



nixCraft → [ハウツー](#) → [暗号化](#) → OpenSUSE 15.4/15.5 で Let's Encrypt を使用して Nginx を保護する方法

 検索するには、「Enter」と入力してキーを押します...

# OpenSUSE 15.4/15.5 で Let's Encrypt を使用して Nginx を保護する方法

著者: Vivek Gite最終更新日: 2023 年 7 月 8 日[コメント3 件](#)

Let's Encrypt は、Web サイト、電子メール サーバー、データベース サーバーなどのための、無料で自動化されたオープンな認証局です。このページでは、Let's Encrypt を使用して Nginx Web サーバーの TLS 証明書をインストールし、OpenSUSE Linux バージョン 15.4/15.5 で SSL ラボ/セキュリティ ヘッダー A+ スコアを取得する方法を示します。



チュートリアル要件	
要件	OpenSUSE Linux 15.4/15.5 (Nginx 搭載)
ルート権限	<a href="#">はい</a>
難易度	<a href="#">中級</a>
カテゴリー	暗号化しましょう
OSの互換性	<a href="#">openSUSE</a> • <a href="#">SUSE</a>
EST（東部基準時。読書の時間）	5分
目次↓	
<a href="#">1手順</a>	

## チュートリアルの要件

[2前提条件](#)[3 acme.shのインストール](#)[4 Nginxの設定](#)[5 dhparamの作成](#)[6 SSL/TLS証明書の取得](#)[7 Nginx HTTPS 構成](#)[8 Let's Encrypt TLS 証明書のインストール](#)[9ファイアウォールを使用して HTTPS/443 ポートを開く](#)[10テストしてみよう](#)[11 の必須の acme.sh コマンド](#)[12結論](#)

## nixCraft: プライバシー第一、リーダー対応

**nixCraft は 1 人で操作できます。** AI や ML の助けを借りずに、すべてのコンテンツを自分で作成します。コンテンツを正確かつ最新の状態に保ちます。

**あなたのプライバシーは私にとって最優先事項です。** 私はあなたを追跡したり、広告を表示したり、スパムメールを送信したりしません。Linux と FLOSS の真の精神に基づいた純粋なコンテンツです。

**高速でクリーンなブラウジング体験。** nixCraft は高速かつ使いやすいように設計されています。ポップアップ、広告、Cookie バナー、その他の邪魔なものに対処する必要はありません。

**独立したコンテンツ作成者をサポートします。** nixCraft は愛の結晶であり、読者のサポートのおかげでのみ可能です。コンテンツをお楽しみいただけましたら、Patreon でサポートしていただくか、このページをソーシャルメディアやブログで共有してください。あらゆる点が役に立ちます。

[パトレオンに参加する→](#)

# OpenSUSE Linux で Let's Encrypt を使用して Nginx を保護する方法

SSL/TLS 証明書を取得する手順は次のとおりです。

1. acme.sh クライアントを取得し、次を実行します。

```
git clone https://github.com/Neilpang/acme.sh.git
```

2. インストールします:

```
./acme.sh --install --accountemail you@your-tld
```

3. デフォルトの CA を letsencrypt に設定します。

```
./acme.sh --set-default-ca --server letsencrypt
```

4. ドメインの nginx 構成を作成します。

```
vi /etc/nginx/vhosts.d/your-domain-name.conf
```

5. ドメインの SSL 証明書を取得します。

```
acme.sh --issue -d your-domain-name --nginx
```

6. Nginx で TLS を構成します。

```
vi /etc/nginx/conf.d/your-domain-name.conf
```

7. TLS 証明書を自動更新するための cron ジョブのセットアップ

8. firewalld を使用してポート 443 (HTTPS) を開きます。

```
sudo firewall-cmd --add-service=https
```

すべてのステップを詳しく見てみましょう。

## ステップ 1 – 必要なソフトウェアをインストールする (前提条件)

ターミナルを開き、次のコマンドを入力します。次のように、[CLI を使用して OpenSUSE Linux ソフトウェアとカーネルを更新してください](#)。acme.sh クライアントには、curl、wc、およびその他のパッケージが必要です。したがって、zypper コマンドを使用して必要なソフトウェアをインストールする必要があります。

```
$ sudo zypper ref
$ sudo zypper up
```

```
$ sudo zypper install wget curl bc git socat cronie
```

## [OpenSUSE Linux に Nginx をインストールする](#)

もう一度ジッパーを使用します。

```
$ sudo zypper install nginx
$ sudo systemctl enable nginx.service
```

シンボリックリンク /etc/systemd/system/multi-user.target.wants/nginx.servic

Nginx サーバーを起動し、systemctl コマンドを使用して確認します。表示される内容は次のとおりです。

```
$ sudo systemctl start nginx.service
$ sudo systemctl status nginx.service
```

- nginx.service - nginx HTTP およびリバース プロキシ サーバー
  - ロード済み: ロード済み (/usr/lib/systemd/system/nginx.service; 有効; ベン
  - アクティブ: 2020-07-06 月 18:49:32 UTC 以降 **アクティブ (実行中)** 。2分4秒前
  - メイン PID: 13990 (nginx)
  - タスク: 2
  - CGroup: /system.slice/nginx.service
    - \$-13990 nginx: マスター プロセス /usr/sbin/nginx -g デーモン オフ
    - └13991 nginx: ワーカープロセス

```
Jul 06 18:49:32 opensuse-nixcraft-42 systemd[1]: nginx HTTP およびリバース
Jul 06 18:49:32 opensuse-nixcraft-42 nginx[13989]: nginx: 設定ファイル /etc
7月06日 18:49:32 opensuse-nixcraft-42 nginx[13989]: nginx: 設定ファイル /et
Jul 06 18:49:32 opensuse-nixcraft-42 systemd[1]: nginx HTTP およびリバース
```

