# SSL Certificate Experiment

**0866007 胡孝德**

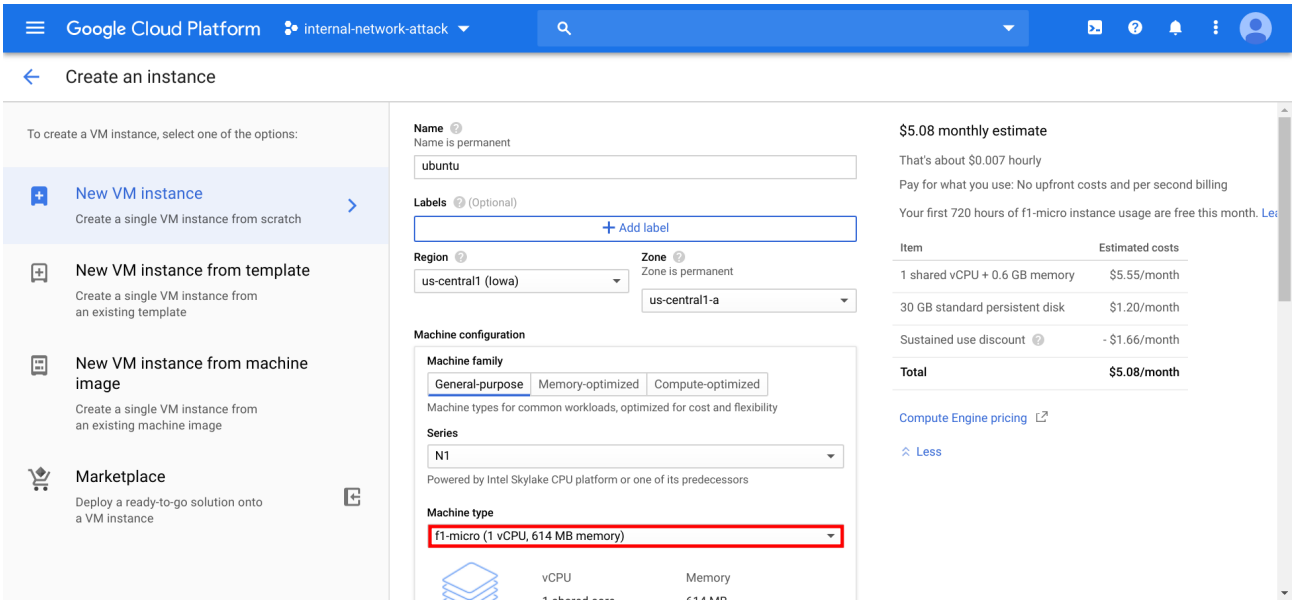
---

## 1. Set up a web server supporting HTTPS with perfect forward secrecy.

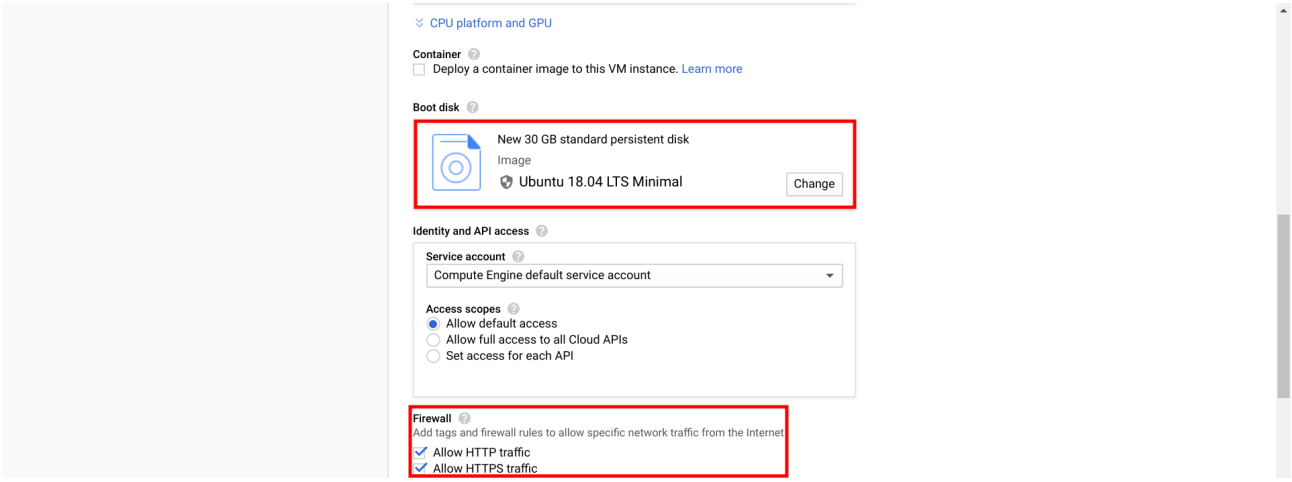
### 1.1 Create VM on Google Cloud Platform

規格如下，除了少量網路費用之外，其餘永久免費。

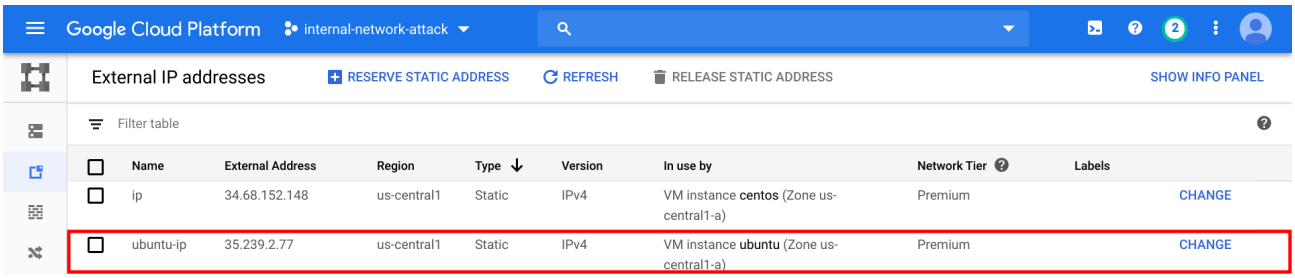
| Item | Setting |
| --- | --- |
| Machine | f1-micro (1 shared vCPU, 614 MB memory) |
| Disk | 30 GB HDD |
| Image | Ubuntu 18.04 TLS Minimal |
| Firewall | Allow HTTP, HTTPS traffic |
| IP | 35.239.2.77 |



選擇 f1-micro Machine




設定 Disk Size, Image, Firewall




設定永久性的 IP

安裝一些常用套件，與時區的設定

```
sudo apt update
sudo apt install vim less bash-completion \
                 man-db policykit-1
sudo timedatectl set-timezone Asia/Taipei
```

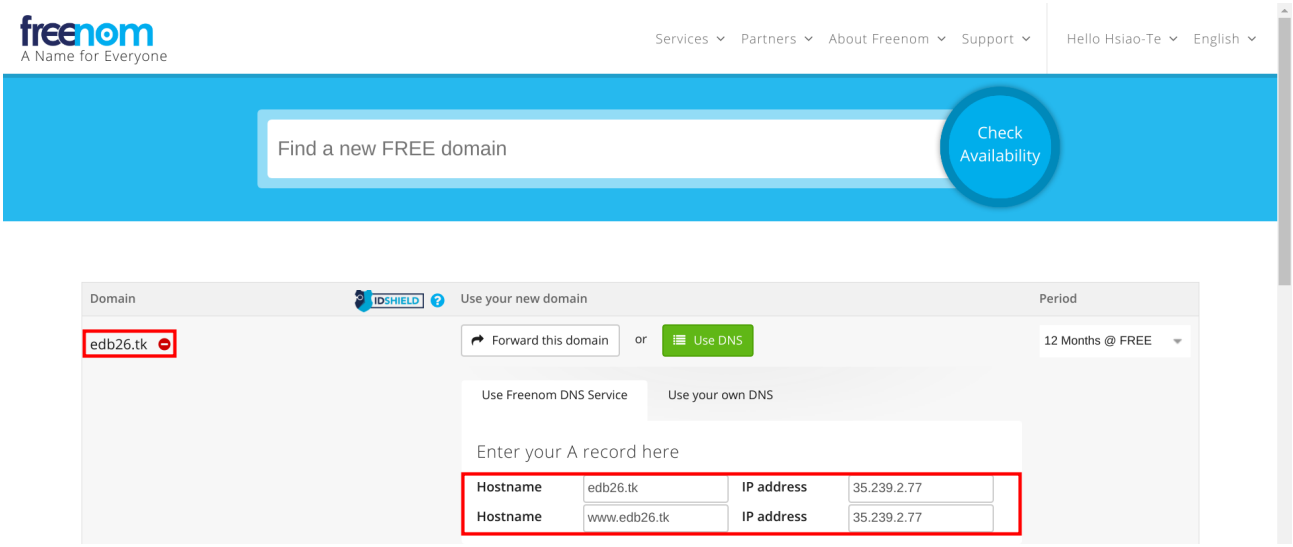
## 1.2 Install Nginx Web Server

```
sudo apt update
sudo apt install nginx
```


