Part 1: Identity-Based Access Using PKI

Step 1: Install Required Software

1. Update and Upgrade Ubuntu

```
sudo apt update && sudo apt upgrade -y
```

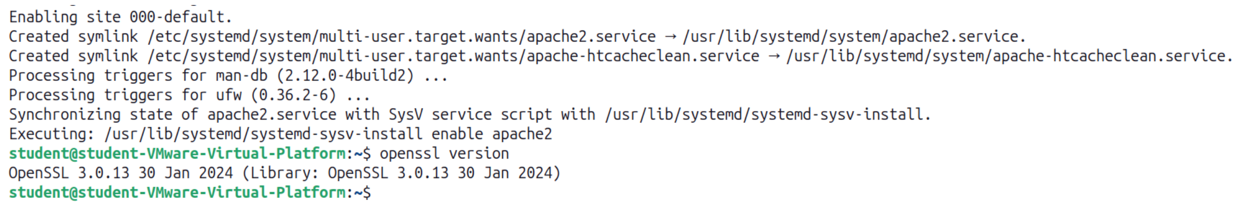
2. Install OpenSSL for Certificate Management

```
sudo apt install openssl -y
openssl version  # Verify installation
```

3. Install Apache Web Server

```
sudo apt install apache2 -y
sudo systemctl start apache2
sudo systemctl enable apache2
```

Test Apache: Open http://localhost in your browser. You should see the Apache default page.



```
Enabling site 000-default.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /usr/lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /usr/lib/systemd/system/apache-htcacheclean.service.
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for ufw (0.36.2-6) ...
Synchronizing state of apache2.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable apache2
student@student-VMware-Virtual-Platform:~$ openssl version
OpenSSL 3.0.13 30 Jan 2024 (Library: OpenSSL 3.0.13 30 Jan 2024)
student@student-VMware-Virtual-Platform:~$
```

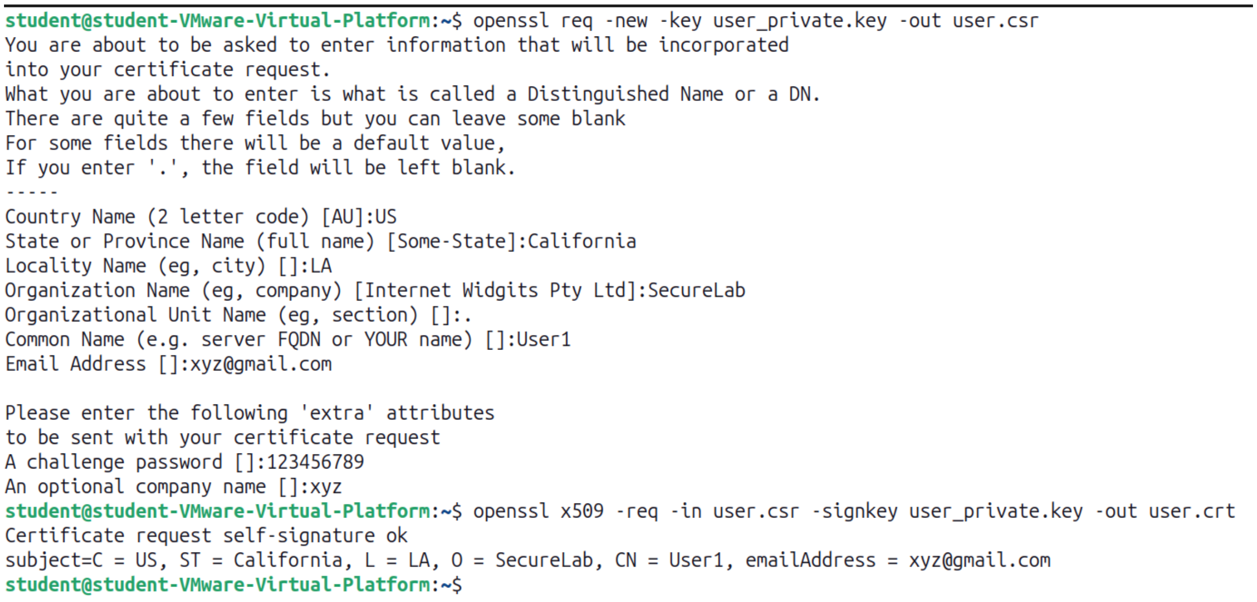
Step 2: Generate Public/Private Key Pair

