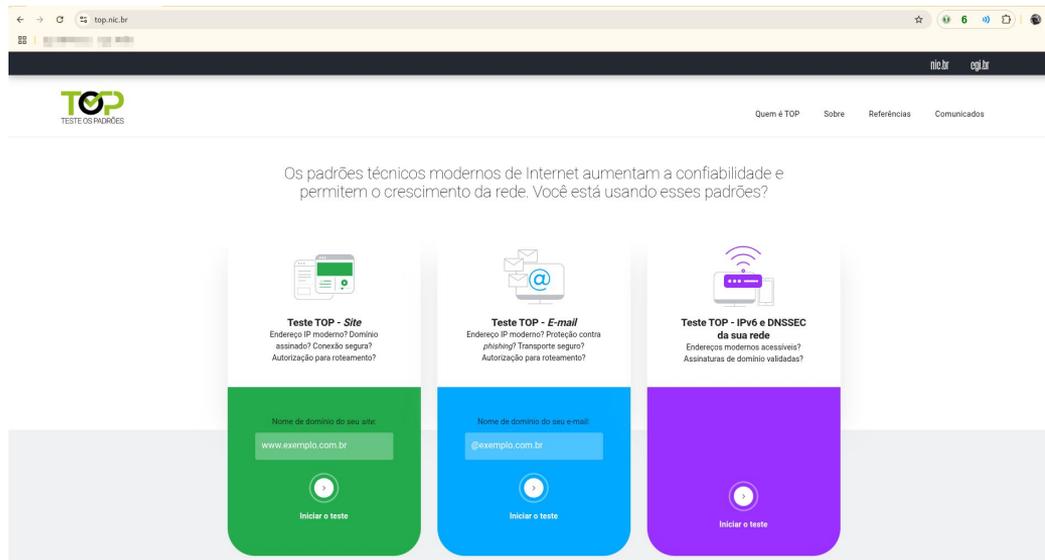


- Se você pensou: **BGP (Border Gateway Protocol)** e **DNS (Domain Name System)**. Acertou!
- O BGP é o protocolo que interliga todos os **Sistemas Autônomos** no mundo, formando assim a **Internet**.
- DNS faz a tradução de nomes em IPs e vice e versa. Bem, isso era suficiente no passado, hoje ele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

```
Terminal
[ root@shadow ]-[ ~ ]-[ 13:04:02 ]
# ssh root@araruama.rio.br
Enter passphrase for key '/root/.ssh/id_ed25519':
Linux debian 6.1.0-9-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.27-1 (2023-05-08) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Jun 10 13:03:32 2025 from [REDACTED]
root@debian:~#
```

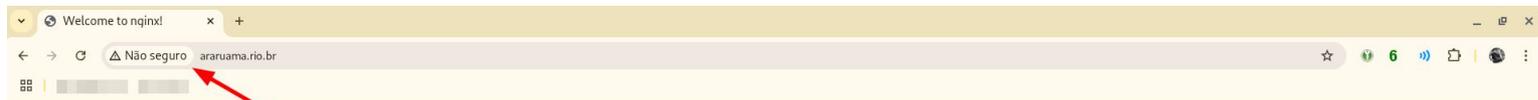
Instalando o Nginx que será nosso primeiro servidor Web de testes:

```
Terminal
root@debian:~# apt install nginx
Lendo listas de pacotes... Pronto
Construindo árvore de dependências... Pronto
Lendo informação de estado... Pronto
nginx is already the newest version (1.22.1-9+deb12u2).
0 pacotes atualizados, 0 pacotes novos instalados, 0 a serem removidos e 0 não atualizados.
root@debian:~# █
```

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

```
Terminal
78006 ?      I      0:00  \_ [kworker/u2:1-ext4-rsv-conversion]
  1 ?      Ss     0:08  /lib/systemd/systemd --system --deserialize=34
 459 ?      Ss     0:02  /usr/sbin/cron -f
 460 ?      Ss     0:00  /usr/bin/dbus-daemon --system --address=systemd: --nofork --nopidfile -
 462 ?      Ss     0:02  /lib/systemd/systemd-logind
 505 tty1    Ss     0:00  /bin/login -p --
 585 tty1    S      0:00  \_ -bash
 588 tty1    S      0:00  \_ su -
 589 tty1    S+     0:00  \_ -bash
 578 ?      Ss     0:00  /lib/systemd/systemd --user
 579 ?      S      0:00  \_ (sd-pam)
34785 ?      Ss     0:00  sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups
77653 ?      Ss     0:00  \_ sshd: root@pts/0
77668 pts/0     Ss     0:00  \_ -bash
78061 pts/0     R+     0:00  \_ ps afx
62800 ?      Ss     0:00  /lib/systemd/systemd-journald
62907 ?      Ssl    0:00  /lib/systemd/systemd-timesyncd
62938 ?      Ss     0:00  /lib/systemd/systemd-udev
77657 ?      Ss     0:00  /lib/systemd/systemd --user
77658 ?      S      0:00  \_ (sd-pam)
78050 ?      S      0:00  nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
78051 ?      S      0:00  \_ nginx: worker process
root@debian:~#
```