安裝完成後，便可由瀏覽器連線到該伺服器

## 1.3 Register a New Domain at freenom.com

註冊域名 **edb26.tk**，並設定 DNS record，使其指向 VM 的 IP (35.239.2.77)。本域名前 12 個月免收費。




註冊域名，同時設定 DNS


設定完成後，便可由該域名連線到伺服器

## 1.4 Configure Nginx with Let's Encrypt Certificate

按照官方網站說明來申請憑證。在申請的過程當中，同時還可以設定 HTTP Redirection, 將 HTTP 連線導向 HTTPS URL。

1. 將 Certbot PPA 新增到 apt repositories 當中。

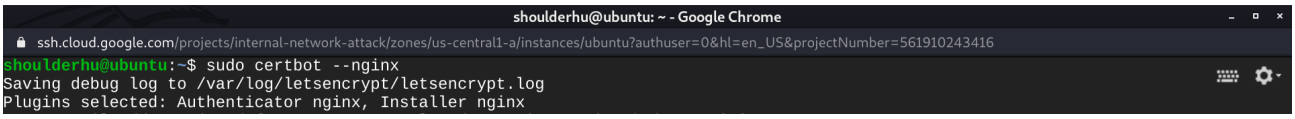
```
sudo apt update
sudo apt install software-properties-common
sudo add-apt-repository universe
sudo add-apt-repository ppa:certbot/certbot
```

2. 安裝 Certbot

```
sudo apt install certbot python-certbot-nginx
```

3. 執行 Cerbot，自動取得憑證，並修改 Nginx 設定檔。

```
sudo certbot --nginx
```



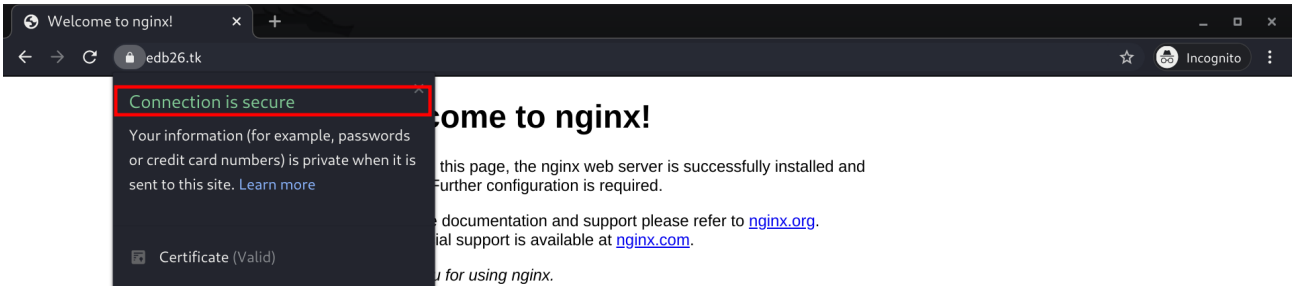

第三步驟節圖，Part 1

```
Enter email address (used for urgent renewal and security notices) (Enter 'c' to
cancel): shoulderhu@gmail.com

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf. You must
agree in order to register with the ACME server at
https://acme-v02.api.letsencrypt.org/directory
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(A)gree/(C)ancel: A

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Would you be willing 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
No names were found in your configuration files. Please enter in your domain
name(s) (comma and/or space separated)  (Enter 'c' to cancel): edb26.tk
Obtaining a new certificate
Performing the following challenges:
http-01 challenge for edb26.tk
Waiting for verification...
Cleaning up challenges
Deploying Certificate to VirtualHost /etc/nginx/sites-enabled/default

Please choose whether or not to redirect HTTP traffic to HTTPS, removing HTTP access.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

第三步驟節圖，Part 2

```
1: No redirect - Make no further changes to the webserver configuration.
2: Redirect - Make all requests redirect to secure HTTPS access. Choose this for
new sites, or if you're confident your site works on HTTPS. You can undo this
change by editing your web server's configuration.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Select the appropriate number [1-2] then [enter] (press 'c' to cancel): 2
Redirecting all traffic on port 80 to ssl in /etc/nginx/sites-enabled/default

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Congratulations! You have successfully enabled https://edb26.tk

You should test your configuration at:
https://www.ssllabs.com/ssltest/analyze.html?d=edb26.tk
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

IMPORTANT NOTES:
 - Congratulations! Your certificate and chain have been saved at:
   /etc/letsencrypt/live/edb26.tk/fullchain.pem
   Your key file has been saved at:
   /etc/letsencrypt/live/edb26.tk/privkey.pem
   Your cert will expire on 2020-07-05. To obtain a new or tweaked
   version of this certificate in the future, simply run certbot again
   with the "certonly" option. To non-interactively renew *all* of
   your certificates, run "certbot renew"
 - Your account credentials have been saved in your Certbot
   configuration directory at /etc/letsencrypt. You should make a
   secure backup of this folder now. This configuration directory will
   also contain certificates and private keys obtained by Certbot so
   making regular backups of this folder is ideal.
 - If you like Certbot, please consider supporting our work by:

   Donating to ISRG / Let's Encrypt:   https://letsencrypt.org/donate
   Donating to EFF:                    https://eff.org/donate-le
```

第三步驟節圖，Part 3

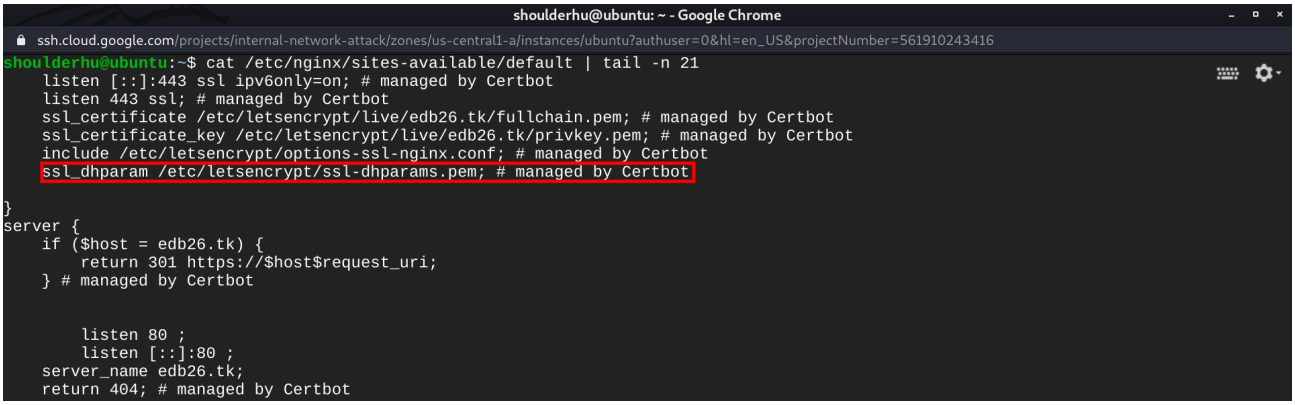



設定完成後，便可透過瀏覽器加密連線到伺服器

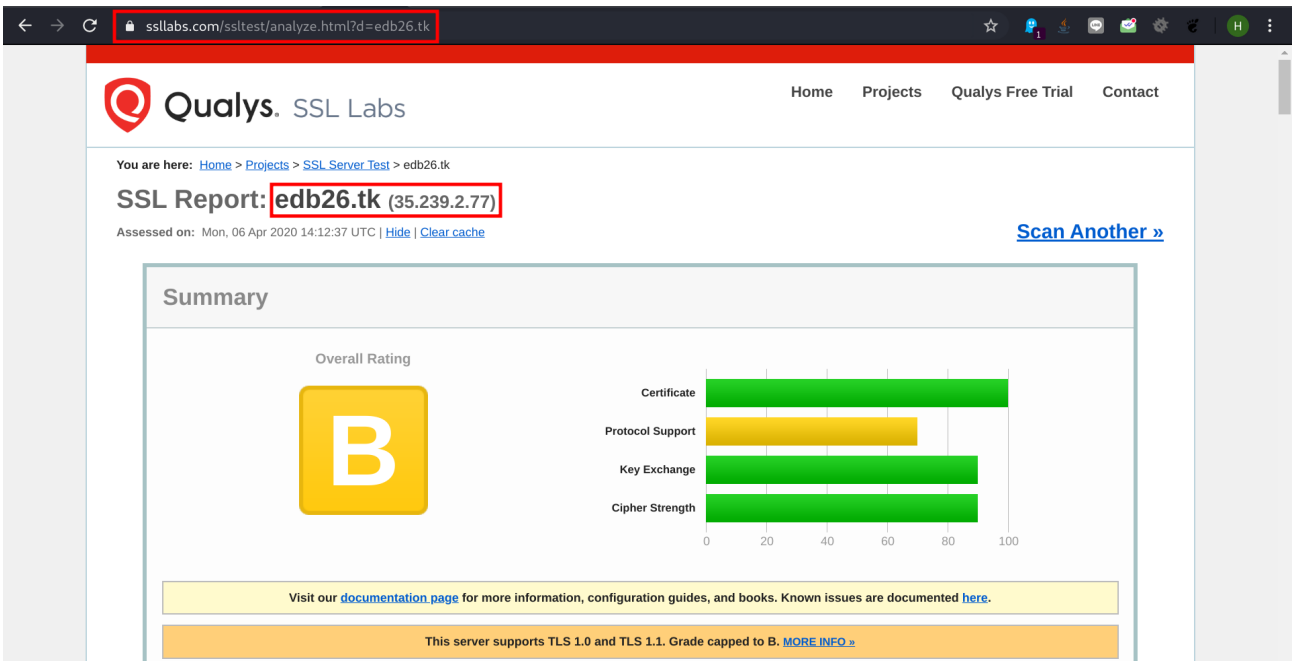
## 1.5 Check SSL Perfect Forward Secrecy

有些 Key agreement protocol 提供 perfect forward secrecy (PFS) feature，確保 session keys 不會因為 private key 被 compromised 而跟著被 compromised。

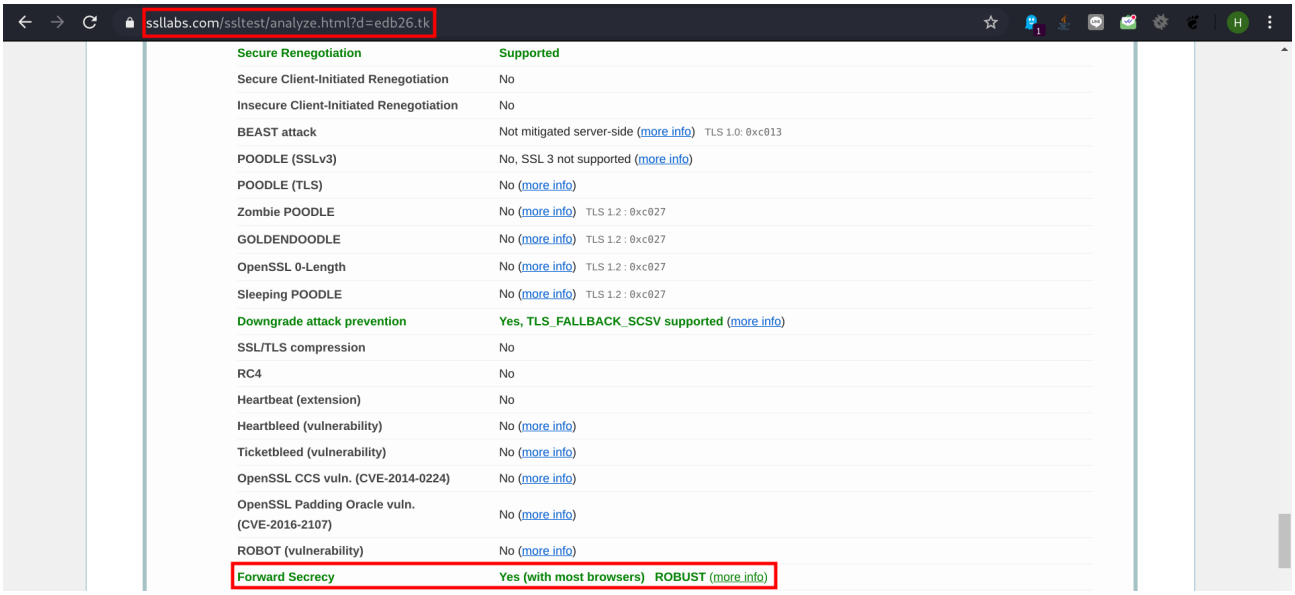
在 1.4 小節的自動設定當中便已完成該項設定，可從 Nginx 的設定檔中得知，如下圖所示。




除此之外，還可以透過一些網站如 SSL Labs 來檢測該網站是否有支援 Forward Secrecy，如下圖所示。




SSL Labs Report, Part 1




SSL Labs Report, Part 2

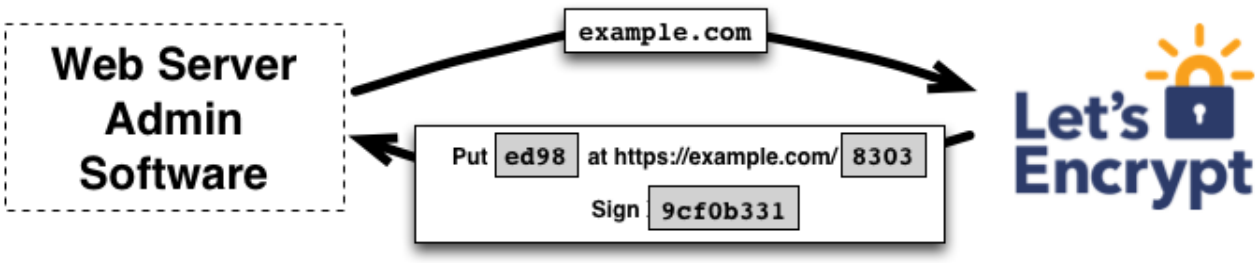
## 1.6 ACME protocol

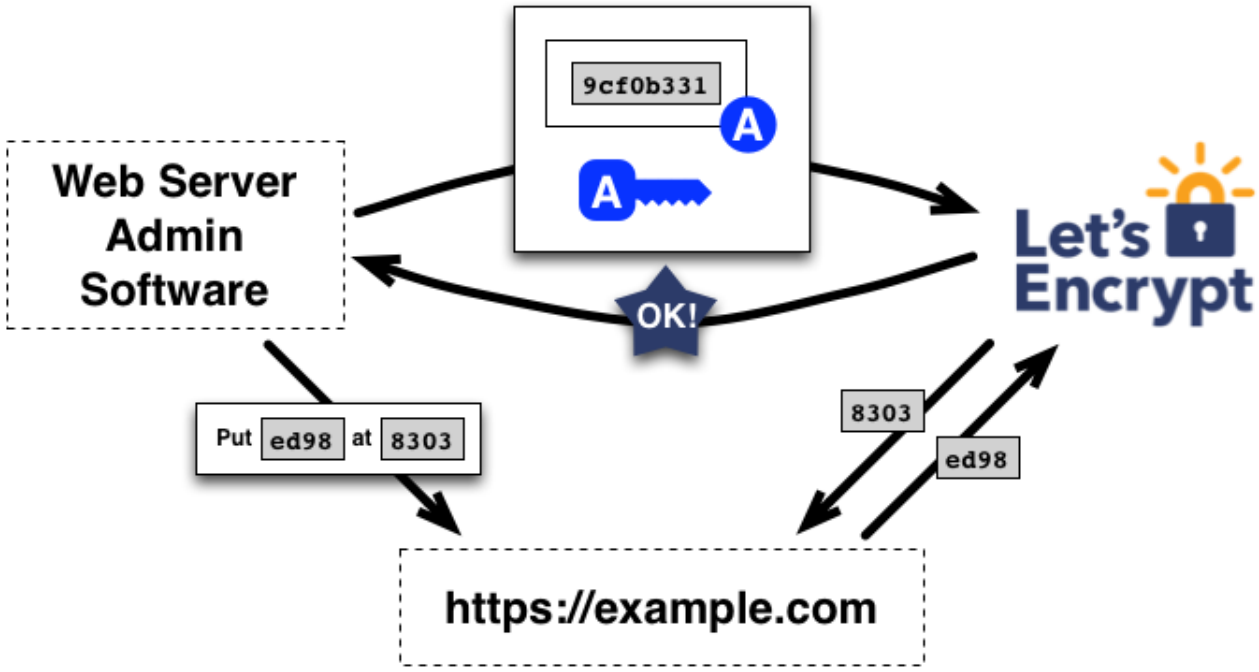
ACME 的目標是在沒有人為介入的情況下，讓伺服器取得憑證。為了達成這項目標，首先要在伺服器安裝一隻 Agent 程式，在上述的實驗當中，我們的 Agent 程式就是 Certbot。

Certbot 在第一次與 Let's Encrypt CA 做連線時，會產生一對授權金鑰 (authorization key)。這對金鑰在後續的憑證申請流程當中用來向 CA 驗證 Certbot 自己的身份時會用到。

一開始 Certbot 需要向 CA 證明該伺服器對網域的擁有權，可透過下列兩種方式。

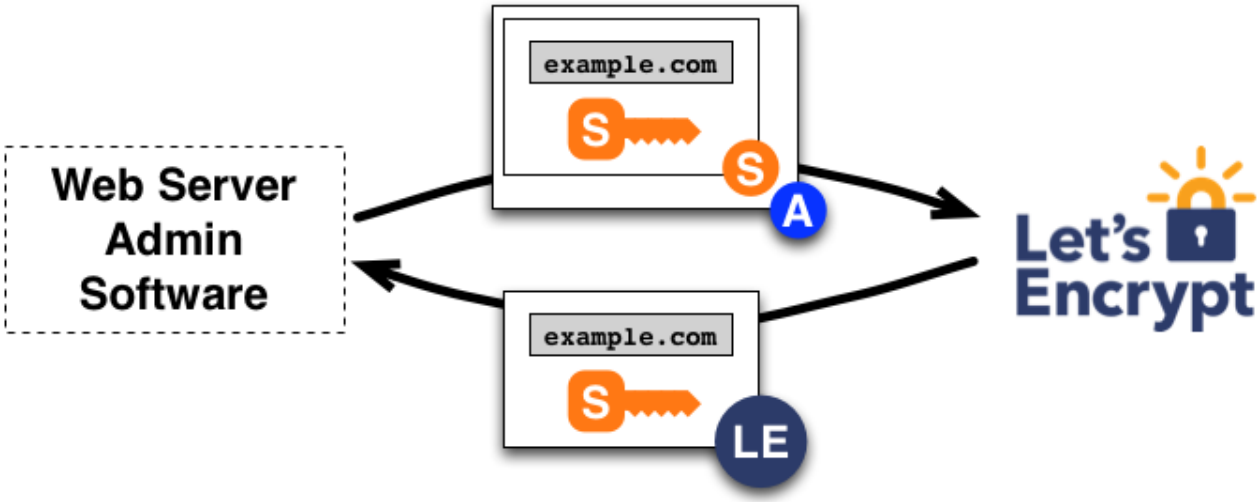
- 在 Name Server 上新增一筆 TXT record，內容由 CA 提供。然後 CA 再對該 TXT record 做查詢，確認內容的一致性。如果兩者相同，就表示你擁有網域的控制權。