[最後に、OpenSUSE Linux で firewalld を使用して HTTP ポート 80 を開きます。](#)

```
$ sudo firewall-cmd --zone=public --add-service=http
$ sudo firewall-cmd --zone=public --add-service=http --permanent
$ sudo firewall-cmd --list-services
```

ssh dhcpv6-クライアント http

## ステップ 2 – acme.sh Let's Encrypt クライアントのインストール

[acme.sh](https://github.com/Neilpang/acme.sh) リポジトリのクローンを作成する必要があります。クライアントをインストールしますが、最初に su コマンド/sudo コマンドを使用して root ユーザーとしてログインします。

```
$ cd /tmp/
$ git clone https://github.com/Neilpang/acme.sh.git
```

```
$ sudo -i
# touch /root/.bashrc
# cd /tmp/acme.sh/
# ./acme.sh --install --accountemail your-email-id@domain-here
# ./acme.sh --set-default-ca --server letsencrypt
```

```

opensuse-nixcraft-42:/tmp # git clone https://github.com/Neilpang/acme.sh.git
Cloning into 'acme.sh'...
remote: Enumerating objects: 10909, done.
remote: Total 10909 (delta 0), reused 0 (delta 0), pack-reused 10909
Receiving objects: 100% (10909/10909), 4.21 MiB | 18.93 MiB/s, done.
Resolving deltas: 100% (6477/6477), done.
opensuse-nixcraft-42:/tmp # touch /root/.bashrc
opensuse-nixcraft-42:/tmp # cd acme.sh/
opensuse-nixcraft-42:/tmp/acme.sh # EMAIL="webmaster@cyberciti.biz"
opensuse-nixcraft-42:/tmp/acme.sh # ./acme.sh --install --accountemail "$EMAIL"
[Mon Jul 6 18:20:55 UTC 2020] Installing to /root/.acme.sh
[Mon Jul 6 18:20:55 UTC 2020] Installed to /root/.acme.sh/acme.sh
[Mon Jul 6 18:20:55 UTC 2020] Installing alias to '/root/.bashrc'
[Mon Jul 6 18:20:55 UTC 2020] OK, Close and reopen your terminal to start using a
cme.sh
[Mon Jul 6 18:20:55 UTC 2020] Installing cron job
28 0 * * * "/root/.acme.sh"/acme.sh --cron --home "/root/.acme.sh" > /dev/null
[Mon Jul 6 18:20:55 UTC 2020] Good, bash is found, so change the shebang to use b
ash as preferred.
[Mon Jul 6 18:20:56 UTC 2020] OK
opensuse-nixcraft-42:/tmp/acme.sh # source /root/.bashrc
opensuse-nixcraft-42:/tmp/acme.sh # acme.sh --list
Main_Domain KeyLength SAN_Domains Created Renew
opensuse-nixcraft-42:/tmp/acme.sh # acme.sh --version
https://github.com/acmesh-official/acme.sh
v2.8.7
© www.cyberciti.biz
opensuse-nixcraft-42:/tmp/acme.sh #

```

acme.sh のバージョンを表示するには、次のコマンドを実行します。

```
# acme.sh --version
```

Outputs:

```
https://github.com/acmesh-official/acme.sh
```

```
v3.0.1
```

## ステップ 3 – OpenSUSE 上の http サーバーの基本的な Nginx 構成

次のように、opensuse.cyberciti.biz という名前のドメインの新しい構成を作成します (opensuse.cyberciti.biz を実際のドメイン名に自由に置き換えてください)。

```
# vi /etc/nginx/vhosts.d/opensuse.cyberciti.biz.conf
```

次のディレクティブを追加します。

```

# http ポート80 の設定
サーバー{
    リッスン      80デフォルトサーバー; # IPv4
    listen [ :: ] : 80デフォルトサーバー; # IPv6
    サーバー名 opensuse.cyberciti.biz; # ドメイン名

```



ファイルを保存して閉じます。次のようにnginx のセットアップをテストし、nginx サーバーをリロードします。

## ステップ 4 – dhparam.pem ファイルを作成する

[illegible]

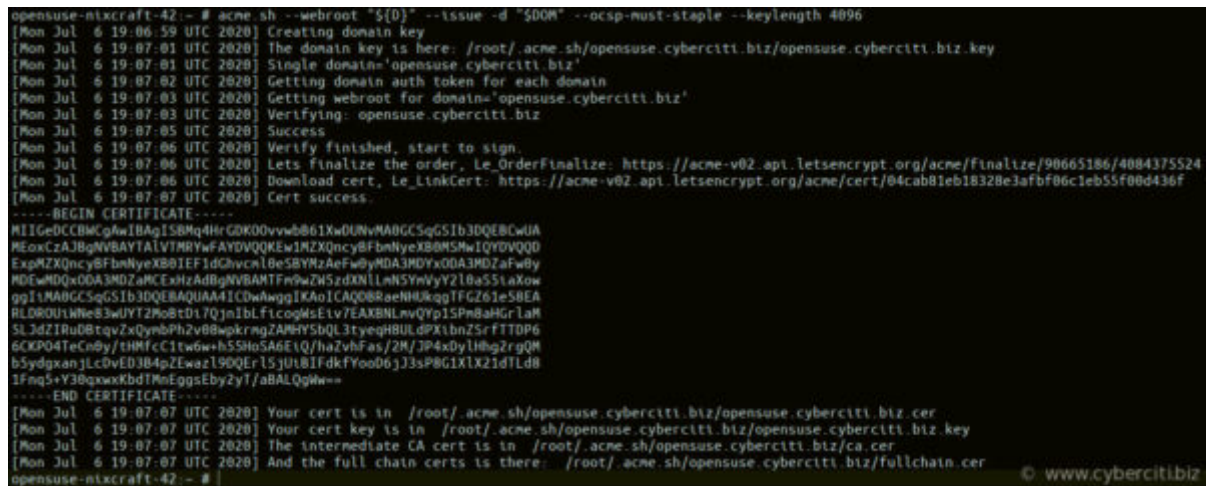
## ステップ5-ドメインの証明書を取得する

手順 3 で構成した Nginx サーバーを使用して証明書を発行できます。ただし、サーバーが Cloudflare などのリバース プロキシ CDN の背後にある場合は、以下で説明するようにスタンドアロン モードを使用してください。