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



Observe

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

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:

The screenshot shows a web browser at the URL `top.nic.br`. The page features the TOP logo and navigation links. The main content area contains three test cards: 'Teste TOP - Site' (green), 'Teste TOP - E-mail' (blue), and 'Teste TOP - IPv6 e DNSSEC da sua rede' (purple). Red arrows and circles highlight the browser address bar (1), the domain input field (2), and the 'Iniciar o teste' button (3).

Os padrões técnicos modernos de Internet aumentam a confiabilidade e permitem o crescimento da rede. Você está usando esses padrões?

Teste TOP - Site
Endereço IP moderno? Domínio assinado? Conexão segura? Autorização para roteamento?

Nome de domínio do seu site:

Iniciar o teste

Teste TOP - E-mail
Endereço IP moderno? Proteção contra phishing? Transporte seguro? Autorização para roteamento?

Nome de domínio do seu e-mail:

Iniciar o teste

Teste TOP - IPv6 e DNSSEC da sua rede
Endereços modernos acessíveis? Assinaturas de domínio validadas?

Iniciar o teste

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**.

Teste TOP - Site

top.nic.br/site/araruama.rio.br/108567/

nic.br egi.br

Quem é TOP Sobre Referências Comunicados

Teste TOP - Site: araruama.rio.br

Resultado

66%

- ✓ Acessível via endereço IP moderno de Internet (IPv6)
- ✓ Nome de domínio assinado (DNSSEC)
- ✗ Servidor web inacessível (HTTPS)
- ! Uma ou mais opções de segurança recomendadas não estão configuradas (Opções de segurança)
- ! Autorização para roteamento não publicada no RPKI

» Descrição do relatório de teste

» Link permanente do resultado do teste (10-06-2025 13:48 -03)

» Segundos até a opção de reteste: 158

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

Este símbolo no Google Chrome garante que o site possui um certificado digital válido.

Vamos criar agora um certificado digital auto-assinado:

```
Terminal
root@debian:~# mkdir -p /etc/nginx/ssl
root@debian:~# cd /etc/nginx/ssl/
root@debian:/etc/nginx/ssl# openssl genrsa -out host.key 2048
root@debian:/etc/nginx/ssl# openssl req -new -key host.key -out host.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:BR
State or Province Name (full name) [Some-State]:Rio de Janeiro
Locality Name (eg, city) []:Rio de Janeiro
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Araruama City
Organizational Unit Name (eg, section) []:TI
Common Name (e.g. server FQDN or YOUR name) []:araruama.rio.br
Email Address []:suporte@araruama.rio.br

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
```

Vamos criar agora um certificado digital auto-assinado:

```
Terminal
root@debian:/etc/nginx/ssl# openssl x509 -days 3650 -req -in host.csr -signkey host.key -out host.crt
Certificate request self-signature ok
subject=C = BR, ST = Rio de Janeiro, L = Rio de Janeiro, O = Araruama City, OU = TI, CN = araruama.r
io.br, emailAddress = suporte@araruama.rio.br
root@debian:/etc/nginx/ssl# ls -l
total 12
-rw-r--r-- 1 root root 1391 jun 10 14:25 host.crt
-rw-r--r-- 1 root root 1090 jun 10 14:24 host.csr
-rw----- 1 root root 1704 jun 10 14:11 host.key
root@debian:/etc/nginx/ssl#
```

