Wprowadzanie do Cyberbezpieczeństwa

Instalacja i aktualizacja certyfikatu Let's Encrypt dla serwera WWW

Stanisław Nieradko 193044, Filip Dawidowski 193433, Bartłomiej Krawisz 193319, Krzysztof Nasuta 193328

Spis treści

1. Wstęp 1
2. Certyfikaty Let's Encrypt 1
3. Porównanie dostępnych dostawców certyfikatów 1
4. Metody autoryzacji domeny 2
5. Metody instalacji certyfikatu 3
6. Środowiska 4
7. Prezentacja praktyczna 5

1. Wstęp

Sprawozdanie z projektu dotyczącego instalacji i aktualizacji certyfikatu Let's Encrypt dla serwera WWW wykonanego w ramach przedmiotu Wprowadzenie do Cyberbezpieczeństwa. Przedstawione zostaną w nim kroki niezbędne do zainstalowania certyfikatu Let's Encrypt z użyciem różnych metod weryfikacji właściciela domeny. Zaprezentowany zostanie proces instalacji certyfikatu dla serwera `nginx` oraz `Caddy` w systemie operacyjnym `Ubuntu 22.04`. Wszystkie operacje zostały przeprowadzone na maszynie wirtualnej w chmurze Oracle Cloud Infrastructure (OCI).

2. Certyfikaty Let's Encrypt

Let's Encrypt to bezpłatny, automatyczny i wolny urząd certyfikacji (CA) działający dla pożytku publicznego. Jest to usługa dostarczana przez Internet Security Research Group (ISRG).

Zasady funkcjonowania fundacji Let's Encrypt:

- Bezpłatnie: Każdy właściciel domeny może użyć Let's Encrypt do uzyskania zaufanego certyfikatu bez żadnych opłat.
- Automatycznie: Oprogramowanie działające na serwerze może bezproblemowo wchodzić w interakcję z Let's Encrypt, aby uzyskać certyfikat, bezpiecznie skonfigurować go do użytku oraz automatycznie zająć się odnowieniem.
- Bezpiecznie: Let's Encrypt spełnia funkcję platformy do doskonalenia najlepszych praktyk zabezpieczeń TLS, zarówno po stronie CA, jak i pomagając operatorom witryn poprawnie zabezpieczyć swoje serwery.
- Otwarcie: Wszystkie wydane lub cofnięte certyfikaty będą publicznie rejestrowane i dostępne dla każdego do wglądu.
- Wolnie: Protokół automatycznego wydawania oraz odnawiania jest opublikowany jako wolny standard, który każdy może zastosować.
- Wspólnie: Tak jak podstawowe protokoły internetowe, Let's Encrypt to wspólny wysiłek na rzecz społeczności pozostający poza kontrolą jakiejkolwiek organizacji.

3. Porównanie dostępnych dostawców certyfikatów

3.1. Let's Encrypt

- Darmowe certyfikaty (90 dni)
- Automatyczne odnawianie (certbot, caddy i inne narzędzia)
- Wsparcie dla wildcardów (tylko DNS-01)

- Duża społeczność i wsparcie

3.2. Płatne certyfikaty

- Dłuższy termin ważności (do 2 lat, chociaż niektóre przeglądarki ograniczają do 1 roku)
- Możliwość wykupienia certyfikatów Organizational Validation (OV) i Extended Validation (EV)
- Wsparcie techniczne

3.3. ZeroSSL

- Zarówno darmowe, jak i płatne certyfikaty
- Pełna konsola oraz REST API do zarządzania certyfikatami
- Monitorowanie certyfikatów SSL

3.4. Cloudflare

- Zarówno darmowe, jak i płatne certyfikaty (większe możliwości konfiguracji szyfrowania oraz wielopoziomowe domeny)
- Zintegrowane z usługami Cloudflare (popularny CDN, firewall i inne usługi)
- Łatwa i natychmiastowa konfiguracja dla użytkowników Cloudflare'a

4. Metody autoryzacji domeny

Przy generowaniu certyfikatu Let's Encrypt wymagane jest potwierdzenie, że domena, dla której certyfikat chcemy wygenerować, należy do nas. Możliwe jest to poprzez spełnienie jednego z warunków opisanych w standardzie ACME (Automated Certificate Management Environment). Obecnie wspierane są trzy metody autoryzacji:

- `http-01`
- `dns-01`
- `tls-alpn-01`

Metoda	Adres IP	Nazwa hosta	Obsługa wildcardów
`http-01`	✓	✓	✗
`dns-01`	✗	✓	✓
`tls-alpn-01`	✓	✓	✗

W przeszłości istniała również metoda `tls-sni-01`, która została wycofana z użycia w 2019 roku.