事前設定された Nginx を使用して証明書を発行する

ドメイン名を \$DOM シェル変数に設定します。

```
# DOM="opensesuse.cyberciti.biz"
# D="/srv/www/htdocs"
# mkdir -pv ${D}/.well-known/acme-challenge/
# acme.sh --webroot "${D}" --issue -d "$DOM" --ocsp-must-staple --
keylength 4096
## GET ecc cert too. Only ec-384 or ec-256 ##
# acme.sh --webroot "${D}" --issue -d "$DOM" --ocsp-must-staple --
keylength ec-384
```



```
opensesuse-nixcraft-42:~ # acme.sh --webroot "${D}" --issue -d "$DOM" --ocsp-must-staple --keylength 4096
[Mon Jul 6 19:06:59 UTC 2020] Creating domain key
[Mon Jul 6 19:07:01 UTC 2020] The domain key is here: /root/.acme.sh/opensesuse.cyberciti.biz/opensesuse.cyberciti.biz.key
[Mon Jul 6 19:07:01 UTC 2020] Single domain='opensesuse.cyberciti.biz'
[Mon Jul 6 19:07:02 UTC 2020] Getting domain auth token for each domain
[Mon Jul 6 19:07:03 UTC 2020] Getting webroot for domain='opensesuse.cyberciti.biz'
[Mon Jul 6 19:07:03 UTC 2020] Verifying: opensesuse.cyberciti.biz
[Mon Jul 6 19:07:05 UTC 2020] Success
[Mon Jul 6 19:07:06 UTC 2020] Verify finished, start to sign.
[Mon Jul 6 19:07:06 UTC 2020] Lets finalize the order, Le_OrderFinalize: https://acme-v02.api.letsencrypt.org/acme/finalize/90665186/4084375524
[Mon Jul 6 19:07:06 UTC 2020] Download cert, Le_LinkCert: https://acme-v02.api.letsencrypt.org/acme/cert/04cab81eb18328e3afb06c1eb55f00d436f
[Mon Jul 6 19:07:07 UTC 2020] Cert success.
-----BEGIN CERTIFICATE-----
MIIEGzCCBMGgAwIBAgISB8q4HrGDk00vwwb61Xw0UNvMA0GCSqGSIb3DQEBAQUA
MGEoxCzA3BgNVBAYTA1VIMRYwFAYDVQQKEw1MZXQncybFbnNyeXB0MScwIjYDVQ0Q
ExpMZXQncybFbnNyeXB0IEF1dChvcn1BeSBYb3R5Y29yYDA3MDYxODAzMDZaFwBy
HDEwMDQxODAzMDZaMCEwH2ZAdBgNVBAMTFm9wZXN5b3R5Y29yY210a55S1aXow
ggI1MA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQD08RaeNHUkqgTFGZ61eS8EA
RLDR0U1WneB3wUYT2MoBtD17Qjn1blf1cogMSE1v7EAXBNLwQp1SPn8aHGrLaM
%3dZIRuDBtqvZxQyhbPh2vB8wKrmgZAMHYSbQL3tyeqHBUldPXi1bnZsrFTT0P6
6CKP04TeCnby/tHMFc1tw6w+h5SHoSA6EiQ/haZvhFas/2M/7P4x0y1lHq2rgQM
bSydpxanjLcDvED384pZLwaz190QEr15jU1BIFdkFYooD6j33sP8G1X121dLd8
1Fnq5+Y30qxxKbdTmEggsEby2y1aBALQgWw==
-----END CERTIFICATE-----
[Mon Jul 6 19:07:07 UTC 2020] Your cert is in /root/.acme.sh/opensesuse.cyberciti.biz/opensesuse.cyberciti.biz.cert
[Mon Jul 6 19:07:07 UTC 2020] Your cert key is in /root/.acme.sh/opensesuse.cyberciti.biz/opensesuse.cyberciti.biz.key
[Mon Jul 6 19:07:07 UTC 2020] The intermediate CA cert is in /root/.acme.sh/opensesuse.cyberciti.biz/ca.cert
[Mon Jul 6 19:07:07 UTC 2020] And the full chain certs is there: /root/.acme.sh/opensesuse.cyberciti.biz/fullchain.cert
opensesuse-nixcraft-42:~ #
```

## スタンドアロンモードで証明書を発行する

```
# DOM="opensesuse.cyberciti.biz"
# acme.sh --issue --standalone -d "$DOM" --ocsp-must-staple --
keylength 4096
## GET ecc cert too. Only ec-384 or ec-256 ##
# acme.sh --issue --standalone -d "$DOM" --ocsp-must-staple --
keylength ec-384
```

どこ、

- `--webroot /srv/www/htdocs` : Web ルート モードの Web ルート フォルダを指定します。 `/.well-known/acme-challenge/` をルートに作成する必要があります。
- `--issue` : 証明書を発行します。
- `-d domain-name` : 発行、更新、取り消しに使用するドメインを指定します。何度でも使えます。例: `acme.sh --issue -d www.cyberciti.biz -d`



```
ftp.cyberciti.biz --ocsp-must-staple --keylength 4096
```