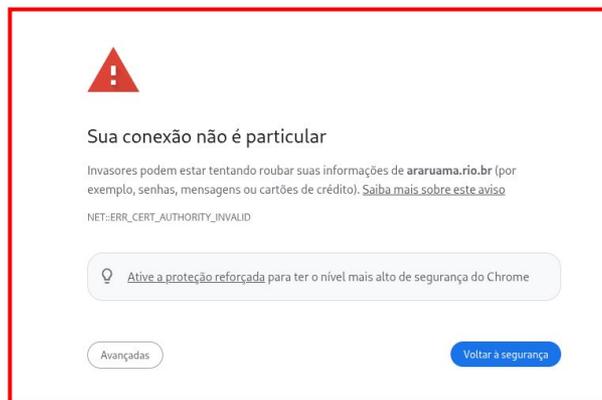
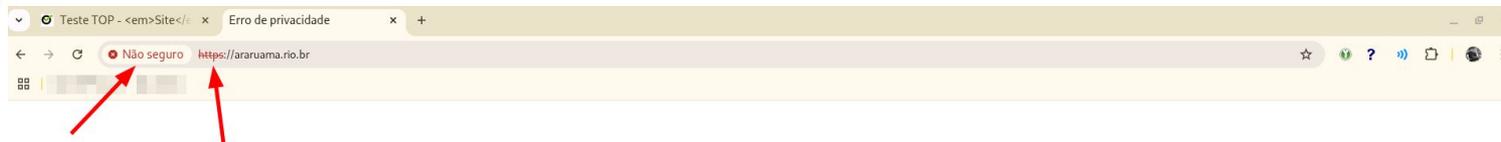
Agora que temos o certificado auto-assinado, vamos configurar o Nginx editando o arquivo `/etc/nginx/sites-available/default`:

```
Terminal
# Default server configuration
#
server {
    listen 80;
    listen [::]:80;

    # SSL configuration
    #
    listen 443 ssl;
    listen [::]:443 ssl;
    ssl_certificate /etc/nginx/ssl/host.crt;
    ssl_certificate_key /etc/nginx/ssl/host.key;
    server_name araruama.rio.br www.araruama.rio.br;

    #
    # Note: You should disable gzip for SSL traffic.
    # See: https://bugs.debian.org/773332
    #
    # Read up on ssl_ciphers to ensure a secure configuration.
    # See: https://bugs.debian.org/765782
    #
    # Self signed certs generated by the ssl-cert package
    ## Don't use them in a production server!
```

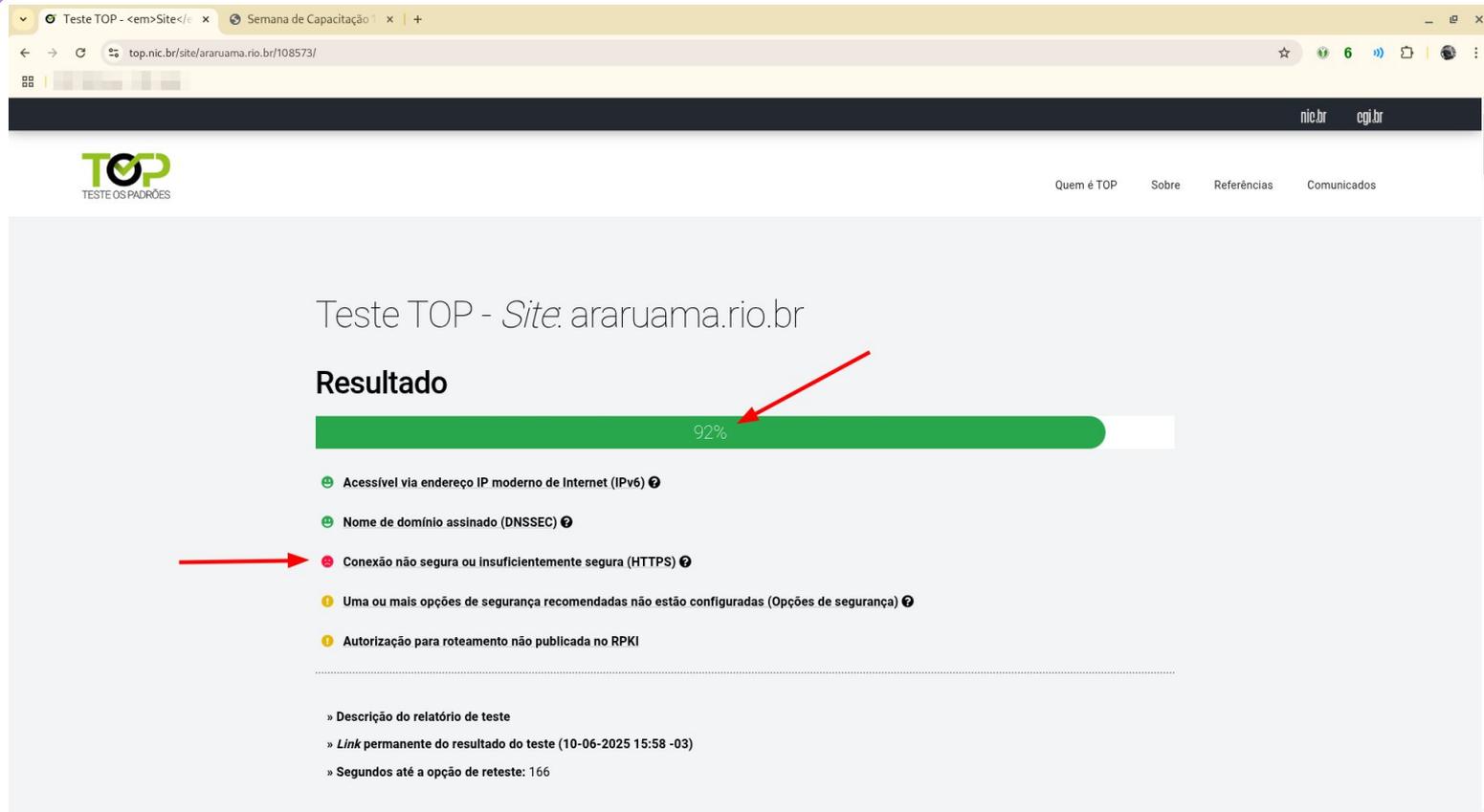
Após fazer um restart no nginx vamos acessar <https://araruama.rio.br>:



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**.