4.1. `HTTP-01`

- Metoda ta jest obecnie najczęściej stosowaną metodą autoryzacji. Polega na umieszczeniu pliku z wygenerowanym przez Let's Encrypt kodem w odpowiednim katalogu na serwerze, który jest dostępny z zewnątrz.
- Let's Encrypt sprawdza, czy plik jest dostępny pod adresem `http://<TWOJA_DOMENA>/.well-known/acme-challenge/<TOKEN>`. Jeśli serwer WWW zwróci odpowiedni kod, jest to potwierdzenie, że domena należy do osoby, która chce wygenerować certyfikat.
- Metoda `HTTP-01` wymaga użycia portu 80 na serwerze, na którym chcemy wygenerować certyfikat.
- Metoda ta jest prosta w implementacji oraz szybka, gdyż nie wymaga żadnych dodatkowych konfiguracji DNS. Potrzebujemy jednak dostępu do serwera HTTP obsługującego naszą domenę.
- Metody tej nie można użyć, aby wygenerować certyfikat wildcard. W przypadku kilku serwerów, każdy z nich musi zwracać ten sam kod.

4.2. `DNS-01`

- Metoda ta polega na dodaniu rekordu TXT do DNS domeny, dla której chcemy wygenerować certyfikat. Rekord ten zawiera, podobnie jak w przypadku `HTTP-01`, wygenerowany przez Let's Encrypt kod.
- Let's Encrypt sprawdza, czy rekord TXT `_acme-challenge.<TWOJA_DOMENA>` zawiera odpowiedni kod. Jeśli tak jest, mamy potwierdzenie, że domena należy do osoby, która chce wygenerować certyfikat.
- Metoda `DNS-01` wymaga dostępu do konfiguracji DNS domeny, dla której chcemy wygenerować certyfikat. Utrudnia to automatyzację procesu generowania certyfikatów. Dostawca DNS musi udostępniać odpowiednie API. Zalecane jest używanie uwierzytelniania API o ograniczonych uprawnieniach bądź walidacja DNS z osobnego serwera, a następnie skopiowanie certyfikatu na serwer.
- Metoda ta jest wolniejsza od `HTTP-01`, gdyż wymaga czasu propagacji rekordów DNS. Jest to jednak jedyna metoda, która pozwala na generowanie certyfikatów wildcard. W przypadku kilku serwerów, wystarczy jedna konfiguracja DNS.

4.3. `TLS-SNI-01`

- Metoda przestarzała
- Do 2019 roku jedną z metod autoryzacji był `TLS-SNI-01`. Polegała ona na przekazaniu przez Let's Encrypt serwerowi specjalnego zapytania TLS, które zawierało wygenerowany przez Let's Encrypt kod. Serwer musiał zwrócić ten sam kod, aby potwierdzić, że domena należy do osoby, która chce wygenerować certyfikat.
- Metoda została wycofana z użycia w 2019 roku i zastąpiona przez `TLS-ALPN-01` z powodu niewystarczającego poziomu bezpieczeństwa.

4.4. `TLS-ALPN-01`

- Polega na przekazaniu przez Let's Encrypt serwerowi specjalnego zapytania TLS, które zawiera wygenerowany przez Let's Encrypt kod. Serwer musi zwrócić ten sam kod, aby potwierdzić, że domena należy do osoby, która chce wygenerować certyfikat.
- Metoda ta jest rzadko stosowana. Nie jest obsługiwana przez Apache, Nginx ani Certbot. Jednym z nielicznych narzędzi, które wspierają tę metodę, jest Caddy.
- Zaletą tej metody jest brak konieczności dostępu do portu 80. Cały proces odbywa się na warstwie TLS.
- Metoda ta, podobnie jak `HTTP-01`, nie pozwala na generowanie certyfikatów wildcard. W przypadku kilku serwerów, każdy z nich musi zwracać ten sam kod.

5. Metody instalacji certyfikatu

5.1. Manualna

- Manualna instalacja certyfikatu polega na ręcznym wygenerowaniu certyfikatu Let's Encrypt, a następnie skonfigurowaniu serwera WWW, aby używał tego certyfikatu.
- Do generowania certyfikatów można użyć narzędzi takich jak Certbot lub ZeroSSL. Następnie należy skonfigurować serwer WWW, aby używał wygenerowanego certyfikatu oraz pamiętać o regularnym odnawianiu certyfikatów.
- Metoda ta jest niewygodna i czasochłonna, dlatego zaleca się automatyzację procesu generowania certyfikatów (ACME) ale ma swoje zastosowanie w przypadku systemów, które nie są obsługiwane przez narzędzia automatyzujące.