- `--ocsp-must-staple` : [ocsp 必須ステープル拡張子を生成します](#)
- `--keylength 4096` : ドメイン キーの長さを指定します: 2048、3072、4096、8192 または ec-256、ec-384、ec-521。
- `--keylength ec-256` : [楕円曲線暗号 \(ECC\)](#) は、有限体上の楕円曲線の代数構造に基づく公開キー暗号へのアプローチです。ECC では、非 EC 暗号化 (単純なガロア体に基づく) と比較して、より小さなキーが許可され、同等のセキュリティが提供されます。

## ステップ 6 – OpenSUSE Linux サーバーで Nginx を構成する

構成ファイルを編集します。

```
# vi /etc/nginx/vhosts.d/opensuse.cyberciti.biz.conf
```

次のように更新します。

```
# http ポート80 の設定
サーバー{
    リッスン      80デフォルトサーバー; # IPv4
    listen [ :: ] : 80デフォルトサーバー; # IPv6
    サーバー名 opensuse.cyberciti.biz;
    アクセス_ログオフ;
    error_ログオフ;
    ルート/srv/www/htdocs;301 https://$host$request_uri

    を返します。}

# https ポート443構成
サーバー{
    リッスン443 ssl http2; # IPv4
    リッスン[ :: ] : 443 ssl http2; # HTTP/ 2 TLS IPv6
    サーバー名 opensuse.cyberciti.biz; # ドメイン名

    # ドキュメントルートを設定する
    場所 / {
        ルート/srv/www/htdocs;
        インデックスindex.htmlインデックス.htm;
    }

    # この vhos のアクセスとエラーのログを設定します
    access_log /var/log/nginx/https.opensuse.cyberciti.biz_access.log;
    エラーログ /var/log/nginx/https.opensuse.cyberciti.biz_error.log;

    # TLS/SSL 設定
    ssl_certificate /etc/nginx/ssl/cyberciti.biz/opensuse.cyberciti.biz.fullchain.cer;
```

```

ssl_certificate_key /etc/nginx/ssl/cyberciti.biz/opensuse.cyberciti.biz.key;
# ECC 証明書
ssl_certificate /etc/nginx/ssl/cyberciti.biz/opensuse.cyberciti.biz.fullchain.cer.ecc;
ssl_certificate_key /etc/nginx/ssl/cyberciti.biz/opensuse.cyberciti.biz.key.ecc;
ssl_dhparam /etc/nginx/ssl/cyberciti.biz/dhparams.pem;
# 少し最適化
ssl_session_timeout 1d;
ssl_session_cache 共有:NixCraftSSL:10m; # 約40000セッション
ssl_session_ticket オフです。

# TLS バージョン1.2および1.3のみ
ssl_protocols TLSv1.2 TLSv1.3;
ssl_ciphers ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SH
ssl_prefer_server_ciphers オフ;

# HSTS ( ngx_http_headers_module が必要です) ( 63072000秒)
add_header Strict-Transport-Security "max-age=63072000"常に;
add_header X-Content-Type-Options は常に"nosniff" ;
add_header X-Frame-Options常に"SAMEORIGIN" ;
add_header X-Xss-Protection "1; mode=block"常に;
add_header Referrer-Policy strict-origin-when-cross-origin 常に;
add_header 機能ポリシー"加速度計 'なし'; カメラ 'なし'; 地理位置情報 'なし'; ジャイロスコープ 'なし';
# 警告: HTTP Content-Security-Policy 応答ヘッダーにより、sysadmin/developers が許可されます。
# ユーザーエージェントが特定のページに対してロードできるリソースを制御します。
# 設定が間違っていると、サードパーティのスクリプト/広告ネットワークに問題が発生する可能性があります。し:
# https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy
add_header content-security-policy "default-src https://opensuse.cyberciti.biz:443"常に;

# OCSP ステープル留め
ssl_stapling オン;
ssl_stapling_verify オン;

# ルート CA と中間証明書を使用して OCSP 応答の信頼チェーンを検証する
ssl_trusted_certificate /etc/nginx/ssl/cyberciti.biz/opensuse.cyberciti.biz.fullchain.cer;

# リゾルバーの IP アドレスに置き換えます
リゾルバ 1.1.1.1;
}

```

## サンプルindex.html

次のように新しいファイルを作成します。

```
# vi /srv/www/htdocs/index.html
```

次のコードを追加します。

```
<!doctype html>
< html lang = "en" >
```

```
< head >
< title > OpenSUSE.Cyberciti.Biz Nginx サーバー< / title >
< meta charset = "utf-8" / >
< meta name = "viewport" content = "width=device-width,initial-scale=1.0" >
< / head >
< body >
<article>
< h2 > Hello, World!< / h2>
< p >これは、OpenSUSE Linux 15.2 と無料の TLS 証明書を備えた Nginx を搭載したテスト サーバーです。< / p>
< hr >
< small >
電子メールでご連絡ください< a href = "mailto:webmaster@cyberciti.biz" > webmaster@cyberciti.biz < /
< / small >
< / body >
< / html >
```