1. Generate the User's Private Key

```
openssl genpkey -algorithm RSA -out user_private.key
```

2. Extract the Public Key

```
openssl rsa -pubout -in user_private.key -out user_public.key
```



```
student@student-VMware-Virtual-Platform:~$ openssl req -new -key user_private.key -out user.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:California
Locality Name (eg, city) []:LA
Organization Name (eg, company) [Internet Widgits Pty Ltd]:SecureLab
Organizational Unit Name (eg, section) []:.
Common Name (e.g. server FQDN or YOUR name) []:User1
Email Address []:xyz@gmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:123456789
An optional company name []:xyz
student@student-VMware-Virtual-Platform:~$ openssl x509 -req -in user.csr -signkey user_private.key -out user.crt
Certificate request self-signature ok
subject=C = US, ST = California, L = LA, O = SecureLab, CN = User1, emailAddress = xyz@gmail.com
student@student-VMware-Virtual-Platform:~$
```

Step 3: Create & Sign a User Certificate

1. Create a Certificate Signing Request (CSR)

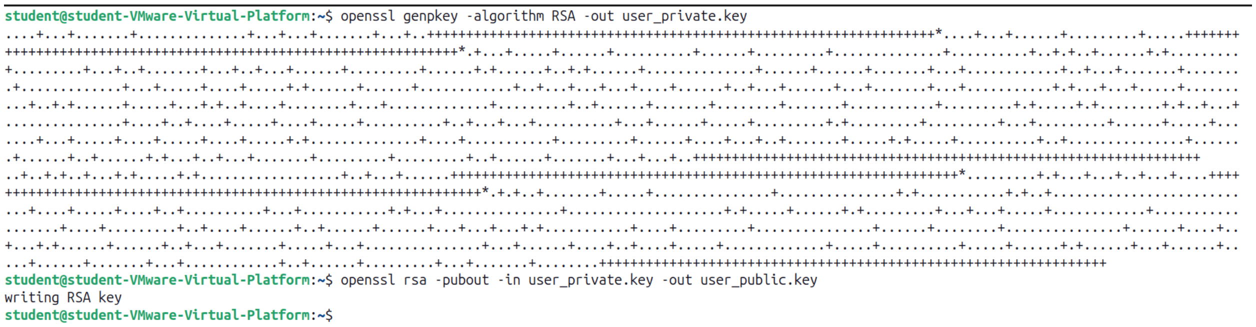
openssl req -new -key user_private.key -out user.csr

When prompted, enter details like:

Country Name: US
Organization Name: SecureLab
Common Name: User1

2. Create a Self-Signed Certificate

openssl x509 -req -in user.csr -signkey user_private.key -out user.crt



```
student@student-VMware-Virtual-Platform:~$ openssl genpkey -algorithm RSA -out user_private.key
....+...+.......+..............+...+...+.......+...+..+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*....+...+......+.........+.....+++++++
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*.+...+.....+......+.........+..........+......+.........+..............+..........+..+.+..+......+.+........
+.........+...+..+.......+...+..+...+......+.........+......+.+......+..+.+......+......+..............+......+......+.......+...+.............+..+...+.......+.......
.+............+...+.....+.....+.....+.+......+......+.............+..+...+...+...+...+...+.....+..+.......+...+.............+.+...+...+.....+.......
...+..+.+......+.....+...+.+..+....+........+...+.............+.........+..+......+.+......+.......+........+.......+...........+.........+.+.....+.+.........+.+..+...+
...............+....+..+....+.....+....+.....+..........+..+...+...+..........+...+...+.......+.....+.........+.+..........+.........+...+.........+......+.....+...
....+...+.....+....+.....+....+.....+.+...............+....+...........+.........+.......+......+....+...+..+.......+.....+.+.....+..........+..+...............+......
.+......+..+......+.+...+..+...+.......+.........+.........+..+......+.......+...+..+...+..+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
..+..+.+..+...+.+.....+.+.................+..+...+......+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*.........+.+...+...+..+...+....++++
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*.+.+..+.......+.....+.......+.....+...............+...............+.+...........+.+..+.........................
...+....+.....+....+..+..........+...+..........+.+...+..............+....................+.+....+......+.+.........+...+...+.....+............+...........
.......+....+.........+..+....+......+..+......+......+...+...+..+.+..........+....+.........+................+......+.........+...............+......+....+..
+...+.+......+......+..+...+.......+.....+...+.................+..+......+....+..+....+..+......+...+.....+............+.....+..........+.....+......+.+......+...+......+..
...+......+.......+...+............+..+......+.........+...+.......+........++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
student@student-VMware-Virtual-Platform:~$ openssl rsa -pubout -in user_private.key -out user_public.key
writing RSA key
student@student-VMware-Virtual-Platform:~$
```

