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。

チュートリアルの要件	
要件	OpenSUSE Linux 15.4/15.5 (Nginx 搭載)
ルート権限	はい
難易度	中級
カテゴリー	暗号化しましょう
OSの互換性	openSUSE • SUSE
EST（東部基準時。読書の時間	5分

目次↓

チュートリアルの要件

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 /et
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 firewllad 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
```

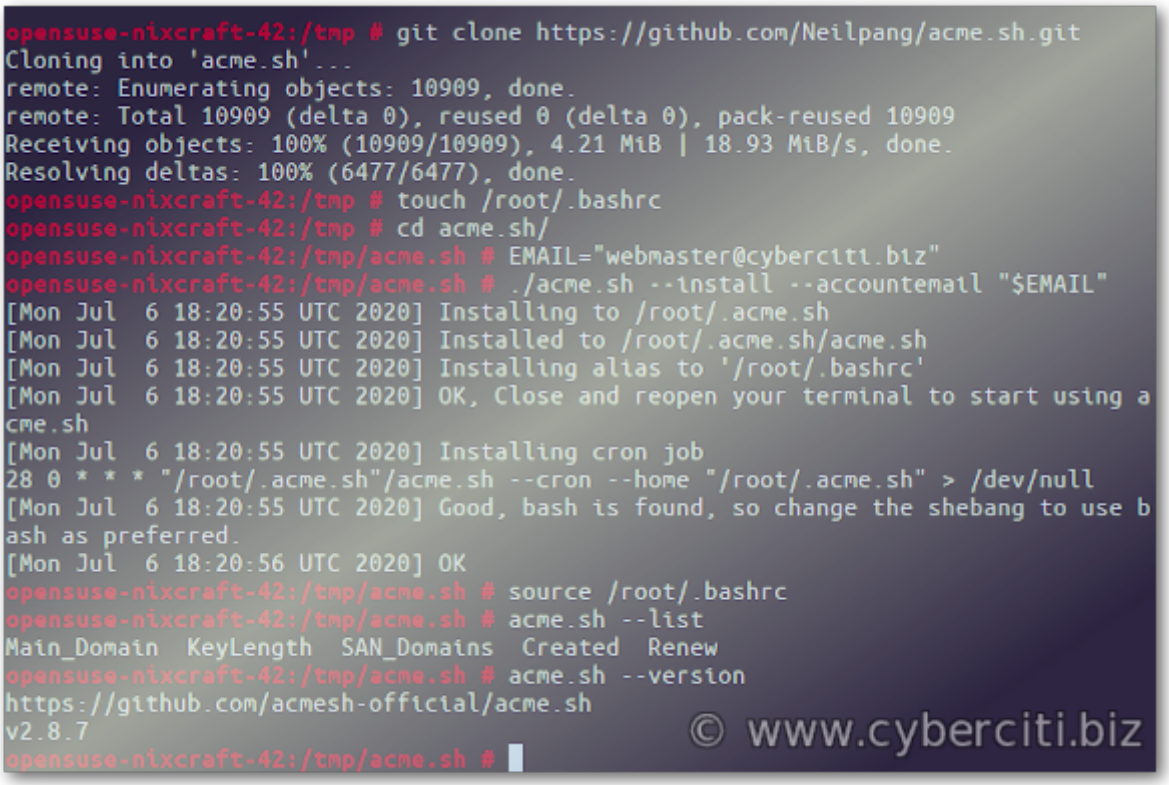
```
ssh dhcpv6-クライアント http
```

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
```