- 在 Web Server 上新增一份檔案，其內容由 CA 指定。如果 CA 能成功下載那新增出來的檔案，就表示你對該網域具有控制權。除此之外，CA 還會給予一個 nonce 給 Cerbot，Certbot 要對該 nonce 用授權金鑰做簽章，然後回傳給 CA 來做驗證，確保執行上述動作的角色是正確的，如下圖所示。





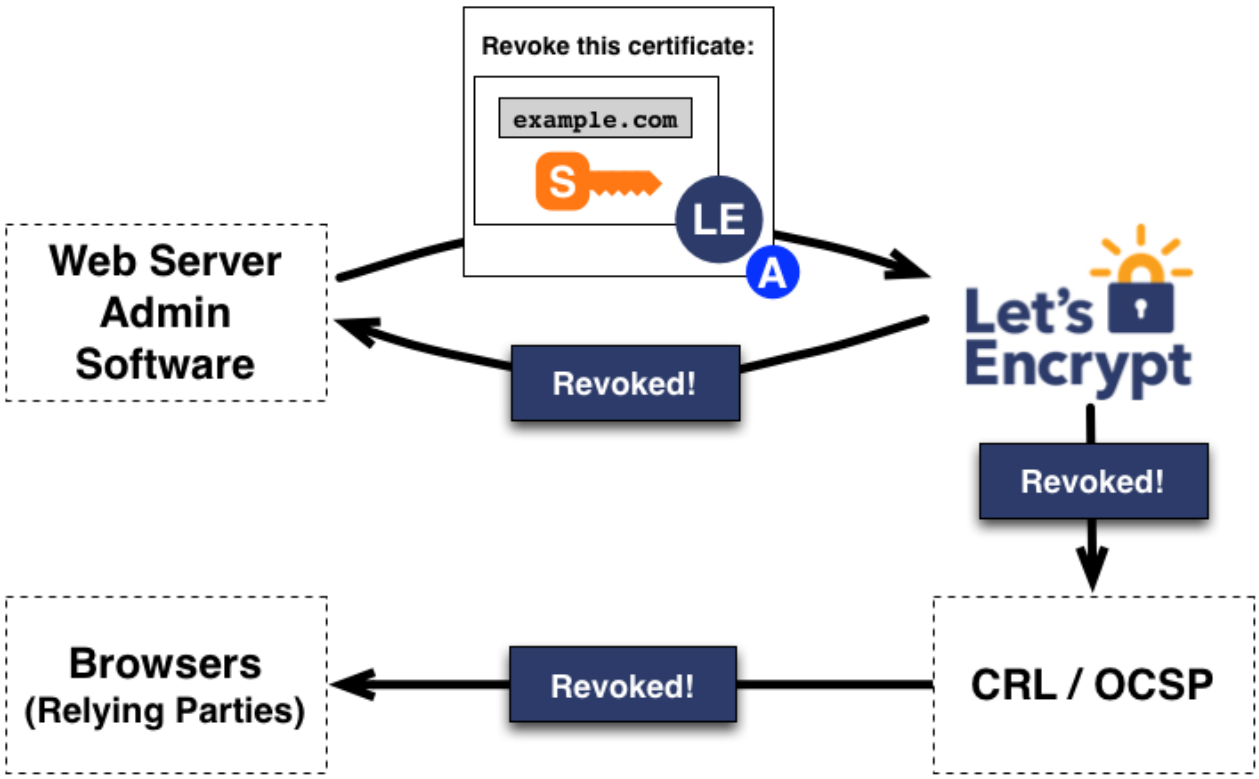

證明了網域的控制權後，Certbot 便有權利能幫該網域申請、更新、註銷憑證。

申請憑證的程序與原來一般申請憑證的流程大致上相同。Certbot 會先建立伺服器的金鑰對，然後是 Certificate Signing Request (CSR)，裡頭含有經過私鑰簽名後的資料與和私要對應的公鑰。除此之外，Certbot 還會用授權金鑰對 CSR 做簽章，讓 CA 得知 Certbot 已獲得授權。CA 收到之後，會先驗證這兩個簽章是否正確，然後才對 CSR 做簽章、產生出憑證，送回去給 Certbot。




註銷憑證的程序也是跟原來一般的註銷程序差不多，只是 Certbot 還需要在對註銷請求做簽章，讓 CA 得知 Certbot 已獲得授權。CA 驗證完成後，便會將註銷的訊息送到 OCSP (Online

Certificate Status Protocol) Server 上，讓所有人知道該憑證已備註消。




## 2. Self-signed certificate

建立 self-signed 憑證有兩種方式，一種是直接產生出一組 Key 與 Certificate。但是這種方式產生出的憑證，在透過瀏覽器連線時會出現警告訊息。另外一種是先建立出 Root CA，在透過 CA 來簽署自行產生出來的憑證，最後將 CA 的憑證加入到瀏覽器信任的 CA 列表當中。如此一來，該瀏覽器在對伺服器連線時，就不會出現警告。

本節的實驗將沿用上一小節的 VM, domain name。

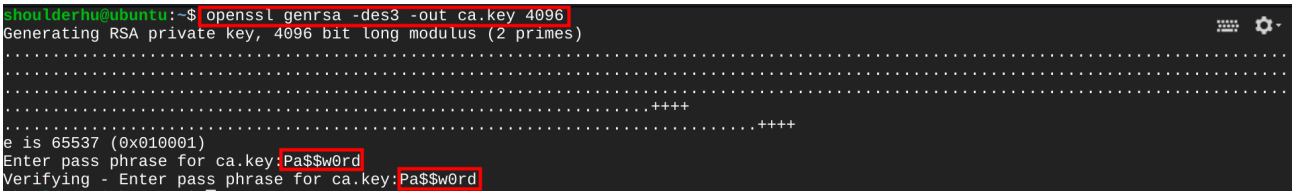
### 2.1 Create Root CA

1. 建立 Root CA 的 Key，用來簽署憑證用。這裡使用 **3DES** 對 key 做加密的動作，避免任何人取得 key 之後，任意使用。

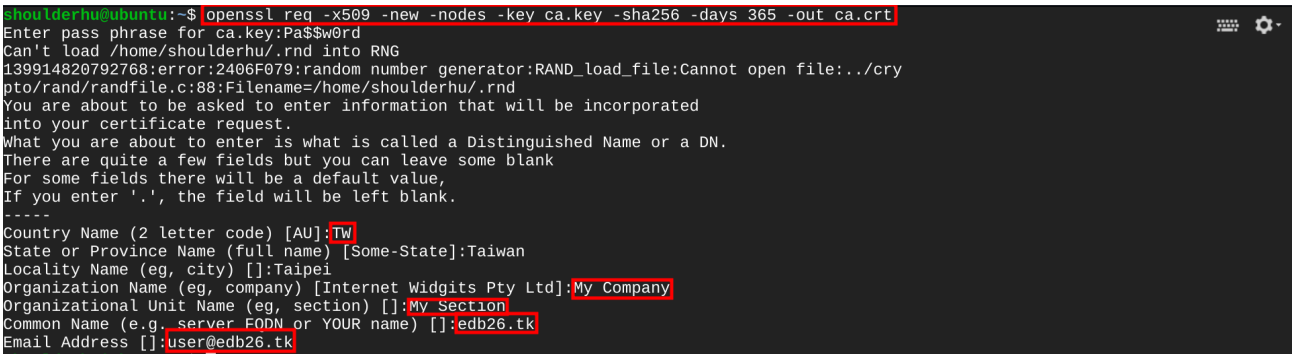
```
openssl genrsa -des3 -out ca.key 4096
```

2. 建立 Root CA 的 Certificate。