## ステップ 7 – OpenSUSE 15.4/15.5 に Let's Encrypt TLS 証明書 をインストールする

発行された証明書を nginx サーバーにインストールし、サーバーをリロードします。 ECC 証明書もインストールします。

```
# DOM="opensuse.cyberciti.biz"
# acme.sh -d "$DOM" \
--install-cert \
--reloadcmd "systemctl reload nginx" \
--fullchain-file "/etc/nginx/ssl/cyberciti.biz/$DOM.fullchain.cer" \
--key-file "/etc/nginx/ssl/cyberciti.biz/$DOM.key" \
--cert-file "/etc/nginx/ssl/cyberciti.biz/$DOM.cer"
```

```
# acme.sh -d "$DOM" \
--ecc \
--install-cert \
--reloadcmd "systemctl reload nginx" \
--fullchain-file "/etc/nginx/ssl/cyberciti.biz/$DOM.fullchain.cer.ecc" \
--key-file "/etc/nginx/ssl/cyberciti.biz/$DOM.key.ecc" \
--cert-file "/etc/nginx/ssl/cyberciti.biz/$DOM.cer.ecc"
```

```
opensuse@nixcraft-42:~$ # DOM="opensuse.cyberciti.biz"
opensuse@nixcraft-42:~$ # acme.sh -d "$DOM" \
> --install-cert \
> --reloadcmd "systemctl reload nginx" \
> --fullchain-file "/etc/nginx/ssl/cyberciti.biz/$DOM.fullchain.cer" \
> --key-file "/etc/nginx/ssl/cyberciti.biz/$DOM.key" \
> --cert-file "/etc/nginx/ssl/cyberciti.biz/$DOM.cer"
[Mon Jul 6 19:37:11 UTC 2020] Installing cert to:/etc/nginx/ssl/cyberciti.biz/opensuse.cyberciti.biz.cer
[Mon Jul 6 19:37:11 UTC 2020] Installing key to:/etc/nginx/ssl/cyberciti.biz/opensuse.cyberciti.biz.key
[Mon Jul 6 19:37:11 UTC 2020] Installing full chain to:/etc/nginx/ssl/cyberciti.biz/opensuse.cyberciti.biz.fullchain.cer
[Mon Jul 6 19:37:11 UTC 2020] Run reload cmd: systemctl reload nginx
[Mon Jul 6 19:37:11 UTC 2020] Reload success
opensuse@nixcraft-42:~$ # acme.sh -d "$DOM" \
> --ecc \
> --install-cert \
> --reloadcmd "systemctl reload nginx" \
> --fullchain-file "/etc/nginx/ssl/cyberciti.biz/$DOM.fullchain.cer.ecc" \
> --key-file "/etc/nginx/ssl/cyberciti.biz/$DOM.key.ecc" \
> --cert-file "/etc/nginx/ssl/cyberciti.biz/$DOM.cer.ecc"
[Mon Jul 6 19:37:20 UTC 2020] Installing cert to:/etc/nginx/ssl/cyberciti.biz/opensuse.cyberciti.biz.cer.ecc
[Mon Jul 6 19:37:20 UTC 2020] Installing key to:/etc/nginx/ssl/cyberciti.biz/opensuse.cyberciti.biz.key.ecc
[Mon Jul 6 19:37:20 UTC 2020] Installing full chain to:/etc/nginx/ssl/cyberciti.biz/opensuse.cyberciti.biz.fullchain.cer.ecc
[Mon Jul 6 19:37:20 UTC 2020] Run reload cmd: systemctl reload nginx
[Mon Jul 6 19:37:20 UTC 2020] Reload success
opensuse@nixcraft-42:~$
```

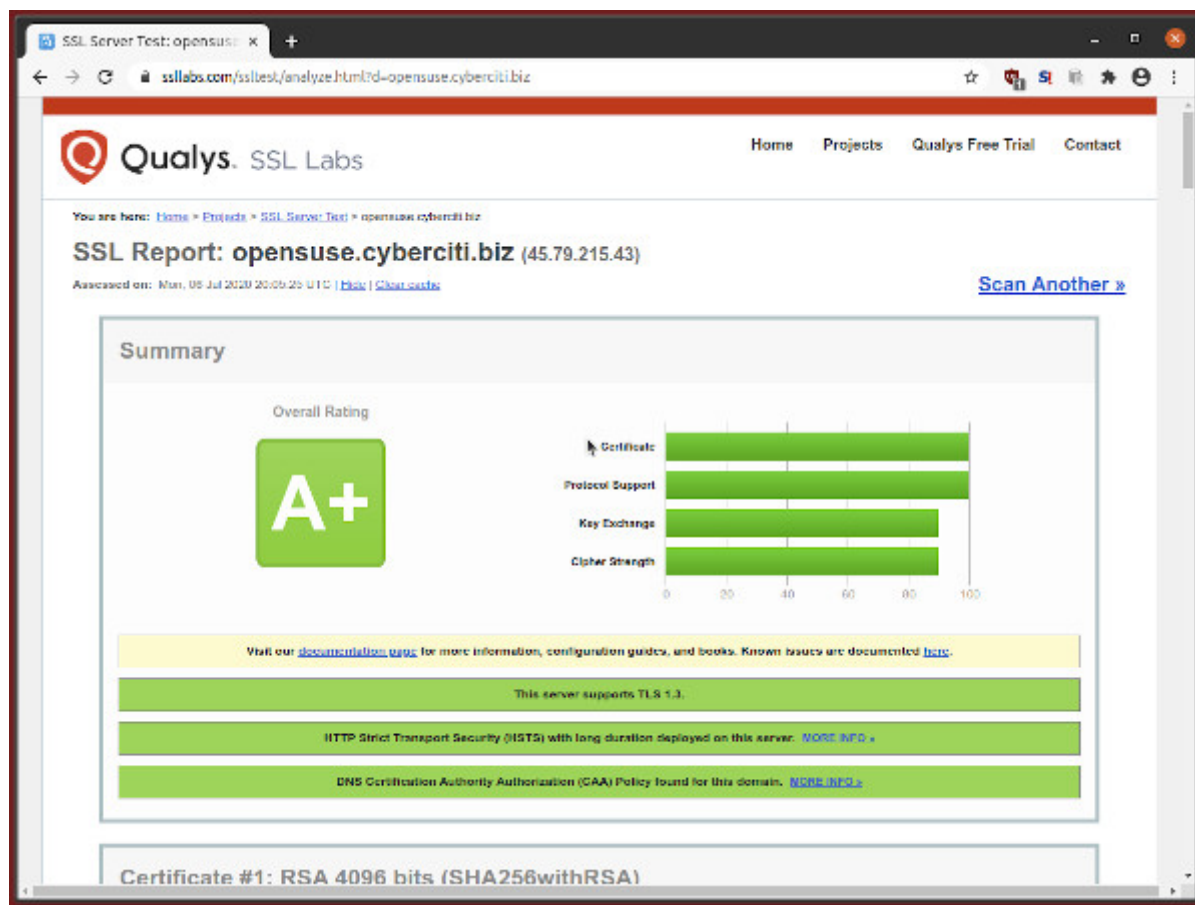
## ステップ 8 – TCP ポート 443 [HTTPS ポート] を開きます。