Teste TOP - *Site: araruama.rio.br*

Resultado

92%

- ✓ Acessível via endereço IP moderno de Internet (IPv6) ⓘ
- ✓ Nome de domínio assinado (DNSSEC) ⓘ
- ✗ Conexão não segura ou insuficientemente segura (HTTPS) ⓘ
- ⚠ Uma ou mais opções de segurança recomendadas não estão configuradas (Opções de segurança) ⓘ
- ⚠ Autorização para roteamento não publicada no RPKI

» Descrição do relatório de teste

» [Link permanente do resultado do teste \(10-06-2025 15:58 -03\)](#)

» Segundos até a opção de reteste: 166

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

Teste TOP - Site x Semana de Capacitação x +

top.nic.br/site/araruama.rio.br/108573/#sitetls

Conexão segura (HTTPS)

Que pena! A conexão com o seu site **não** é segura ou é **insuficientemente** segura (HTTPS), portanto, as informações em trânsito entre o seu site e seus visitantes **não** estão suficientemente protegidas contra interceptação e adulteração. Solicite ao seu provedor de hospedagem para ativar o HTTPS e configurá-lo de maneira segura.

[» Mostrar detalhes](#)

HTTP

- HTTPS disponível
- Redirecionamento para HTTPS
- Compressão HTTP
- HSTS

TLS

- Versão de TLS
- Cifras (Seleções de algoritmos)

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

The screenshot shows a web browser window with the address bar displaying 'top.nic.br/site/araruama.rio.br/108573/#control-panel-8'. The main content area is a configuration panel for 'Redirecionamento para HTTPS'. It includes a 'Resultado' section stating that the server does not offer automatic redirection, a table for technical details, and a 'Descrição do teste' section with instructions and examples.

Endereço IP do servidor web	Redirecionamento para HTTPS
2804:ad4:ff18::5	não

Descrição do teste:

Verificamos se seu servidor *web* redireciona automaticamente os visitantes de HTTP para HTTPS no mesmo domínio, por meio de um código de status de redirecionamento 3xx como 301 e 302, ou se ele oferece suporte apenas para HTTPS e não para HTTP.

Em caso de redirecionamento, um domínio deve primeiro atualizar-se, redirecionando para sua versão HTTPS antes de redirecionar para outro domínio. Isso também garante que a política do HSTS será aceita pelo navegador *web*. Exemplos de ordem correta de redirecionamento:

- `http://exemplo.br ⇒ https://exemplo.br ⇒ https://www.exemplo.br`
- `http://www.exemplo.br ⇒ https://www.exemplo.br`

Observe que este subteste avalia apenas se o domínio fornecido redireciona corretamente de HTTP para HTTPS. Um eventual redirecionamento adicional para um domínio diferente, incluindo um subdomínio do domínio em teste, não é considerado. Você pode iniciar um teste separado para o domínio para o qual está sendo redirecionado.

Ver [Protecting Your Website With Always On SSL](#), Internet Society.

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

```
Terminal
# Default server configuration
#
server {
    listen 80;
    listen [::]:80;

    # SSL configuration
    #
    listen 443 ssl;
    listen [::]:443 ssl;
    ssl_certificate /etc/nginx/ssl/host.crt;
    ssl_certificate_key /etc/nginx/ssl/host.key;
    server_name araruama.rio.br www.araruama.rio.br;
    #
    # Note: You should disable gzip for SSL traffic.
    # See: https://bugs.debian.org/773332
    #
    # Read up on ssl_ciphers to ensure a secure configuration.
    # See: https://bugs.debian.org/765782
    #
    # Self signed certs generated by the ssl-cert package
    # Don't use them in a production server!
```

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.