```
openssl req -x509 -new -nodes -key ca.key -sha256 -days 365 -out ca.crt
```




2.1.1




2.1.2

## 2.2 Create Server Certificate

1. 建立伺服器所需的 Key。

```
openssl genrsa -out edb26.key 2048
```

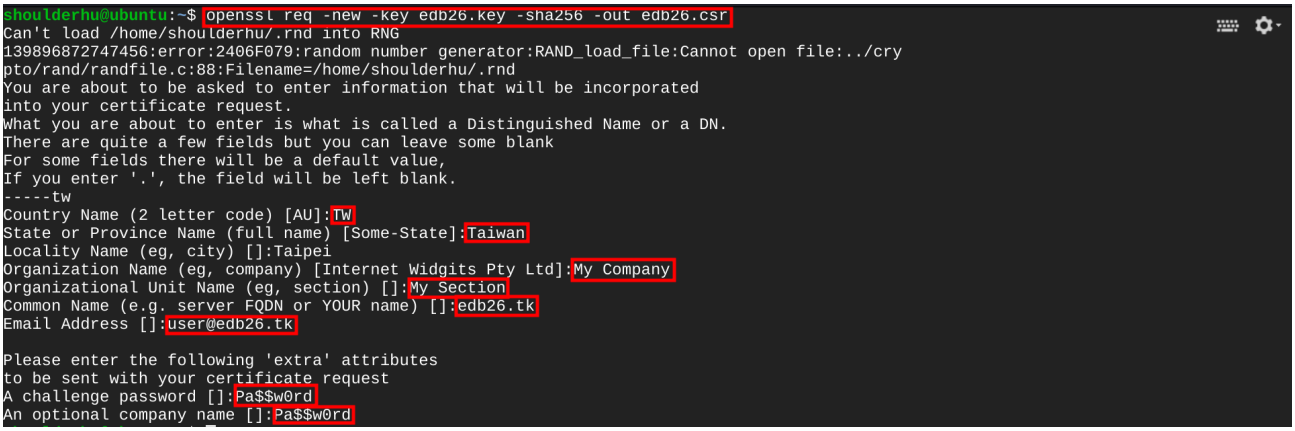
2. 建立伺服器所需的 Certificate Signing Request (CSR)。

```
openssl req -new -key edb26.key -sha256 -out edb26.csr
```

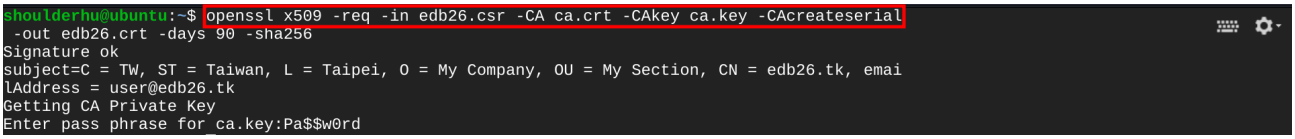
3. 用 Root CA 對 CSR 做簽署的動作，產出伺服器所需的憑證。

```
openssl x509 -req -in edb26.csr -CA ca.crt -CAkey ca.key -CAcreateseri
             -out edb26.crt -days 90 -sha256
```


2.2.1




2.2.2




2.2.3

## 2.3 Modify Nginx Configuration file

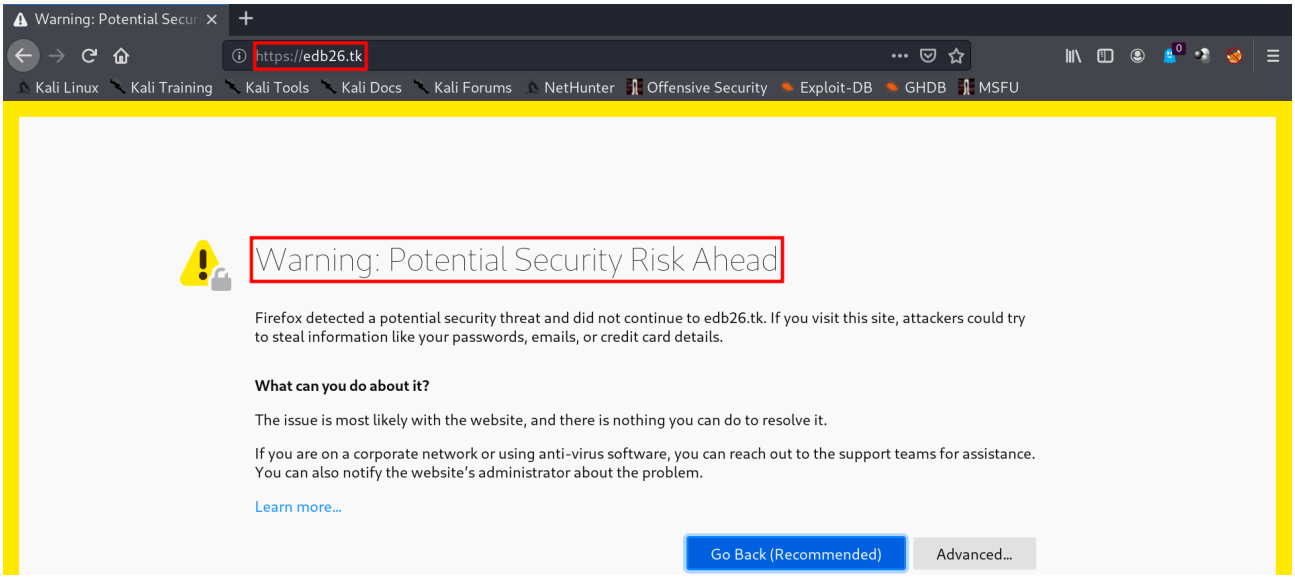
將原先 Certbot 新增的 ssl_certificate 與 ssl_certificate_key 改為剛剛建立出的 Certificate 與 Key。