5.2. Certbot

- Certbot to prosty w użyciu program, który automatyzuje proces uzyskiwania certyfikatów Let's Encrypt. Automatycznie konfiguruje serwer WWW (np. apache, nginx), aby używał nowego

certyfikatu, a także automatycznie odnawia certyfikaty, gdy zbliżają się do wygaśnięcia. Certbot wspiera autoryzację `http-01` oraz `dns-01`.
- Certbot jest dostępny na większość popularnych systemów operacyjnych, takich jak Linux, Windows oraz macOS. Dostępne są również wtyczki do popularnych serwerów WWW, takich jak Apache i Nginx, umożliwiające automatyczne przystosowanie konfiguracji używanego serwera WWW.

5.3. Caddy
- Caddy to serwer WWW, który automatycznie obsługuje certyfikaty Let's Encrypt. Wystarczy dodać konfigurację serwera WWW do pliku Caddyfile, a Caddy automatycznie wygeneruje certyfikat Let's Encrypt i skonfiguruje serwer WWW, aby używał tego certyfikatu.
- Caddy obsługuje autoryzację `http-01` oraz `tls-alpn-01`. W przypadku autoryzacji `http-01`, Caddy automatycznie dodaje odpowiednią konfigurację do pliku Caddyfile, aby umożliwić Let's Encrypt weryfikację domeny.
- Caddy jest dostępny na systemy operacyjne Linux, Windows oraz macOS. Oprócz obsługi certyfikatów Let's Encrypt, Caddy oferuje wiele innych funkcji, takich jak load balancing, obsługa protokołu HTTP/2, eksperymentalna obsługa protokołu QUIC czy możliwość konfiguracji poprzez API, dzięki czemu jest to ciekawa alternatywa dla bardziej popularnych serwerów WWW, takich jak Apache czy Nginx.

5.4. Cert-Manager
- cert-manager to narzędzie do zarządzania certyfikatami w środowiskach opartych na Kubernetes. Automatycznie generuje certyfikaty Let's Encrypt dla aplikacji działających w klastrze Kubernetes, a także automatycznie odnawia certyfikaty, gdy zbliżają się do wygaśnięcia.
- cert-manager obsługuje autoryzację `http-01` oraz `dns-01`. W przypadku obu metod, cert-manager automatycznie dodaje odpowiednie zasoby do klastra Kubernetes, aby umożliwić Let's Encrypt weryfikację domeny.

5.5. Dostawcy hostingu
- Wiele firm hostingowych oferuje integrację z Let's Encrypt. W takim przypadku proces generowania certyfikatu jest zautomatyzowany, a użytkownik nie musi się martwić o konfigurację serwera WWW. Jedną z wad takiego rozwiązania jest ograniczona kontrola nad konfiguracją oraz limit na ilość certyfikatów, które można wygenerować.

6. Środowiska
Z uwagi na ograniczenia nałożone przez organizację Let's Encrypt, zaleca się używanie certyfikatów wystawionych w środowisku testowym do testowania automatyzacji procesu generowania certyfikatów.

6.1. Staging
- Certyfikaty wystawione w środowisku testowym są podpisane przez inny certyfikat root, co sprawia, że nie są one uznawane przez przeglądarki internetowe.
- Wystawianie certyfikatów w tym środowisku podlega niższym limitom, co pozwala na testowanie automatyzacji procesu wystawiania certyfikatów, bez dużego ryzyka zablokowania dostępu do usługi Let's Encrypt z powodu przekroczenia ograniczeń.
- W przypadku użycia certbot, aby wygenerować certyfikat staging wystarczy dodać flagę `--staging`.

6.2. Production
- Środowisko produkcyjne to środowisko aplikacji dla użytkowników końcowych. Certyfikaty wystawione w tym środowisku są uznawane przez przeglądarki internetowe.

- Wystawianie certyfikatów w środowisku produkcyjnym podlega limitom nałożonym przez organizację Let's Encrypt takimi jak ilość certyfikatów na zarejestrowaną domenę lub ilość zamówień certyfikatów na godzinę. W przypadku przekroczenia limitów, dostęp do usługi może zostać zablokowany na określony czas.
- Certbot domyślnie generuje certyfikaty w środowisku produkcyjnym.