Teste TOP - *Site:* araruama.rio.br

Resultado

94%

- Acessível via endereço IP moderno de Internet (IPv6)
- Nome de domínio assinado (DNSSEC)
- Conexão não segura ou insuficientemente segura (HTTPS)
- Uma ou mais opções de segurança recomendadas não estão configuradas (Opções de segurança)
- Autorização para roteamento não publicada no RPKI

» Descrição do relatório de teste

» [Link permanente do resultado do teste \(10-06-2025 16:42 -03\)](#)

» Segundos até a opção de reteste: 192

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

The screenshot shows a browser window with the address bar displaying 'top.nic.br/site/araruama.rio.br/108576/#sitels'. A security warning is displayed, titled 'Conexão segura (HTTPS)', with a red warning icon. The message reads: 'Que pena! A conexão com o seu site não é segura ou é insuficientemente segura (HTTPS), portanto, as informações em trânsito entre o seu site e seus visitantes não estão suficientemente protegidas contra interceptação e adulteração. Solicite ao seu provedor de hospedagem para ativar o HTTPS e configurá-lo de maneira segura.' Below the message is a link '» Mostrar detalhes'. Under the heading 'HTTP', there is a list of settings: 'HTTPS disponível' (green checkmark), 'Redirecionamento para HTTPS' (green checkmark), 'Compressão HTTP' (blue information icon), and 'HSTS' (red warning icon). A red arrow points to the 'HSTS' setting. Under the heading 'TLS', there are two settings: 'Versão de TLS' (green checkmark) and 'Cifras (Seleções de algoritmos)' (yellow warning icon).

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.

Detalhes técnicos:

Endereço IP do servidor web	Política de HSTS
2804:ad4:ff18::5	Nenhum(a)

Descrição do teste:

Verificamos se seu servidor *web* oferece suporte a HSTS.

Os navegadores se "lembram" do HSTS por (sub)domínio. Não adicionar um cabeçalho HSTS a cada (sub)domínio, em uma cadeia de redirecionamento, pode deixar os usuários vulneráveis a ataques do tipo MITM (*man-in-the-middle*), portanto, verificamos a existência de HSTS no primeiro contato, ou seja, antes de qualquer redirecionamento.

O HSTS força um navegador *web* a se conectar diretamente via HTTPS ao visitar seu *site*. Isso ajuda a prevenir ataques MITM. Consideramos um período de validade do cache HSTS de **pelo menos um ano** (`max-age=31536000`) para ser suficientemente seguro. Um período longo é benéfico pois também protege os visitantes pouco frequentes. Entretanto, se você quiser parar de dar suporte a HTTPS, o que não é uma boa ideia, terá que esperar mais tempo até que a validade da política do HSTS em todos os navegadores que visitaram seu *site* tenha expirado.

Recomendação para implantação

O HSTS exige que seu *site* funcione totalmente em HTTPS. Isso também inclui ter um certificado válido de uma autoridade de certificação publicamente confiável. Portanto, quando seu site não oferece suporte adequado para HTTPS, observe que ele **não poderá ser acessado** por navegadores que já entraram em contato com seu site antes. Uma alteração de política de HSTS, para reverter estes efeitos, não terá efeito imediato para esses navegadores, pois eles armazenaram em cache sua política de HSTS anterior.

Portanto, aconselhamos que você siga as etapas de implementação abaixo:

1. Certifique-se de que o *site* em seu domínio funcione totalmente em HTTPS agora e no futuro, por exemplo, com um procedimento sólido de substituição de certificado;
2. Aumente o período de validade do cache HSTS seguindo as etapas abaixo. Durante cada etapa, verifique cuidadosamente se há problemas de acessibilidade e páginas quebradas. Corrija quaisquer problemas que surgirem e aguarde o tempo correspondente ao `max-age` da etapa antes de passar para o próximo estágio.
 - Inicie com 5 minutos (`max-age=300`);
 - Aumente para 1 semana (`max-age=604800`);
 - Aumente para 1 mês (`max-age=2592000`);
 - Aumente para 1 ano (`max-age=31536000`) ou mais.
3. Repita as etapas 1 e 2 para cada subdomínio que deseja proteger com HSTS. Use `includeSubDomains` apenas se tiver certeza de que **todos** os subdomínios de seu domínio oferecem suporte adequado a HTTPS agora e no futuro.
4. **Não use `preload` a menos que tenha muita certeza de que deseja que seu domínio seja incluído na `HSTS Preload List` e souber**

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

```
Terminal
# Default server configuration
#
server {
    # SSL configuration
    #
    listen 443 ssl;
    listen [::]:443 ssl;
    ssl_certificate /etc/nginx/ssl/host.crt;
    ssl_certificate_key /etc/nginx/ssl/host.key;
    server_name araruama.rio.br www.araruama.rio.br;
    add_header Strict-Transport-Security "max-age=31536000;" always;
    #
    # Note: You should disable gzip for SSL traffic.
    # See: https://bugs.debian.org/773332
    #
    # Read up on ssl_ciphers to ensure a secure configuration.
    # See: https://bugs.debian.org/765782
    #
    # Self signed certs generated by the ssl-cert package
    # Don't use them in a production server!
    #
    # include snippets/snakeoil.conf;
```

Um outro detalhe. Lembram do certificado auto-assinado?

OCSP stapling

Certificado

Cadeia de confiança do certificado

Resultado:

A cadeia de confiança do certificado do seu site não está completa e/ou não é assinada por uma autoridade certificadora raiz confiável.

