Using Cert Bot to get a Valid SSL certificate

This guide will detail how to get a valid SSL cert using certbot on Linux operating systems

Certbot user guide

Commands for defining Key Type

https://www.onepagezen.com/letsencrypt-auto-renew-certbot-apache/

Configuration file

https://eff-certbot.readthedocs.io/en/stable/using.html#config-file

Follow this guide to force Cerbot to use Elliptical Curve Diffe-Hellman Curves for all certificates. Any existing certs will be updated upon the next reboot.

Certbot accepts a global configuration file that applies its options to all invocations of Certbot. Certificate specific configuration choices should be set in the `.conf` files that can be found in `/etc/letsencrypt/renewal`.
By default no cli.ini file is created (though it may exist already if you installed Certbot via a package manager, for instance). After creating one it is possible to specify the location of this configuration file with `certbot --config cli.ini` (or shorter `-c cli.ini`). An example configuration file is shown below:

```
# This is an example of the kind of things you can do in a configuration file.
# All flags used by the client can be configured here. Run Certbot with
# "--help" to learn more about the available options.
#
# Note that these options apply automatically to all use of Certbot for
# obtaining or renewing certificates, so options specific to a single
# certificate on a system with several certificates should not be placed
# here.

# Use ECC for the private key
key-type = ecdsa
elliptic-curve = secp384r1

# Use a 4096 bit RSA key instead of 2048
rsa-key-size = 4096

# Uncomment and update to register with the specified e-mail address
# email = foo@example.com

# Uncomment to use the standalone authenticator on port 443
# authenticator = standalone

# Uncomment to use the webroot authenticator. Replace webroot-path with the
# path to the public_html / webroot folder being served by your web server.
# authenticator = webroot
# webroot-path = /usr/share/nginx/html

# Uncomment to automatically agree to the terms of service of the ACME server
# agree-tos = true

# An example of using an alternate ACME server that uses EAB credentials
# server = https://acme.sectigo.com/v2/InCommonRSAOV
# eab-kid = somestringofstuffwithoutquotes
# eab-hmac-key = yaddayaddahexhexnotquoted
```

If on the internal network set the DNS on the machine to use CloudFlared or Google. DNS will not resolve properly if using the internal DNS servers

1. install the package certbot *sudo apt install certbot -y*
2. navigate to the /etc/letsencrypt directory
3. Two types on install Manual and Auto
    1. sudo certbot certonly --manual --preferred-challenges dns -d "*.coltscomputer.services"
        1. this will pull a SSL cert from Let's Encrypt
    2. Go to the DNS hosted domain on Route 53 and add the _acme-challenge text to the existing _acme-challenge TXT record
4. For the auto install
    1. This guide shows how to configure a virtual host
        1. cd /etc/apache2/sites-enabled

    2. edit the .conf files there for the necessary domain name
        1. sudo nano *.conf
        2. edit the YOUR-DOMAIN-NAME line
2. sudo certbot certonly --apache --key-type ecdsa --preferred-challenges dns -d "*.coltscomputer.services"
    1. This should auto renew
    2. use *sudo certbot renew --dry-run* to test if the renewal will work automatically

---

Revision #12
Created 25 December 2023 07:26:24 by ColtM
Updated 7 August 2024 23:24:39 by ColtM