```
vim /etc/nginx/sites-enabled/default
# ssl_certificate     /home/shoulderhu/edb26.crt;
# ssl_certificate_key /home/shoulderhu/edb26.key;
```

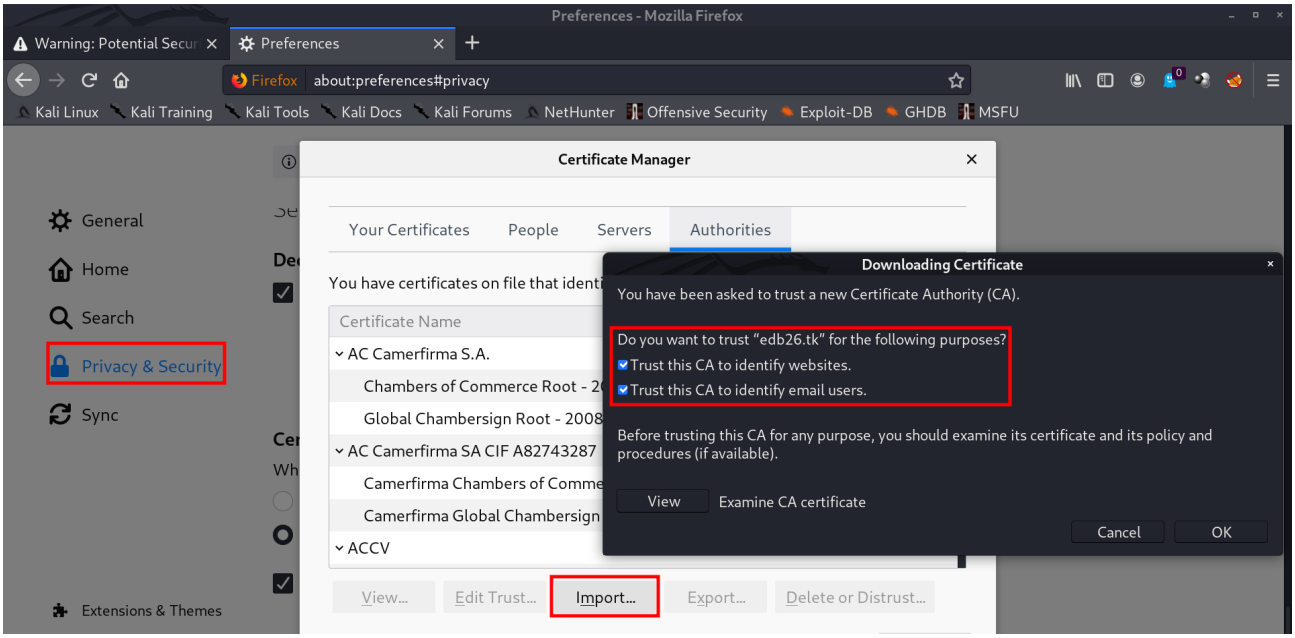
```
shoulderhu@ubuntu:~$ cat /etc/nginx/sites-enabled/default | tail -n 22
    listen [::]:443 ssl ipv6only=on; # managed by Certbot
    listen 443 ssl; # managed by Certbot
    #ssl_certificate /etc/letsencrypt/live/edb26.tk/fullchain.pem; # managed by Certbot
    ssl_certificate /home/shoulderhu/edb26.crt;
    #ssl_certificate_key /etc/letsencrypt/live/edb26.tk/privkey.pem; # managed by Certbot
    ssl_certificate_key /home/shoulderhu/edb26.key;
    #include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    #ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot
}
```

## 2.4 Add Root CA Certificate to Firefox

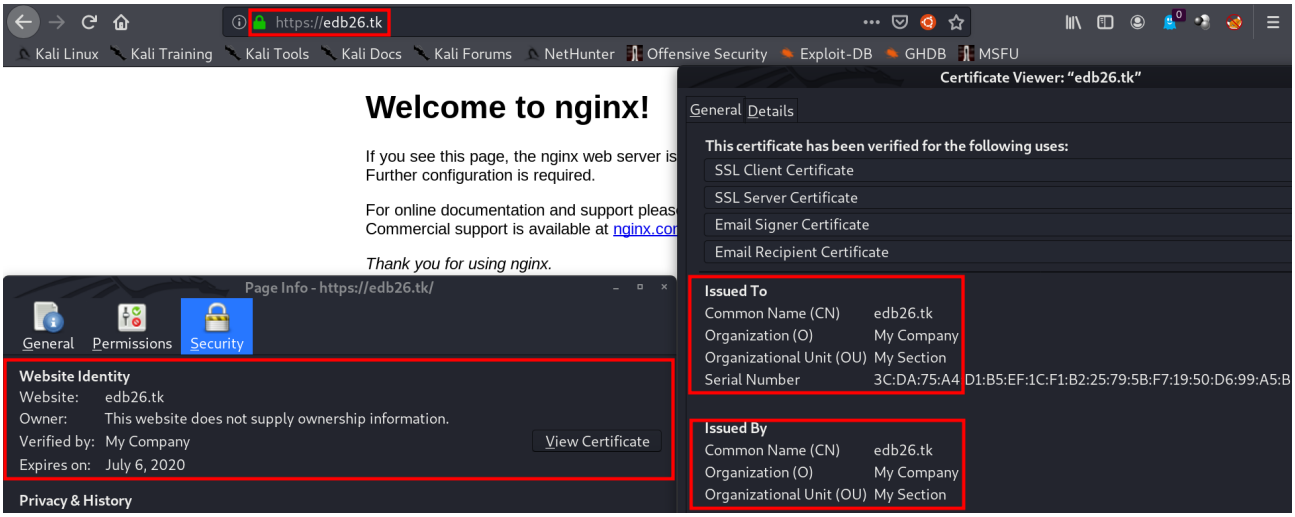
將 Root CA Certificate 加到瀏覽器所信任的 Root CA list 當中，這樣瀏覽網站時就不會出現警告。




Import Root CA 之前，瀏覽器會跳出警告




Import Root CA on Firefox




Import Root CA 後，就不會再有 Warning

---

# 3. Use man-in-the-middle to decrypt HTTPS encryption

## 3.1 Get mitmproxy

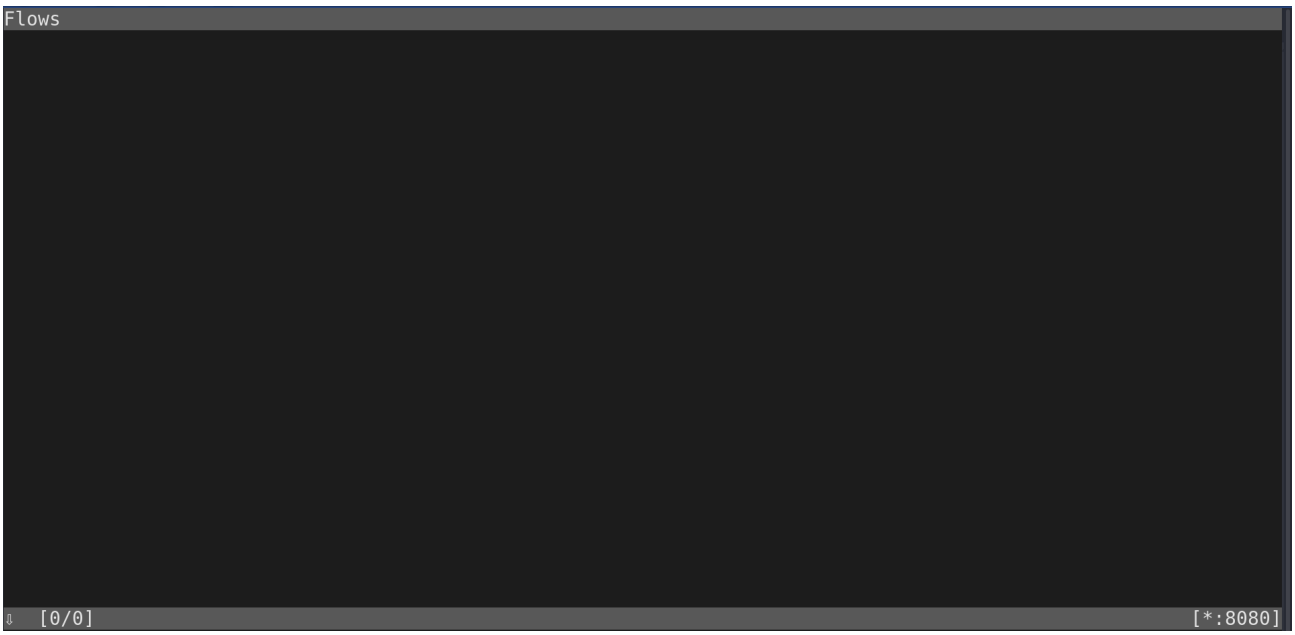
官方網站上直接提供能在 linux 上執行的 binary。