Detalhes técnicos:

Endereço IP do servidor web	Cadeia não confiável de certificados
2804:ad4:ff18::5	araruama.rio.br

Descrição do teste:

Verificamos se somos capazes de construir uma cadeia de confiança válida para o certificado do seu site.

Para ter uma cadeia de confiança válida, seu certificado deve ser publicado por uma **autoridade certificadora publicamente confiável**, e seu servidor web deve apresentar todos os certificados intermediários necessários.

Ver [IT Security Guidelines for Transport Layer Security \(TLS\) v2.1](#), NCSC-NL, diretriz B3-4.

Chave pública do certificado

Assinatura do certificado

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

```
Terminal
root@debian:~# apt install certbot python3-certbot-nginx
Lendo listas de pacotes... Pronto
Construindo árvore de dependências... Pronto
Lendo informação de estado... Pronto
certbot is already the newest version (2.1.0-4).
python3-certbot-nginx is already the newest version (2.1.0-2).
0 pacotes atualizados, 0 pacotes novos instalados, 0 a serem removidos e 0 não atualizados.
root@debian:~#
```

Executamos o comando: **letsencrypt**

```
Terminal
root@debian:~# letsencrypt
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Enter email address (used for urgent renewal and security notices)
(Enter 'c' to cancel): suporte@araruama.rio.br

-----
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.5-February-24-2025.pdf. You must
agree in order to register with the ACME server. Do you agree?
-----
(Y)es/(N)o: Y

-----
Would you be willing, once your first certificate is successfully issued, to
share your email address with the Electronic Frontier Foundation, a founding
partner of the Let's Encrypt project and the non-profit organization that
develops Certbot? We'd like to send you email about our work encrypting the web,
EFF news, campaigns, and ways to support digital freedom.
-----
(Y)es/(N)o: N
Account registered.

Which names would you like to activate HTTPS for?
We recommend selecting either all domains, or all domains in a VirtualHost/server block.
```

Continuando...

```
Account registered.

Which names would you like to activate HTTPS for?
We recommend selecting either all domains, or all domains in a VirtualHost/server block.
-----
1: araruama.rio.br
2: www.araruama.rio.br
-----

Select the appropriate numbers separated by commas and/or spaces, or leave input
blank to select all options shown (Enter 'c' to cancel):
Requesting a certificate for araruama.rio.br and www.araruama.rio.br

Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/araruama.rio.br/fullchain.pem
Key is saved at: /etc/letsencrypt/live/araruama.rio.br/privkey.pem
This certificate expires on 2025-09-08.
These files will be updated when the certificate renews.
Certbot has set up a scheduled task to automatically renew this certificate in the background.

Deploying certificate
Successfully deployed certificate for araruama.rio.br to /etc/nginx/sites-enabled/default
Successfully deployed certificate for www.araruama.rio.br to /etc/nginx/sites-enabled/default
Congratulations! You have successfully enabled HTTPS on https://araruama.rio.br and https://www.araruama.rio.br
```

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:

top.nic.br/site/araruama.rio.br/108673/#

rio.br egi.br

TOP
TESTE OS PADRÕES

Quem é TOP Sobre Referências Comunicados

Teste TOP - Site: araruama.rio.br

Resultado

Parabéns, seu domínio será adicionado em breve ao **Quem é TOP!**

100%

- ✔ Acessível via endereço IP moderno de Internet (IPv6) ⓘ
- ✔ Nome de domínio assinado (DNSSEC) ⓘ
- ✔ Conexão suficientemente segura (HTTPS) ⓘ
- ⚠ Uma ou mais opções de segurança recomendadas não estão configuradas (Opções de segurança) ⓘ
- ⚠ Autorização para roteamento não publicada no RPKI

» Descrição do relatório de teste

» [Link permanente do resultado do teste \(12-06-2025 09:39 -03\)](#)

Realize o teste novamente

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



» Mostrar detalhes

Cabeçalhos de segurança HTTP

-  *X-Frame-Options* 
-  *X-Content-Type-Options* 
-  *Content-Security-Policy (CSP)* 
-  *Existência de Referrer-Policy* 