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
```

```
        access_log /var/log/nginx/http_opensuse.cyberciti.biz_access.log;
        エラーログ /var/log/nginx/http_opensuse.cyberciti.biz_error.log;
        ルート/srv/www/htdocs;
    }
```

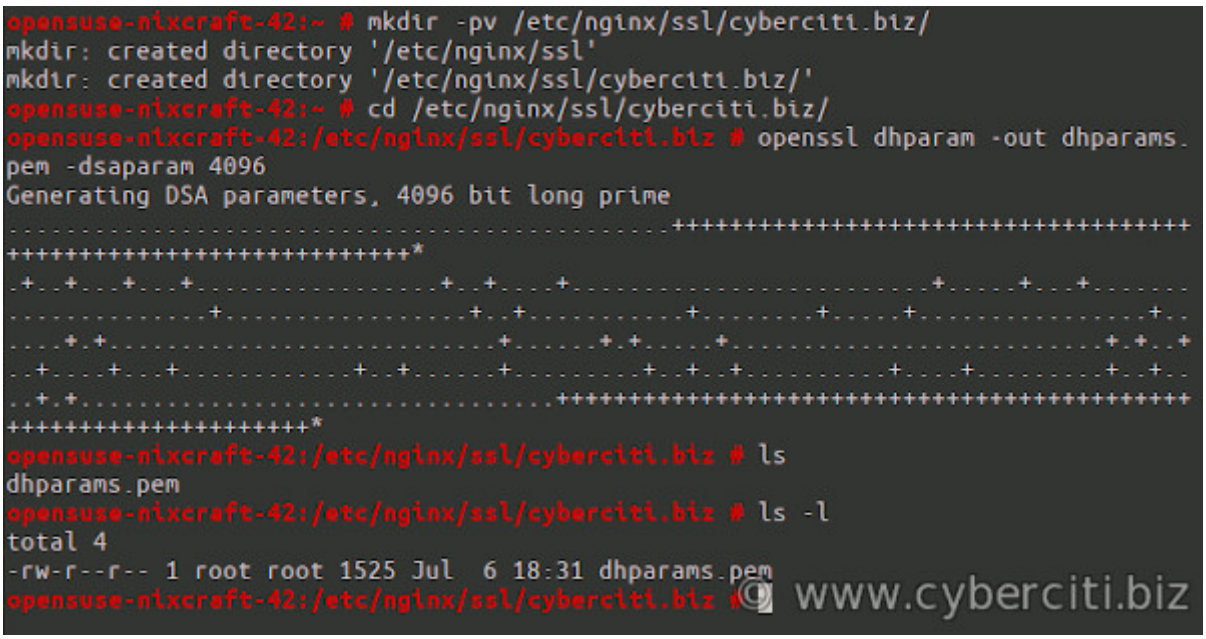
ファイルを保存して閉じます。次のようにnginx のセットアップをテストし、nginx サーバーをリロードします。

```
# nginx -t && systemctl restart nginx.service
```

ステップ 4 – dhparam.pem ファイルを作成する

openssl コマンドを使用して、次のように Diffie-Hellman 鍵交換ファイルを作成する必要があります。

```
# mkdir -pv /etc/nginx/ssl/cyberciti.biz/
# cd /etc/nginx/ssl/cyberciti.biz/
# openssl dhparam -out dhparams.pem -dsaparam 4096
# ls -l
```



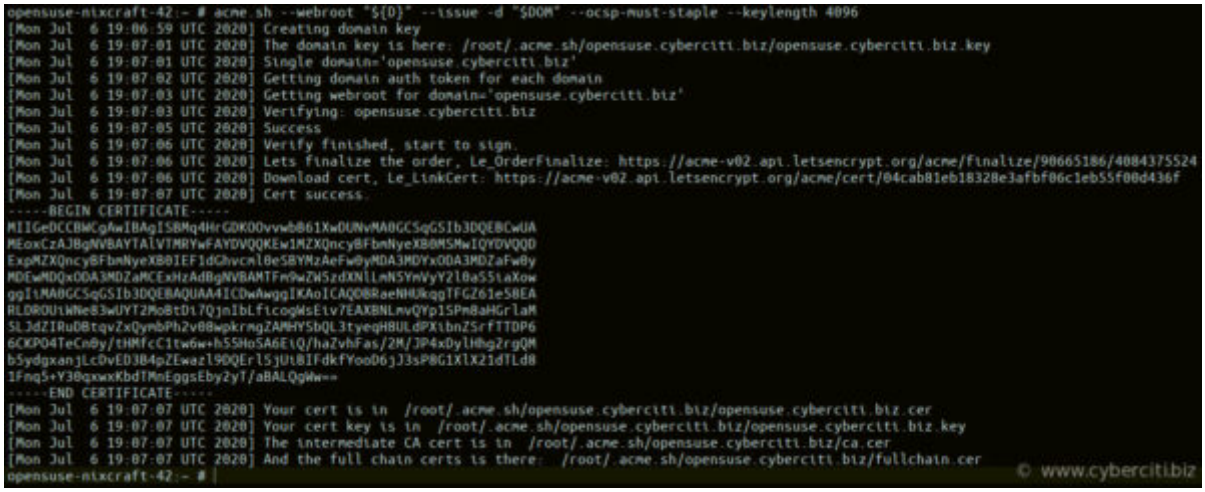
ステップ 5 – ドメインの証明書を取得する

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="opensuse.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
```