Step 4: Set Up a Certificate Authority (CA)

1. Create the CA Private Key

openssl genpkey -algorithm RSA -out ca_private.key

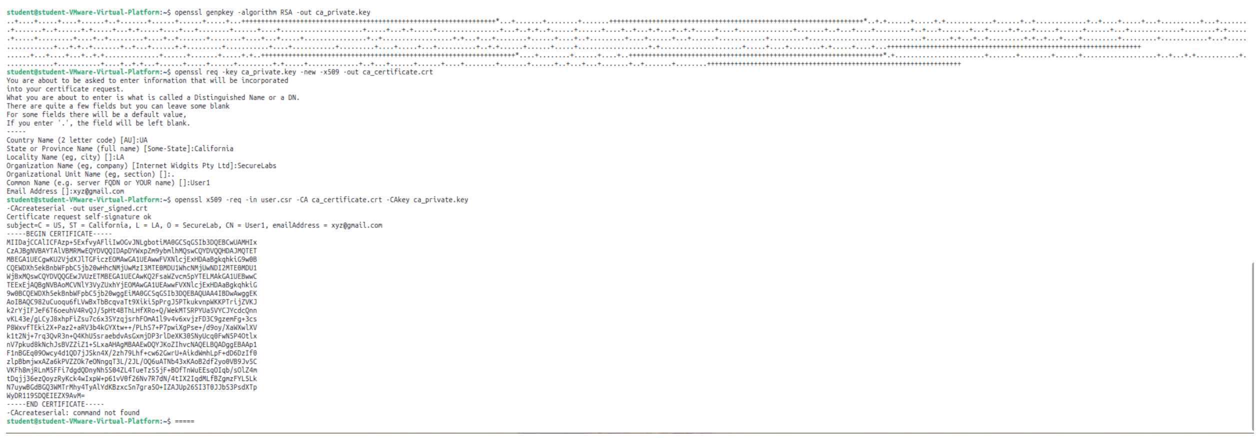
2. Create the CA Certificate

openssl req -key ca_private.key -new -x509 -out ca_certificate.crt

3. Sign the User's Certificate Using the CA

openssl x509 -req -in user.csr -CA ca_certificate.crt -CAkey ca_private.key -CAcreateserial -out user_signed.crt