Outras opções de segurança

-  *Security.txt* 

 Autorização para roteamento no RPKI

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:ECDSA-CHACHA20-POLY1305:ECDSA-AES128-GCM-SHA256:ECDSA-AES256-GCM-SHA384:ECDSA-CHACHA20-POLY1305:ECDSA-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:

```
Terminal
# This file contains important security parameters. If you modify this file
# manually, Certbot will be unable to automatically provide future security
# updates. Instead, Certbot will print and log an error message with a path to
# the up-to-date file that you will need to refer to when manually updating
# this file. Contents are based on https://ssl-config.mozilla.org

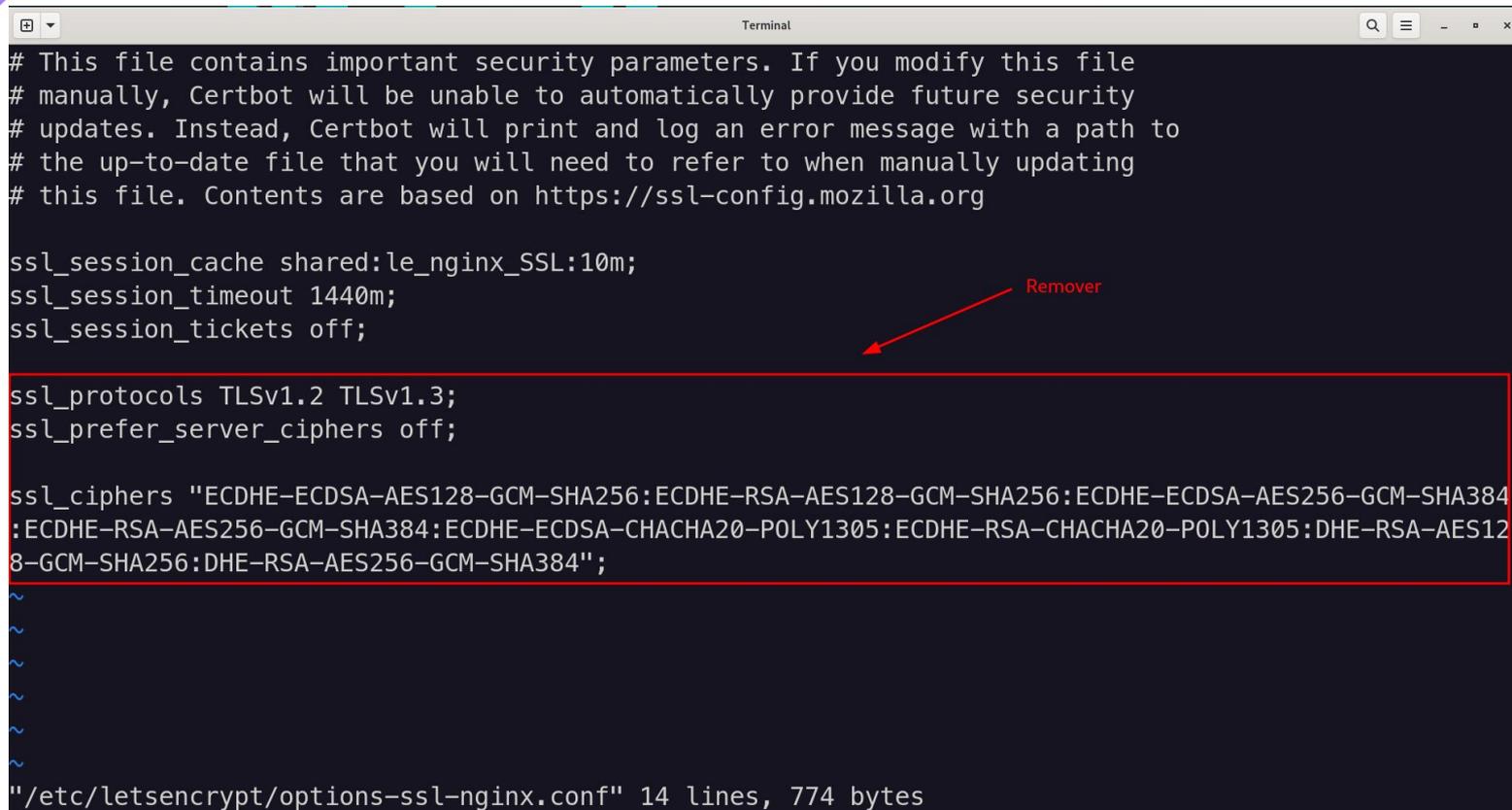
ssl_session_cache shared:le_nginx_SSL:10m;
ssl_session_timeout 1440m;
ssl_session_tickets off;

ssl_protocols TLSv1.2 TLSv1.3;
ssl_prefer_server_ciphers off;

ssl_ciphers "ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384
:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-AES12
8-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384";

~/
~/
~/
~/
~/

"/etc/letsencrypt/options-ssl-nginx.conf" 14 lines, 774 bytes
```



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):



» Mostrar detalhes

Cabeçalhos de segurança HTTP

-  *X-Frame-Options* 
-  *X-Content-Type-Options* 
-  *Content-Security-Policy (CSP)* 
-  Existência de *Referrer-Policy* 

Outras opções de segurança

-  **Security.txt** 



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 válida 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 eletrônico: 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](https://tools.ietf.org/html/rfc9116).

```
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-----