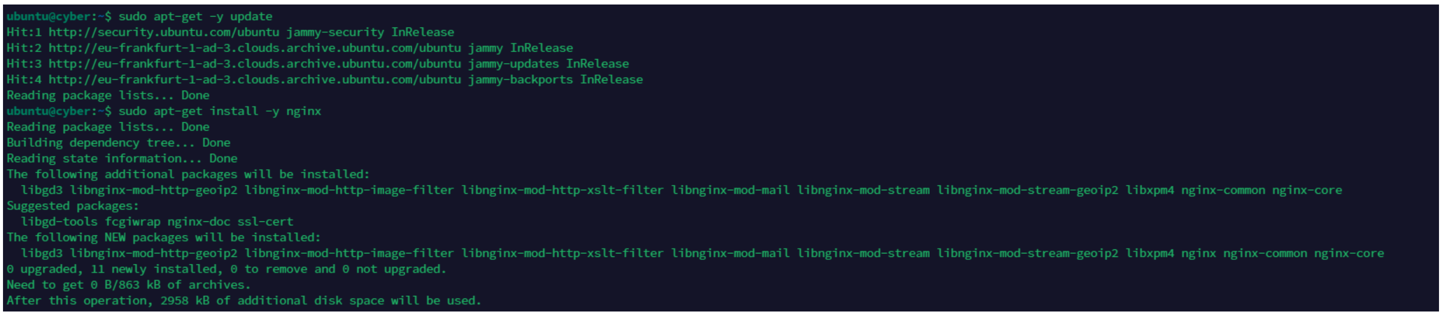
7. Prezentacja praktyczna

7.1. nginx

Przykładowa instalacja certyfikatu Let's Encrypt dla serwera WWW `nginx` w systemie operacyjnym `Ubuntu 22.04` z użyciem autoryzacji `http-01`. Komendy należy wykonać jako użytkownik z uprawnieniami administratora. W przypadku użycia innej dystrybucji systemu Linux niż Ubuntu, należy dostosować komendy do używanej dystrybucji.

- Instalacja `nginx` z repozytorium Ubuntu

```
apt-get -y update
apt-get install -y nginx
```



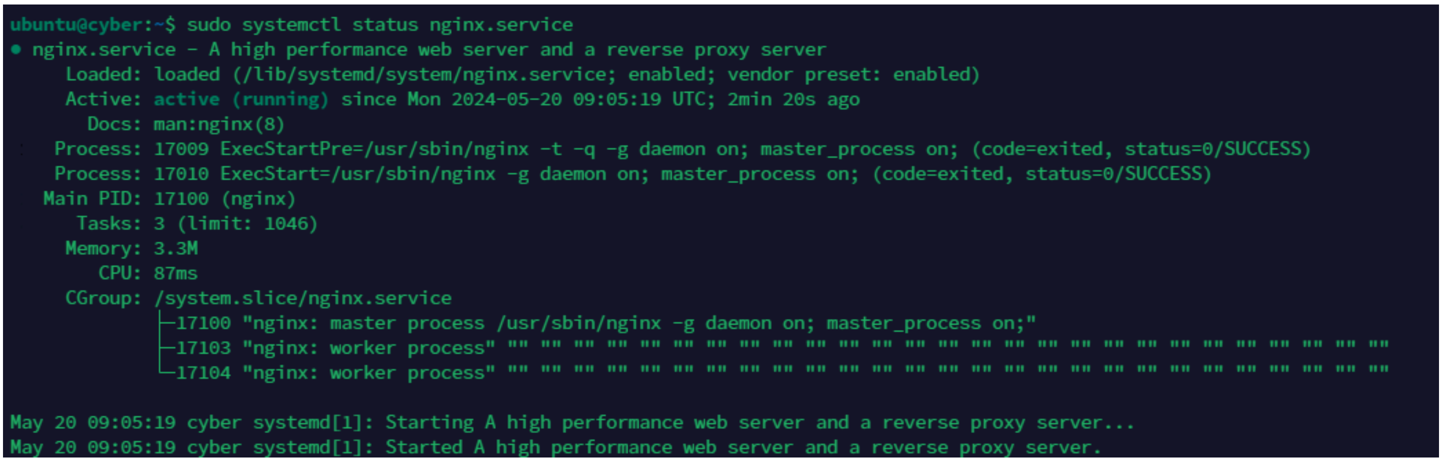
```
ubuntu@cyber:~$ sudo apt-get -y update
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:2 http://eu-frankfurt-1-ad-3.clouds.archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://eu-frankfurt-1-ad-3.clouds.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://eu-frankfurt-1-ad-3.clouds.archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
ubuntu@cyber:~$ sudo apt-get install -y nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libgd3 libnginx-mod-http-geoip2 libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream libnginx-mod-stream-geoip2 libxpm4 nginx-common nginx-core
Suggested packages:
  libgd-tools fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
  libgd3 libnginx-mod-http-geoip2 libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream libnginx-mod-stream-geoip2 libxpm4 nginx nginx-common nginx-core
0 upgraded, 11 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/863 kB of archives.
After this operation, 2958 kB of additional disk space will be used.
```