Now, user_signed.crt is issued and signed by the CA.



```
student@student-VMware-Virtual-Platform:~$ openssl genpkey -algorithm RSA -out ca_private.key
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
student@student-VMware-Virtual-Platform:~$ openssl req -key ca_private.key -new -x509 -out ca_certificate.crt
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:UA
State or Province Name (full name) [Some-State]:California
Locality Name (eg, city) []:LA
Organization Name (eg, company) [Internet Widgits Pty Ltd]:SecureLabs
Organizational Unit Name (eg, section) []:.
Common Name (e.g. server FQDN or YOUR name) []:User1
Email Address []:xyz@gmail.com
student@student-VMware-Virtual-Platform:~$ openssl x509 -req -in user.csr -CA ca_certificate.crt -CAkey ca_private.key
-CAcreateserial -out user_signed.crt
Certificate request self-signature ok
subject=C = US, ST = California, L = LA, O = SecureLab, CN = User1, emailAddress = xyz@gmail.com
-----BEGIN CERTIFICATE-----
MIIDajCCAlICFAzp+5ExfvyAFliIwOGvJNLgbotiMA0GCSqGSIb3DQEBCwUAMHIx
CzAJBgNVBAYTAlVBMRMwEQYDVQQIDApDYWxpZm9ybmlhMQswCQYDVQQHDAJMQTET
MBEGA1UECgwKU2VjdXJlTGFiczEOMAwGA1UEAwwFVXNlcjExHDAaBgkqhkiG9w0B
CQEWDXh5ekBnbWFpbC5jb20wHhcNMjUwMzI3MTE0MDU1WhcNMjUwNDI2MTE0MDU1
WjBxMQswCQYDVQQGEwJVUzETMBEGA1UECAwKQ2FsaWZvcm5pYTELMAkGA1UEBwwC
TEExEjAQBgNVBAoMCVNlY3VyZUxhYjEOMAwGA1UEAwwFVXNlcjExHDAaBgkqhkiG
9w0BCQEWDXh5ekBnbWFpbC5jb20wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEK
AoIBAQC982uCuoqu6fLVwBxTbBcqvaTt9Xiki5pPrgJ5PTkukvnpWKKPTrijZVKJ
k2rYjIFJeF6T6oeuhV4RvQJ/5pHt4BThLHfXRo+Q/WekMTSRPYUa5VYCJYcdcQnn
vKL43e/gLCyJ8xhpFiZsu7c6x3SYzqjsrhFOmA1l9v4v6xvjzFD3C9gzemFg+3cs
P8WxvfTEki2X+Paz2+aRV3b4kGYXtw++/PLhS7+P7pwiXgPse+/d9oy/XaWXwlXV
k1t2Nj+7rq3QvR3n+Q4KhU5sraebdvAsGxmjDP3rlDeXK30SNyUcq0FwN5P4Otlx
nV7pkud8kNchJsBVZZiZ1+SLxaAHAgMBAAEwDQYJKoZIhvcNAQELBQADggEBAAp1
F1nBGEq09Owcy4d1QD7jJSkn4X/2zh79Lhf+cw62GwrU+AikdWmhLpF+dD6DzIf0
zlpBbmjwxAZa6kPVZZOk7eONngqT3L/2JL/OQ6uATNb43xKAoB2df2yo0VB9Jv5C
VKFh8mjRLnM5FFi7dgdQDnyNh5S04ZL4TueTzS5jF+BOfTnWuEEsqOIqb/sOlZ4m
tDqjj36ezQoyzRyKck4wIxpW+p61vV0f26Nv7R7dN/4tIX2IqdMLfBZgmzFYL5Lk
N7uywBGdBGQ3WMTrMhy4TyAlYdKBzxcSn7gra5O+IZAJUp26SI3T0JJb53PsdXTp
WyDR119SDQEIEZX9AvM=
-----END CERTIFICATE-----
-CAcreateserial: command not found
student@student-VMware-Virtual-Platform:~$ =====
```

Step 5: Configure Apache for Identity-Based Access

1. Enable SSL Module

sudo a2enmod ssl

2. Configure SSL Certificate in Apache

Edit the Apache SSL configuration:

sudo nano /etc/apache2/sites-available/default-ssl.conf

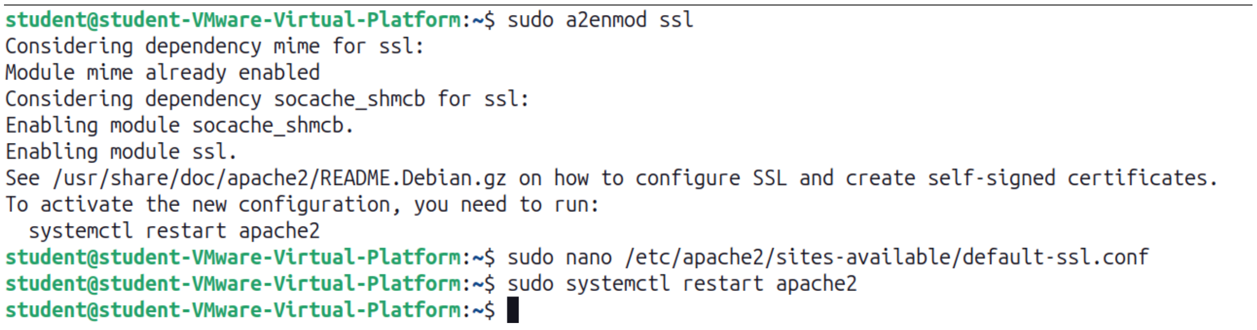
Replace <VirtualHost *:443> section with:

```
<VirtualHost *:443>
DocumentRoot /var/www/html
SSLEngine on
SSLCertificateFile /etc/ssl/certs/user_signed.crt
SSLCertificateKeyFile /etc/ssl/private/user_private.key
```