```
iQJLBAEBCgAlFiEEETLx9Tf3EoO627cgu0rIXDcmJnoFAmhKT6AXHGdvbmRpbUBp
c3Bmb2N1cy5uZXQuYnIACgkQu0rIXDcmJnq07A//X0XnvkfuhHHrs96DvBJMv3Hb
YtatxKQtSPX82srd0irFU9D9vhejlrnCk+2FzzKpdIDdCpoDj7C6PN4Db9w1MdbM
iqjswt+EnbdB6vTafbEuVFO6WicFCiVbBantyOxV4U3GcWvsSYyLkRbX0ME3OHQK
SFo7CdegSZn6aGnz8axA9YPRQldEbw9uv03ndrdRE8C8aualwpLdLgJpBKNtO7Za
lYQ/UNwWYEQCx4meo8TWDF4++TNVLGqxV1HovJvuU4Gz/bjtC4S/7j0GqO6TyH7L
jiWBDFWnmgwLQ+cIOLdDaTNSstXH/PYYvdRbJPM7CjcbiX3LwIy8yE64n9B+BL5UT
3hmLvELOEiMy1AEAqbJuHNtT6opMDNjAaKYbt2bU52axTPdYrH472OoUACtB/L+U
pGLypnA9HDwwDKjIjTowEnaoWPbEULkmkgZQy1NheGYc1kucrnAcvMNZK0Q+98SF
HA0+LkHvch2OqR9WGFESzLtKL3TaKdow6z2vqgm05oyxwCoUoprGPIGjDtDiVAoW
4Mptvnqo/+poK7rYidB6prODXa484QmbJDMkYmTYR6js4mB5YbOTahBfkoDgNLwT
QnNEwuyvDVDnFV8ThRQCsvq1t06i/zipCUB6Bh8mciVXSh+pk87y2BsArgWqUDMG
cdGAN8IWNqImbqZ4Abk=
```

=TV8q

-----END PGP SIGNATURE-----

Vamos ao resultado final:



Security.txt

Resultado:

Seu servidor *web* oferece um arquivo `security.txt` no local correto e seu conteúdo é sintaticamente válido.

Detalhes técnicos:

Endereço IP do servidor web	Informações encontradas
2804:ad4:ff18::5	Arquivo <code><code>security.txt</code></code> obtido de araruama.rio.br.

Descrição do teste:

Verificamos se o seu servidor *web* fornece um arquivo `security.txt` no local correto, se ele pode ser recuperado e seu conteúdo é sintaticamente válido.

O `security.txt` é um arquivo de texto com informações de contato que você disponibiliza em seu servidor *web*. Os especialistas em segurança podem utilizar estas informações para entrar em contato diretamente com o departamento ou a pessoa mais adequada em sua organização sobre vulnerabilidades encontradas e seu *site* ou sistemas de TI. Isso pode acelerar a correção de vulnerabilidades encontradas, reduzindo a oportunidade de exploração por partes mal-intencionadas.

A sintaxe do arquivo tem por objetivo a leitura por máquinas e humanos. As informações de contato podem ser um endereço de *e-mail*, um número de telefone e/ou uma página de Internet, por exemplo, um formulário. Observe que as informações de contato são públicas e podem ser abusadas, por exemplo, para enviar *e-mails* de *spam* para um endereço de *e-mail* publicado.

Além das informações de contato, o arquivo `security.txt` também deve conter uma data que indica sua validade. Para evitar informações desatualizadas, é recomendável que esta data seja inferior a um ano no futuro. Opcionalmente, podem ser incluídas outras informações relevantes para os especialistas em segurança, como um *link* para sua política sobre como deve ser informado sobre vulnerabilidades de segurança, geralmente chamado de política de Divulgação Coordenada de Vulnerabilidades (*Coordinated Vulnerability Disclosure*).

É recomendado que o arquivo `security.txt` seja assinado digitalmente usando uma assinatura de texto claro OpenPGP e deve ser incluir o campo `Canonical`, permitindo que a assinatura digital autentique a localização do arquivo. No momento **não** verificamos o conteúdo dos campos `Canonical`, pois a especificação do `security.txt` não é clara, especialmente quando redirecionamentos estão envolvidos. Verificamos se o arquivo `security.txt` está assinado, no entanto, nós **não** validamos a assinatura digital porque, infelizmente, não há um local padronizado para recuperar a chave PGP pública, que é necessária para validação da assinatura digital.

O arquivo deve ser publicado no caminho `/.well-known/`, ou seja, `https://example.nl/.well-known/security.txt`, para um nome de domínio ou um endereço IP, conforme RFC8615. Observe que colocar o arquivo `security.txt` no caminho de nível superior não é mais recomendado. Cada (sub)domínio deve ter seu próprio arquivo `security.txt`. O redirecionamento para um arquivo

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. Contribuiu 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.