[次のように、OpenSUSE Linux 上のファイアウォール](#)を使用して HTTPS TCP ポート 443 を開きます。

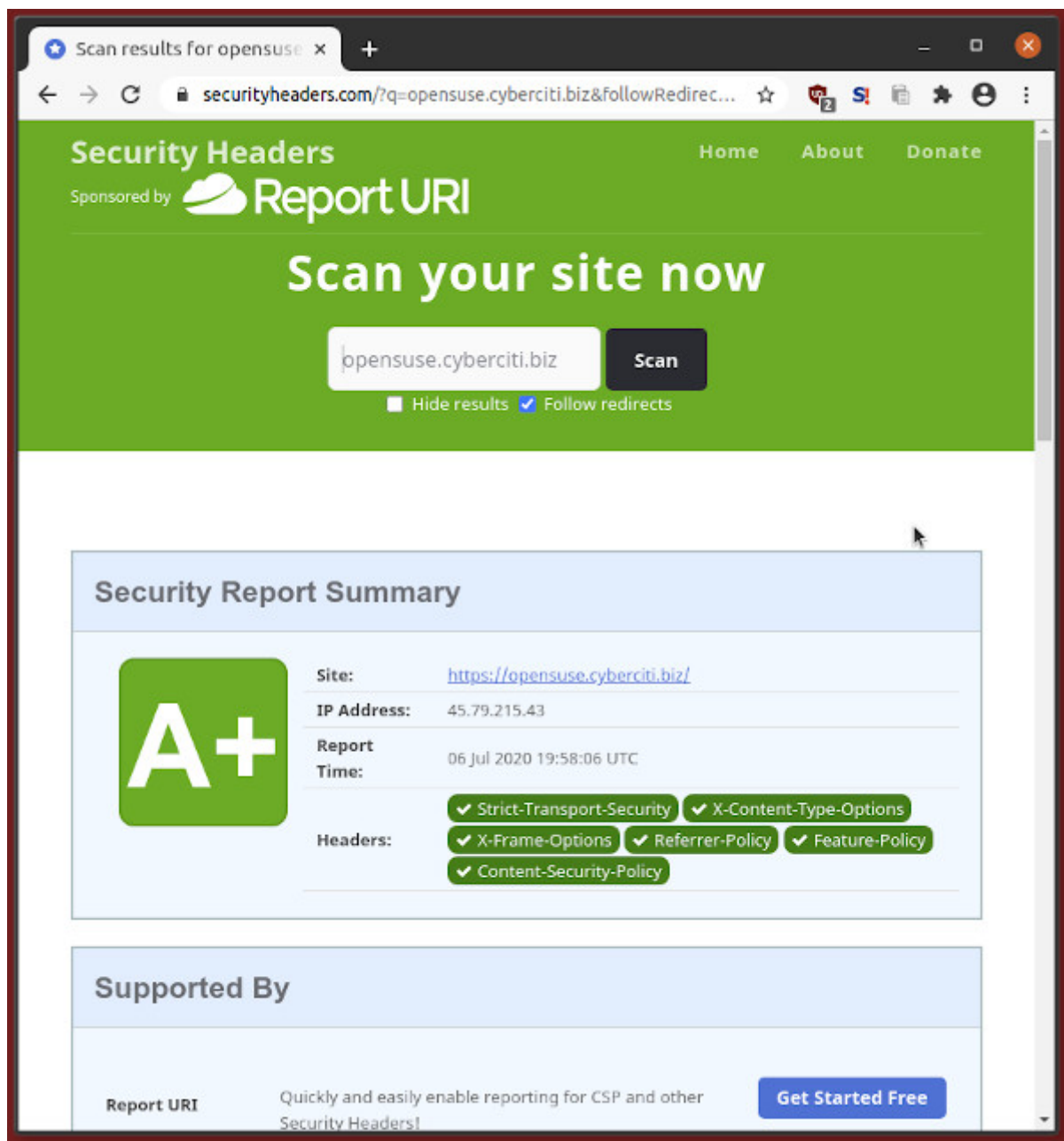
```
# firewall-cmd --zone=public --add-service=https
# firewall-cmd --zone=public --add-service=https --permanent
# firewall-cmd --list-services
# curl -I https://opensuse.cyberciti.biz/
```

## ステップ 9 – テストする

[SSL ラボ](#)のテスト:



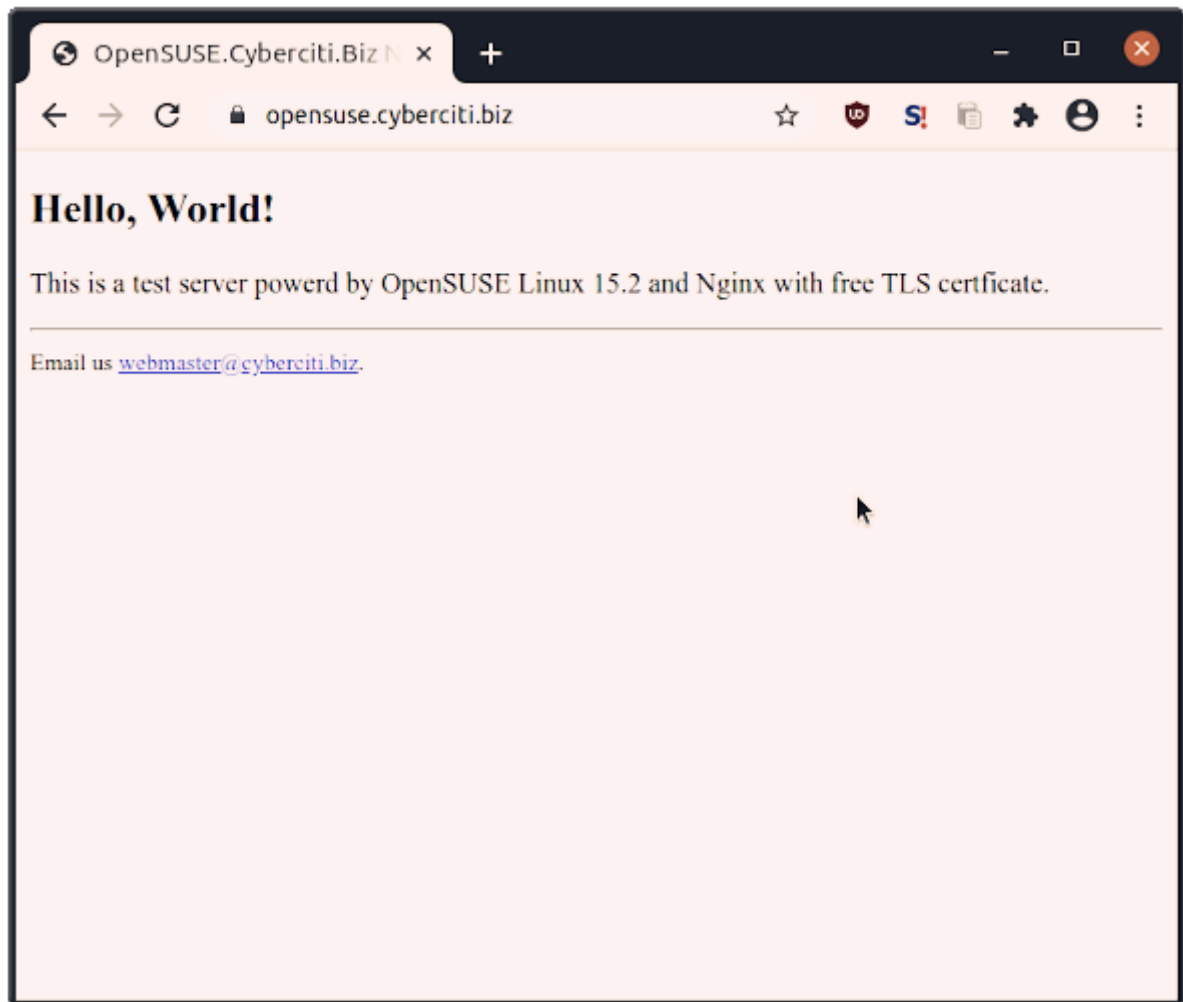
[セキュリティヘッダーのテスト:](#)



Web ブラウザを起動し、次のようにドメインを入力します。

https://opensuse.cyberciti.biz





## ステップ 10 – 必須の acme.sh コマンド

すべての証明書をリストするには、次を実行します。

```
# acme.sh --list
```

```
Main_Domain KeyLength SAN_Domains Created Renew
opensuse.cyberciti.biz "4096" no Mon Jul 6 19:07:07 UTC 2020 Fri Sep 4 19
opensuse.cyberciti.biz "ec-384" いいえ 2020 年 7 月 6 日月曜日 19:11:54 UTC
```

opensuse.cyberciti.biz という名前のドメインの証明書を更新します cron ジョブは証明書の更新も試行することに注意してください。これはデフォルトで次のようにインストールされます (ユーザー側でのアクションは必要ありません)。 [cron ジョブの実行を確認するには](#):

```
# acme.sh --renew -d opensuse.cyberciti.biz
# acme.sh --force --renew -d opensuse.cyberciti.biz -d
```

```
www.cyberciti.biz
```

```
# crontab -l
```

```
28 0 * * * "/root/.acme.sh"/acme.sh --cron --home "/root/.acme.sh" > /dev,
```

acme.sh クライアントをアップグレードするには、次を実行します。

```
# acme.sh --upgrade
```

出力:

```
[木曜日 15 June 2023 06:40:57 PM UTC ]オンライン アーカイブからインストールしています。
[ 2023 年 6 月 15 日木曜日 06:40:57 PM UTC ]ダウンロード https://github.com/acmesh-official/acme.sh/
[ 2023 年 6 月 15 日木曜日 06:40:58 PM UTC ] master.tar.gz を抽出
[木曜日 15 6 月 2023 06:40:58 PM UTC ] /root/.acme.sh にインストール
[木曜日 15 6 月 2023 06:40:58 PM UTC ] /root/.acme.sh にインストール/acme.sh
[木曜日 15 June 2023 06:40:58 PM UTC ]よし、bash が見つかったので、優先的に bash を使用するようにシバン
[木曜日 15 June 2023 06:40:59 PM UTC ] OK
[木曜日 15 June 2023 06:40:59 PM UTC ]インストールは成功しました。
[ 2023 年 6 月 15 日木曜日 06:41:05 PM UTC ]アップグレードが成功しました。
```