SSLCACertificateFile /etc/ssl/certs/ca_certificate.crt
SSLVerifyClient require
SSLVerifyDepth 1
</VirtualHost>

3. Restart Apache

sudo systemctl restart apache2



```
student@student-VMware-Virtual-Platform:~$ sudo a2enmod ssl
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
  systemctl restart apache2
student@student-VMware-Virtual-Platform:~$ sudo nano /etc/apache2/sites-available/default-ssl.conf
student@student-VMware-Virtual-Platform:~$ sudo systemctl restart apache2
student@student-VMware-Virtual-Platform:~$ 
```

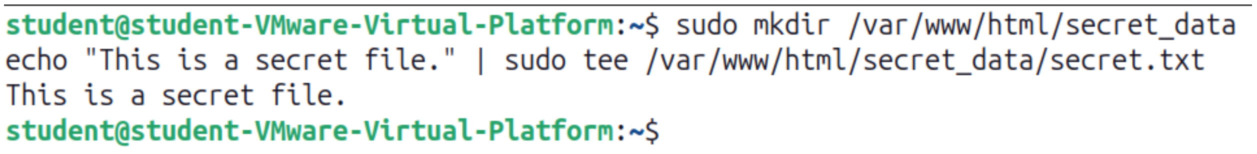
Step 6: Create a Protected Directory

1. Create a Secure Folder

sudo mkdir /var/www/html/secret_data
echo "This is a secret file." | sudo tee /var/www/html/secret_data/secret.txt

2. Restrict Access to the Directory

Apache will require valid certificates to access /secret_data/.
Apache is now configured for IBAC!



```
student@student-VMware-Virtual-Platform:~$ sudo mkdir /var/www/html/secret_data
echo "This is a secret file." | sudo tee /var/www/html/secret_data/secret.txt
This is a secret file.
student@student-VMware-Virtual-Platform:~$
```

Step 7: Simulate the Confinement Problem

1. Create Python Script

nano access_control.py

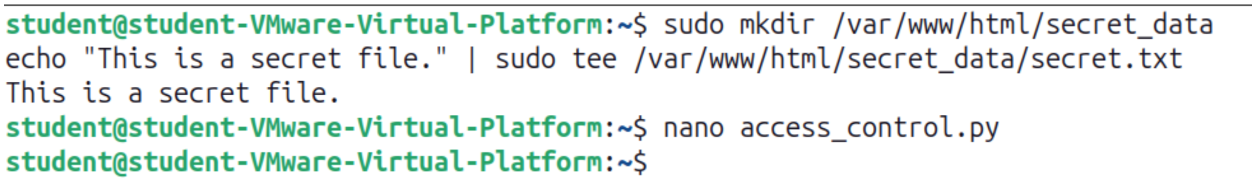
2. Add the Following Code

# Function to check access based on clearance levels
def check_access(user_clearance, resource_clearance):

```
if user_clearance >= resource_clearance:
        return "Access Granted"
else:
        return "Access Denied"

# Clearance Levels
clearance_levels = {
1: "Top Secret",
2: "Secret",
3: "Confidential",
4: "Unclassified"
}

# Get user input
print("Available Clearance Levels:")
for level, name in clearance_levels.items():
   print(f"{level}: {name}")

user_clearance = int(input("Enter your clearance level (1-4): "))
resource_clearance = int(input("Enter the clearance level of the resource (1-4): "))

# Check Access
access_result = check_access(user_clearance, resource_clearance)

# Display Result
print(f"Your Clearance Level: {clearance_levels[user_clearance]}")
print(f"Resource Clearance Level: {clearance_levels[resource_clearance]}")

print(access_result)
```

3. Code Run the Script

```
python3 access_control.py
```



```
student@student-VMware-Virtual-Platform:~$ sudo mkdir /var/www/html/secret_data
echo "This is a secret file." | sudo tee /var/www/html/secret_data/secret.txt
This is a secret file.
student@student-VMware-Virtual-Platform:~$ nano access_control.py
student@student-VMware-Virtual-Platform:~$
```