- Włączenie serwera `nginx`

```
systemctl enable nginx
systemctl start nginx
```

- Sprawdzenie statusu serwera `nginx`

```
systemctl status nginx
```



```
ubuntu@cyber:~$ sudo systemctl status nginx.service
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2024-05-20 09:05:19 UTC; 2min 20s ago
       Docs: man:nginx(8)
    Process: 17009 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
    Process: 17010 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Main PID: 17100 (nginx)
      Tasks: 3 (limit: 1046)
     Memory: 3.3M
        CPU: 87ms
     CGroup: /system.slice/nginx.service
             ├─17100 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             ├─17103 "nginx: worker process" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ""
             └─17104 "nginx: worker process" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ""

May 20 09:05:19 cyber systemd[1]: Starting A high performance web server and a reverse proxy server...
May 20 09:05:19 cyber systemd[1]: Started A high performance web server and a reverse proxy server.
```

- Instalacja `certbot` z repozytorium Ubuntu

Dodatkowo instalujemy pakiet `python3-certbot-nginx`, który pozwoli na automatyczną konfigurację serwera `nginx` do użycia certyfikatu Let's Encrypt.

```
apt-get install -y certbot python3-certbot-nginx
```

- Generowanie certyfikatu Let's Encrypt

```
certbot --nginx
```

Aby wygenerować certyfikat Let's Encrypt, należy podać adres e-mail, zaakceptować regulamin oraz zdecydować, czy chcemy otrzymywać informacje o nowościach. Następnie należy wybrać domenę, dla której chcemy wygenerować certyfikat. Po zakończeniu procesu, certyfikat zostanie zainstalowany na serwerze `nginx`.

Jeśli chcemy zainstalować certyfikat w środowisku testowym, należy dodać flagę `--staging` do komendy `certbot --nginx`. Aby utworzyć certyfikat bez podawania adresu e-mail, należy dodać flagę `--register-unsafely-without-email`.

```
ubuntu@cyber:~$ sudo certbot --nginx --register-unsafely-without-email --staging
Saving debug log to /var/log/letsencrypt/letsencrypt.log

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.4-April-3-2024.pdf. You must agree in
order to register with the ACME server. Do you agree?
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(Y)es/(N)o: Y
Account registered.
Please enter the domain name(s) you would like on your certificate (comma and/or
space separated) (Enter 'c' to cancel): [redacted]
Requesting a certificate for [redacted]

Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/[redacted]/fullchain.pem
Key is saved at:         /etc/letsencrypt/live/[redacted]/privkey.pem
This certificate expires on 2024-08-18.
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 demo.nieradko.com to /etc/nginx/sites-enabled/default
Congratulations! You have successfully enabled HTTPS on https://[redacted]
```

7.2. Caddy

Przykładowa konfiguracja serwera WWW `Caddy` z automatycznym generowaniem certyfikatu Let's Encrypt.

- Instalacja `Caddy` z oficjalnej strony

Pobieranie pliku binarnego `caddy` oraz nadanie uprawnień do wykonywania. Inne metody instalacji dostępne są na stronie Caddy.

```
wget "https://caddyserver.com/api/download?os=linux&arch=amd64" -O caddy
chmod +x caddy
```

- Tworzenie pliku konfiguracyjnego `Caddyfile`

```
# Caddyfile
example.com {
    root * /var/www/html
    file_server
    tls demo@example.com {
        #ca https://acme-staging-v02.api.letsencrypt.org/directory # staging
        ca https://acme-v02.api.letsencrypt.org/directory # production
    }
}
```

- Uruchomienie serwera `Caddy`

Plik `Caddyfile` należy umieścić w katalogu, w którym znajduje się plik binarny `caddy`.

```
./caddy run
```

Po uruchomieniu serwera `Caddy`, certyfikat Let's Encrypt zostanie automatycznie wygenerowany i zainstalowany na serwerze.