スタンドアロンモードで証明書を発行する

```
# DOM="opnesuse.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 フォルダーを指定します。/.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.cybercit.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 はオフです。

    # 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" >
```

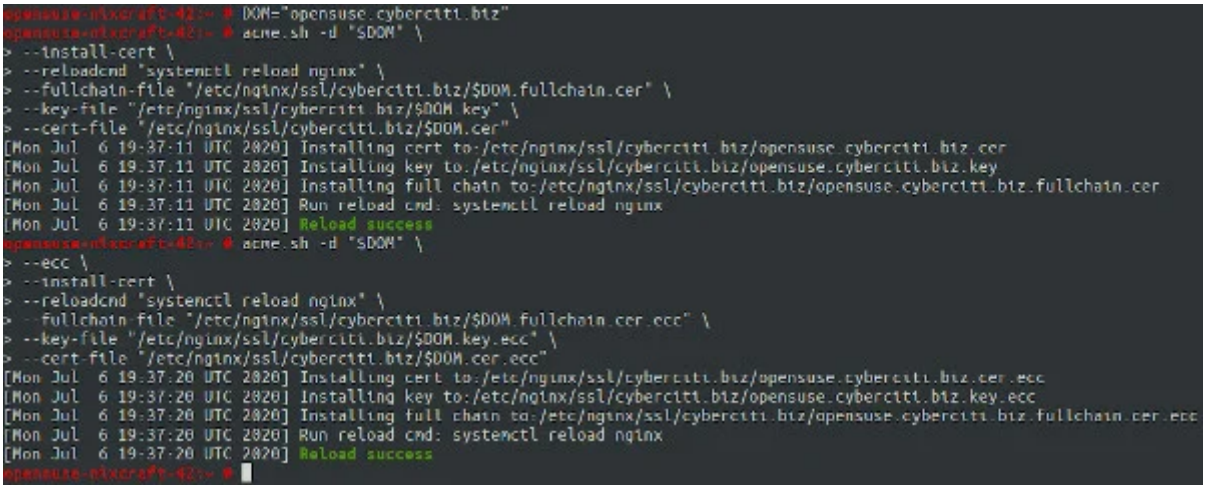
```
< head >
< title > OpenSUSE.Cyberciti.Biz Nginx サーバー< / title >
< meta  charset = "utf-8"  / >
< meta  name = "viewport"  content = "width=device-width,initial-scale=1.0" >
< / head >
< body >
<article>
< h2 > Hello, World!< / h2>
< p >これは、OpenSUSE Linux 15.2 と無料の TLS 証明書を備えた Nginx を搭載したテスト サーバーです。< / 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"
```