バージョンを再度確認します:

```
# acme.sh --version
```

出力:

```
https://github.com/acmesh-official/acme.sh
v3.0.6
```

助けを得るのは簡単です。[more コマンド](#)または[lessコマンド](#)ページャーを使用して実行し、一度に 1 つの画面を表示します。

```
# acme.sh --help | more
```

または、[grep コマンド](#)または[egrep コマンド](#)を

```
# acme.sh --help | more
```

使用してヘルプを絞り込むこともできます。例: または

```
# acme.sh --help | grep -w -- 'version'
```

```
# acme.sh --help | grep -wE -- '--(version|upgrade)'
```

## 結論

OCSP Stapling および ECC 証明書を使用して、OpenSUSE Linux 15.4/15.5 nginx ベースのサーバーに Let's Encrypt TLS/SSL 証明書をインストールしてセットアップする方法を説明します。詳細については、[acme.sh プロジェクトのホームページ](#)を参照してください。

このエントリは、**OpenSUSE Linux LEMP スタック チュートリアル**シリーズの3つのうち2つです。シリーズの残りの部分を読み続けてください。

1. [OpenSUSE Linux に Nginx をインストールして使用する](#)
2. OpenSUSE Linux で Let's Encrypt を使用して Nginx を保護する
3. [OpenSUSE Linux 15.2/15.1 に PHP をインストールする](#)

このエントリは、「**Let's Encrypt を使用した Secure Web Server チュートリアル**」シリーズの15件中9 件目です。シリーズの残りの部分を読み続けてください。

1. [Debian/Ubuntu Linux で Lets Encrypt をセットアップする](#)
2. [Debian/Ubuntu で Lets Encrypt 証明書を使用してLighttpdを保護する](#)
3. [Alpine Linuxで Lets Encrypt 証明書を使用してNginxを構成する](#)
4. [CentOS 7で Lets Encrypt を使用したNginx](#)

5. [RHEL 8で Lets Encrypt 証明書を使用するApache](#)
6. [Lets Encrypt 証明書を使用したCentOS 8とApache](#)
7. [Nginx用のCentOS 8に Lets Encrypt 証明書をインストールする](#)
8. [Let's Encrypt 証明書を強制的に更新する](#)
9. [Let's Encrypt 証明書を使用したOpenSUSE Linuxおよび Nginx](#)
10. [TLS 1.2 / 1.3のみを使用するようにNginxを構成する](#)
11. [acme.sh とCloudflare DNSを使用してワイルドカード証明書を暗号化しましょう](#)
12. [DNS 検証を使用した Ubuntu 18.04 で Let's Encrypt を使用した Nginx](#)
13. [AWS Route 53 acme.sh を使用してワイルドカード証明書を暗号化しましょう](#)
14. [AWS Route 53 をCloudflare に変換する acme.sh を使用して DNS を暗号化しましょう](#)
15. [証明書がスキップ、更新、またはエラーになった場合の Let's Encrypt の電子メール通知](#)

気づきましたか？ 🤔

nixCraft には広告が表示されず、プライバシーとセキュリティが保護されます。サイトの運営を継続するには読者のサポートに依存しています。Patreon で購読するか、PayPal を通じて 1 回限りのサポートでサポートしていただくことをご検討ください。あなたのサポートは、ホスティング、CDN、DNS、チュートリアルを作成にかかるコストをカバーするのに役立ちます。

[パトレオンに参加する→](#)

[ペイパル→](#)

**著者について:** Vivek Gite は、Linux とオープンソースに関する最も古い運営ブログである nixCraft の創設者です。彼は 7,000 以上の投稿を執筆し、多くの読者が IT トピックを習得できるよう支援しました。[RSS フィード](#)または[電子メールニュースレター](#)を通じて nixCraft コミュニティに参加してください。

🤔これは役に立ちましたか? [感謝の気持ちやフィードバックを示すためにコメント](#)を追加してください。

🔍 検索するには、「Enter」と入力してキーを押します...

## コメント3 件... 1 つ追加

**Esha**2020年7月18日 4時15分


素晴らしいくて、まるで魔法のように働きました。

↩ ∞

**フラン**2023年6月13日 13時35分

こんにちは。80 や 443 以外のポートを使用している場合はどうなりますか？

↩ ∞

 **Vivek Gite**2023年6月15日 18時34分



listenを使用して IP のアドレスとポートを設定します。デフォルトは、nginx vhost 構成で次のように 80 と 443 です。

```
80 デフォルトサーバーをリッスンします。# IPv4  
リッスン [::]:80 デフォルトサーバー; # IPv6
```

そして：

```
443 ssl http2 をリッスンします。# IPv4  
リッスン [::]:443 ssl http2; # HTTP/2 TLS IPv6
```

必要に応じて 80 と 443 を変更します。たとえば、443 の 8080 です。

```
リッスン 8080 デフォルトサーバー; # IPv4  
リッスン [::]:8080 デフォルトサーバー; # IPv6
```

または TLS/SSL の場合

```
4433 ssl http2 をリッスンします。# IPv4  
リッスン [::]:4433 ssl http2; # HTTP/2 TLS IPv6
```

[nginx サービスを再起動します。](#)

```
sudo systemctl nginx.service を再起動する
```

試して。ブラウザを開きます。

```
## HTTP URL  
http://あなたのドメイン:8080/  
## TLS URL ##  
https://あなたのドメイン:4433/>
```

↩ ∞

## 返信を残す

あなたのメールアドレスが公開されることはありません。 \*が付いているフィールドは必須です

## コメント\*

## 名前

コメントを投稿



コードサンプルにはHTML `<pre>...</pre>`を使用します。コメントはサイト管理者の承認後にのみ表示されます。

---

次の FAQ: [FreeBSD サーバー/jail で SSHD を有効にする方法](#)

前の FAQ: [OpenSUSE で Isof パッケージをインストールして「zypper ps」エラーを解決する](#)