```
wget https://snapshots.mitmproxy.org/5.0.1/mitmproxy-5.0.1-linux.tar.gz
tar xzvf mitmproxy-5.0.1-linux.tar.gz
# mitmproxy
# mitmdump
# mitmweb
```

## 3.2 Setup mitmproxy

直接執行 mitmproxy 即可，預設端口為 8080，或是用 `-p` 指定其他的端口。

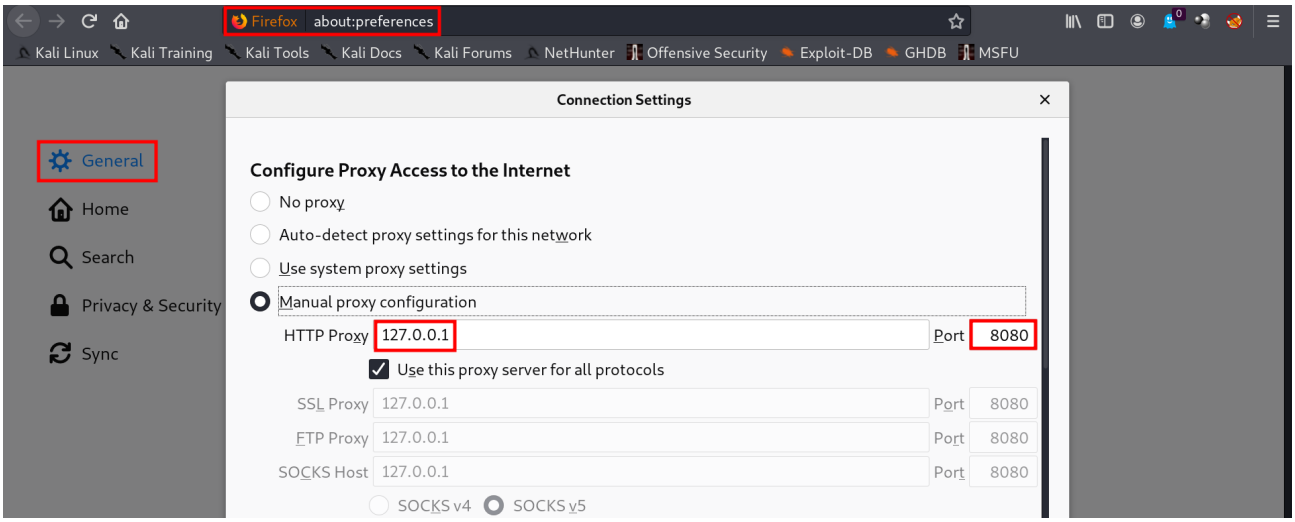
```
./mitmproxy
```




mitm proxy 啟動後的畫面

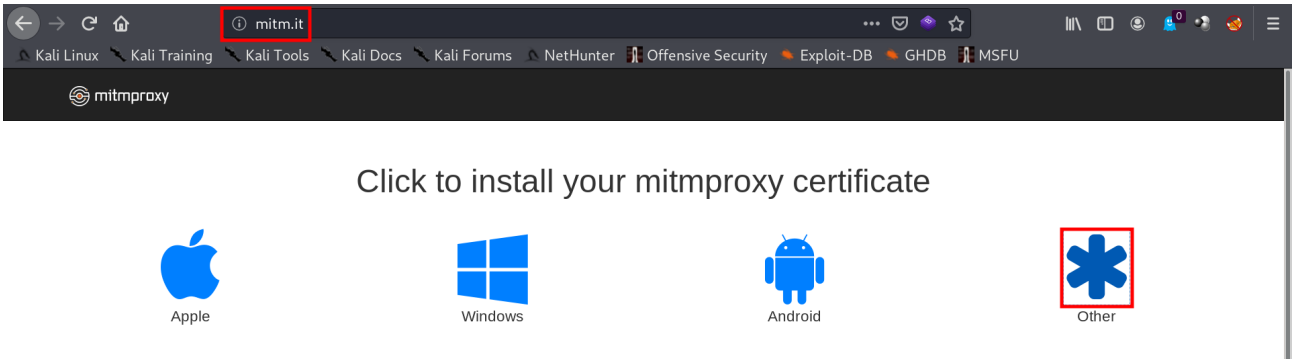
## 3.3 Setup Firefox & Install mitmproxy Root CA

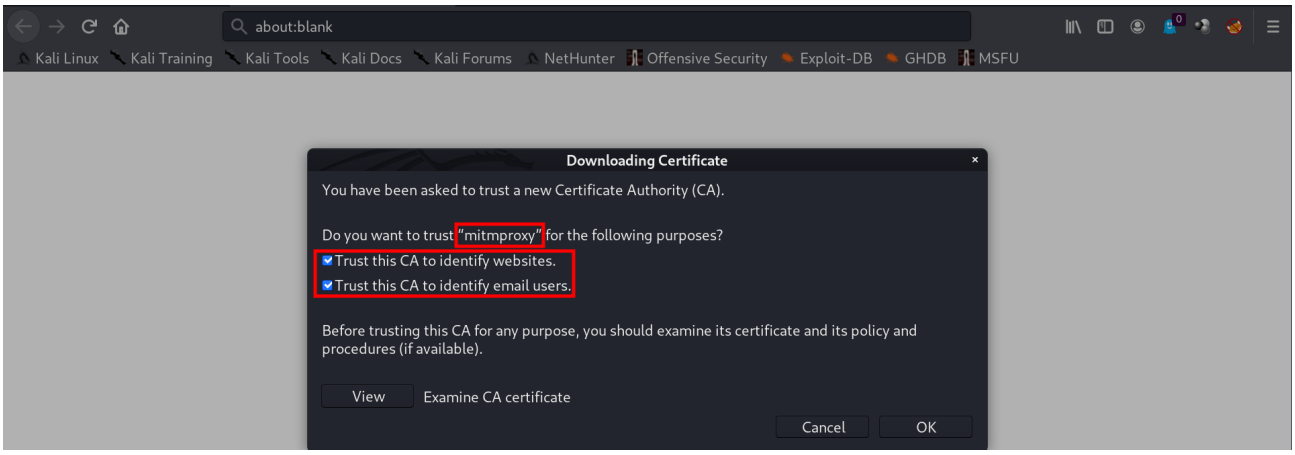
在 Firefox 的偏好設定當中設定 proxy，指向 mitmproxy。