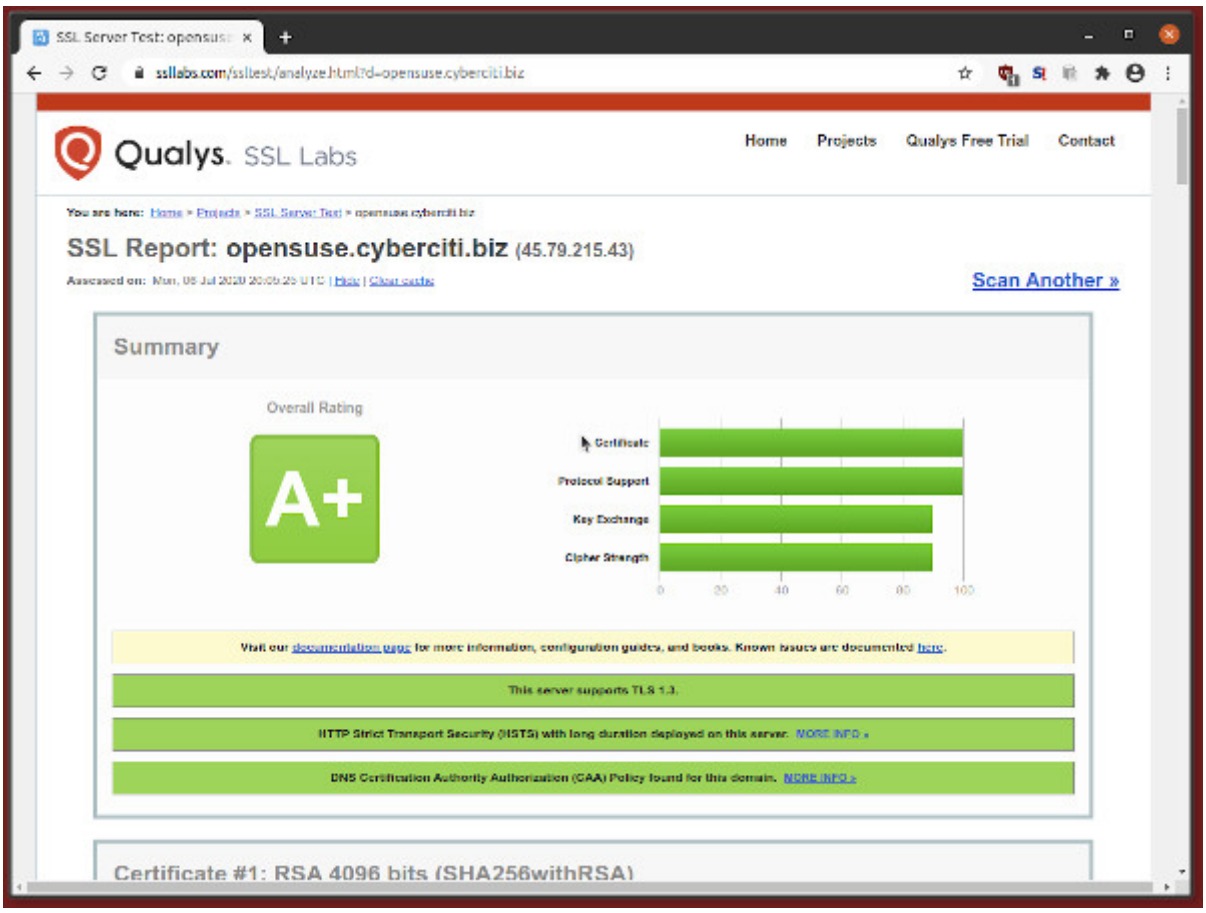
ステップ 8 – TCP ポート 443 [HTTPS ポート] を開きます。

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。

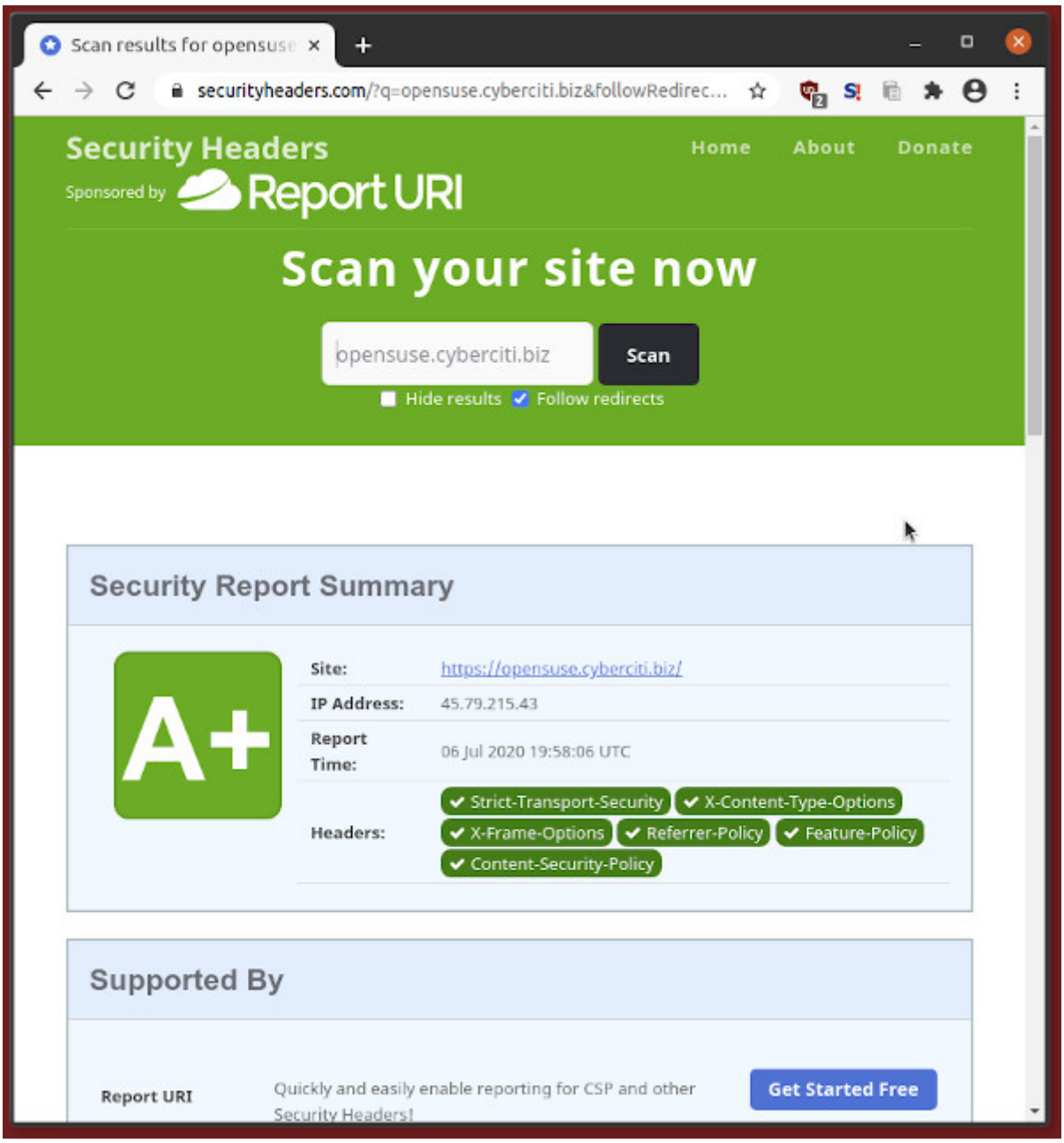
```
# firewall-cmd --zone=public --add-service=https
# firewall-cmd --zone=public --add-service=https --permanent
# firewall-cmd --list-services
# curl -I https://opensuse.cyberciti.biz/
```

ステップ 9 – テストする

SSL ラボのテスト:

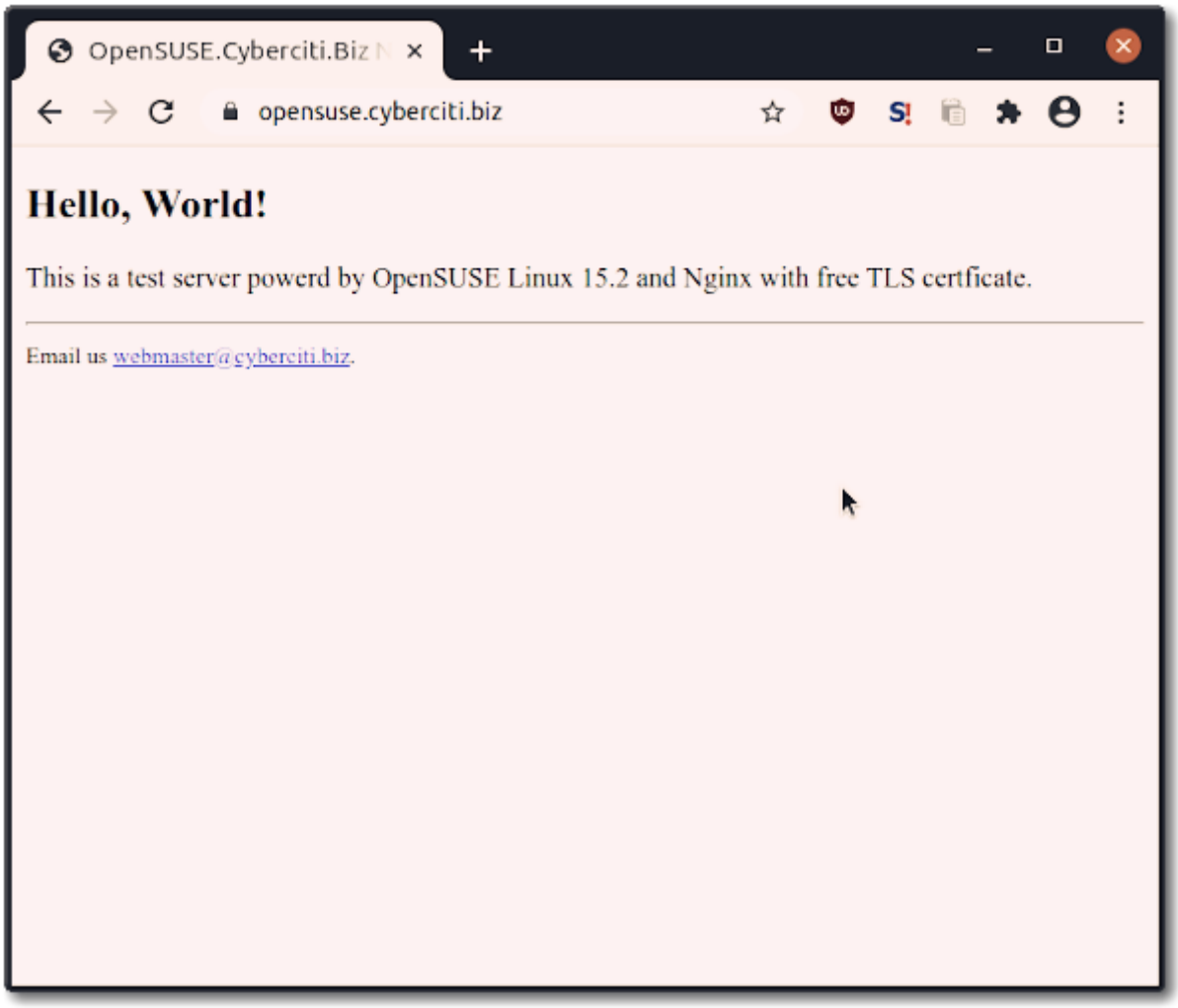


[セキュリティヘッダーの](#)テスト:



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

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。

コメント3 件... 1 つ追加

Esha2020年7月18日 4時15分

素晴らしくて、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 Gite2023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、4433 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 lsof パッケージをインストールして「zypper ps」エラーを解決する