Foundations to Computer Security

Sameer Gupta	Lakshay Chauhan	Chaitanya Arora	Ankit Kumar
2021093	2021060	2021033	2021015

Project Description

The Real Estate Aggregator System: The focus of this project is to create a portal that facilitates the secure exchange and verification of property-related documents and to enable secure transactions during property buying/selling/renting.

August Milestone

Tech Stack:

- Frontend: HTML, CSS, JS
- Backend: Django
- Database: SQLite
- Web Server: Nginx & Gunicorn

> Self signed SSL certificate generated using Certbot and Let's Encrypt.

Commands used to setup the entire server:

A total of 705 Commands were used to setup the entire server. The commands are listed in the file `commands_history.txt` File.

The sequence of commands used to setup the server is as follows:

1. Installing NGINX and configuring it

```
> sudo apt install nginx
> sudo systemctl start nginx
> sudo systemctl enable nginx
> sudo nano /etc/nginx sites-available/twentyfiveacres
```

then I edited the config file to redirect port 80 to 443 and also added the SSL certificate path. The certificate was generated using Certbot and Let's Encrypt.

```
> sudo apt install certbot python3-certbot-nginx
> sudo certbot --nginx -d 192.168.2.235
> sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout
/etc/ssl/private/nginx-selfsigned.key -out /etc/ssl/certs/nginx-selfsigned.crt
```

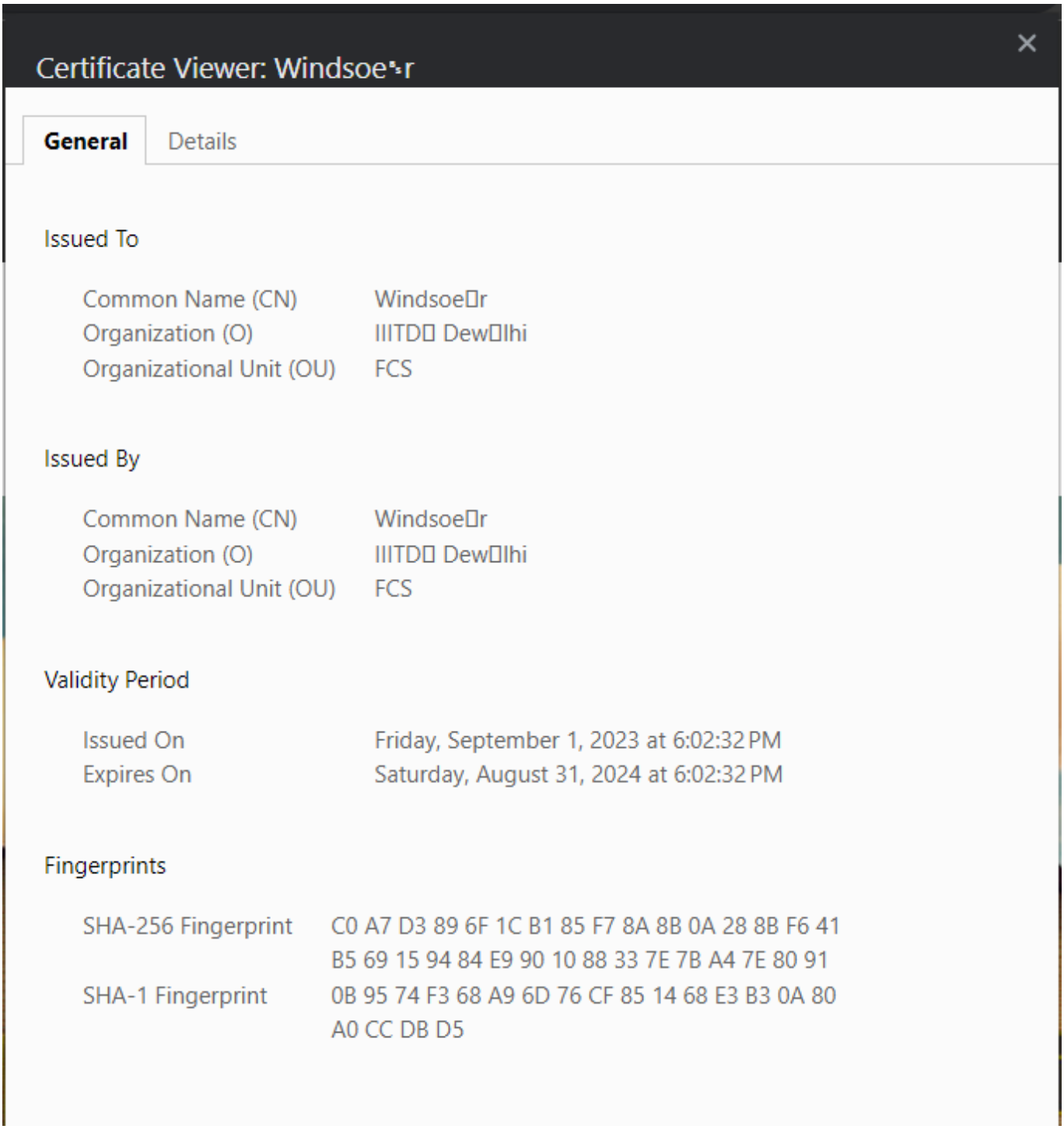
The certificates are located in the `/etc/ssl` directory.