Firefox 設定 proxy

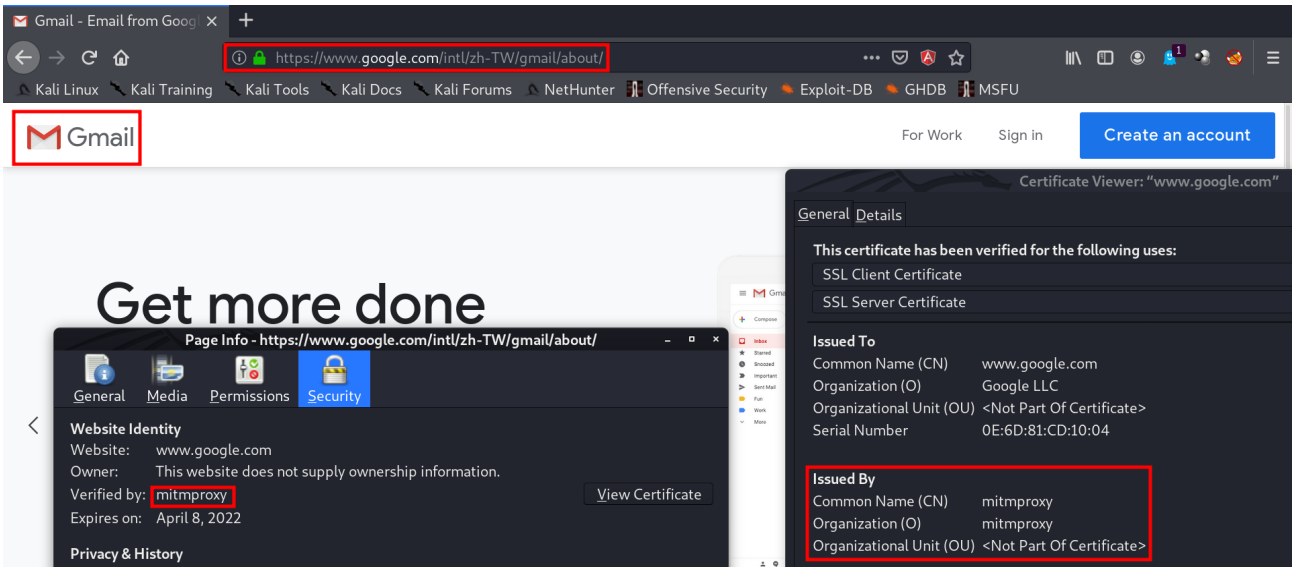
將 mitmproxy 的 certificate 加入到 Firefox 信任的 CA list 當中。



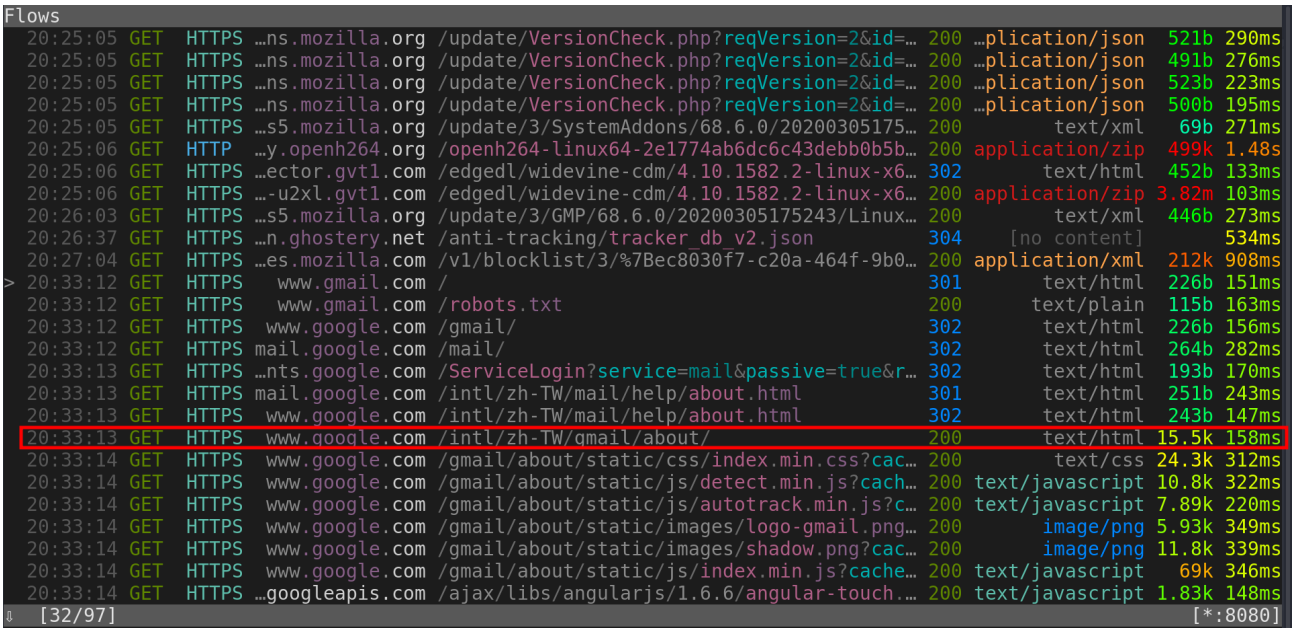




## 3.4 Intercept & Decrypt HTTPS connection

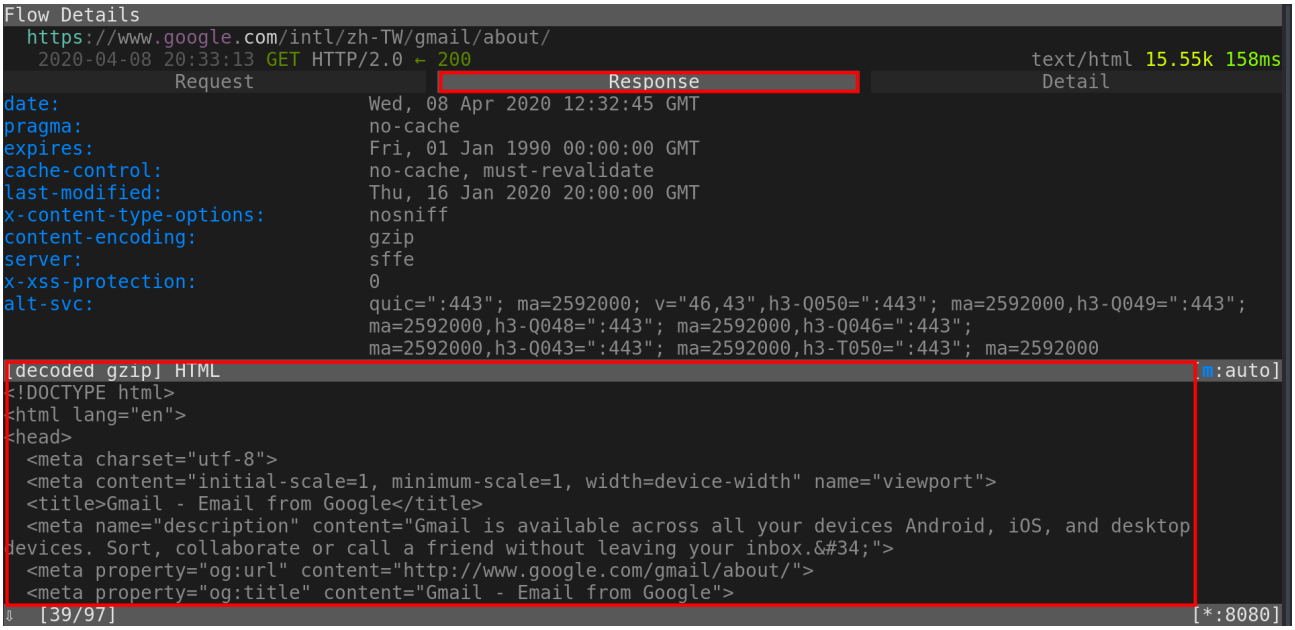
以瀏覽器連線到 gmail 為例，可以看到 mitmproxy Root CA 所簽署的 www.google.com 憑證。



接下來從 mitmproxy 的 console 當中找到相符的 HTTP request，按下 Enter 觀看封包的細節資訊。




在 Response Tab 當中，便可看見攔截下來的明文 html。




# Reference

How It Works https://letsencrypt.org/how-it-works/