> /etc/ssl/certs/nginx-selfsigned.crt
> /etc/ssl/private/nginx-selfsigned.key

The final nginx config file looks like this:

```
server {
        listen 80;
        server_name 192.168.2.235;

        location / {
                return 301 https://$host$request_uri;
        }
}

server {
        listen 443 ssl;
        server_name 192.168.2.235;

        ssl_certificate /etc/ssl/certs/nginx-selfsigned.crt;
        ssl_certificate_key /etc/ssl/private/nginx-selfsigned.key;

        location = /favicon.ico {
                alias /home/iiitd/windsor/twentyfiveacres/static_collected/videoplayer/logo.png;
 access_log off; log_not_found off; }
        location /static/ {
                alias /home/iiitd/windsor/twentyfiveacres/static_collected/;
        }

        location / {
                proxy_set_header Host $http_host;
                proxy_set_header X-Real-IP $remote_addr;
                proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
                proxy_set_header X-Forwarded-Proto $scheme;
                proxy_pass http://127.0.0.1:8000;
        }
}
```

and the SSL certificate looks like this:



2. Installing Django and Gunicorn

```
> pip3 install Django
> mkdir windsor
> cd windsor
> python3 -m venv venv
> source venv/bin/activate
> django-admin startproject > twentyfiveacres
> pip install gunicorn
> sudo nano /etc/systemd/system/> gunicorn.service
> sudo systemctl daemon-reload
> sudo systemctl start gunicorn
```

After this I edited the gunicorn.service file to add the path to the project and the wsgi file.

And finally, I created a html file in Django and ran the server using gunicorn.

You can find some of the debugging commands and file permission commands in the following code block.

```
> sudo nginx -t
> sudo systemctl reload nginx
> sudo journalctl -u gunicorn
> sudo chown www-data:www-data /home/iiitd/windsor/twentyfiveacres/gunicorn.sock
> sudo chmod +x /home/iiitd/windsor
> sudo systemctl restart gunicorn
> sudo systemctl restart nginx
> sudo cat /var/log/nginx/error.log
> sudo chown -R iiitd:www-data /home/iiitd/windsor/twentyfiveacres/
> sudo chmod 750 /home/iiitd/windsor/twentyfiveacres/
> sudo chown iiitd:www-data /home/iiitd/windsor/twentyfiveacres/gunicorn.sock
> sudo chmod 770 /home/iiitd/windsor/twentyfiveacres/gunicorn.sock
```

with this, the Final website looks like this: