TP SSR Report

PKI B Linux

Directed by : Messar Cylia and Tagzirt Elissa
Group: SIQ1
2023-2024

Introduction:

This report documents the implementation of a security infrastructure based on SSL/TLS digital certificates using a Certification Authority (CA) on a local network. The main objective of this practical work is to establish a CA and manage SSL digital certificates to secure communications on an Apache web server running under Linux. To achieve this, we are using three separate machines:

Machine	Distribution	Adresse
Apache Web Server	Ubuntu 20	10.10.0.2
CA	Ubuntu 20	10.10.0.3
Client	Ubuntu 20	10.10.1.2

The Apache web server hosts HTTPS-secured sites, while the CA is responsible for managing and issuing the SSL certificates required for this security.

Procedure:

1) CA machine configuration:
   - Set up directory and file infrastructure for OpenSSL.
   - Create a self-signed CA certificate.
   - Modify the OpenSSL configuration file to adjust specific parameters.
   - Sign certificate requests issued by the Apache web server.
2) Configure the Apache web server:
   - Install and configure Apache to support SSL/TLS.
   - Generate a certification request for the server certificate.
   - Transfer and validate the certification request with the CA.
   - Configure SSL certificates and private keys for Apache.
   - Enable SSL support in Apache and restart service to apply configurations.
3) Client configuration:
   - Import and configure certificate authority in client browser.
   - Validate secure access (HTTPS) to the site hosted on the Apache web server.

In the following, we detail each step taken to establish a secure SSL/TLS certificate infrastructure.

Note:

The client and server are on two different networks. Routing is handled by a different machine because we had continued to run the SSLsniff test on the same machines used for that test. Under normal circumstances, different networks would not have been required for this TP, which means we could simply have used NAT and had automatic addresses. This lab was done a long time ago, so there are only screenshots of the results (files, created directory, and SSL certificate result) but no screenshots of the steps as we did not keep a record.

The steps of realization:

1) CA machine:

a) Switch to root user:
```
# sudo su
```
b) Change to the /root directory:
```
# cd /root
```
c) Create a directory named tp-openssl and its subdirectories:
```
# mkdir -p /root/tp-openssl/certs /root/tp-openssl/crl
/root/tp-openssl/private
```
d) Copy the openssl configuration file to the tp-openssl directory:
```
cp /etc/ssl/openssl.cnf /root/tp-openssl/
```
e) These files are created:
```
# touch /root/tp-openssl/index.txt
# echo 01 | sudo tee /root/tp-openssl/serial > /dev/null
```
f) Create a self-signed certification authority (CA) certificate:
```
openssl req -new -x509 -extensions v3_ca -keyout
/root/tp-openssl/private/cakey.pem -out
/root/tp-openssl/cacert.pem -days 3650 -config
/root/tp-openssl/openssl.cnf
```
g) Display certificate content:
```
openssl x509 -text -in /root/tp-openssl/cacert.pem
```
h) Archive files:
```
tar -czf rootca.tar.gz -C /root/tp-openssl private/cakey.pem
cacert.pem
```

2) Apache Server Machine :

a) After installing apache server.

b) Change to the root user:
```
# sudo su
```
c) Change to the /root directory:
```
# cd /root
```
d) Create a directory named tp-openssl and its subdirectories:
```
# mkdir -p /root/tp-openssl/certs /root/tp-openssl/crl
/root/tp-openssl/private
```
e) Copy the openssl configuration file to the tp-openssl directory:
```
cp /etc/ssl/openssl.cnf /root/tp-openssl/
```
f) These files are created:
```
# touch /root/tp-openssl/index.txt
# echo 01 | sudo tee /root/tp-openssl/serial > /dev/null
```
g) Creating a certification request :
```
openssl req -config ./openssl.cnf -new -keyout
private/webkey.pem -out certs/newreq.pem
```
h) Copy newreq from VM server to VM CA (put it in certs) (using SSH protocol).

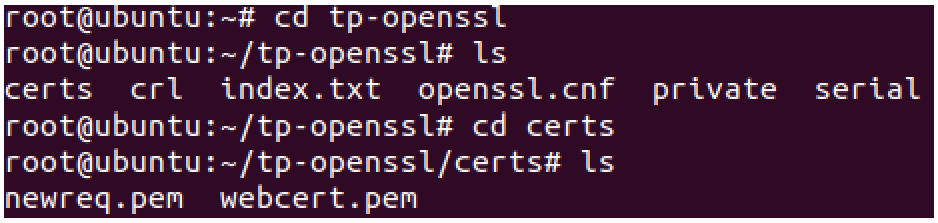
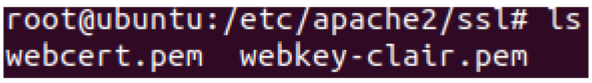
3) Return to the CA machine:

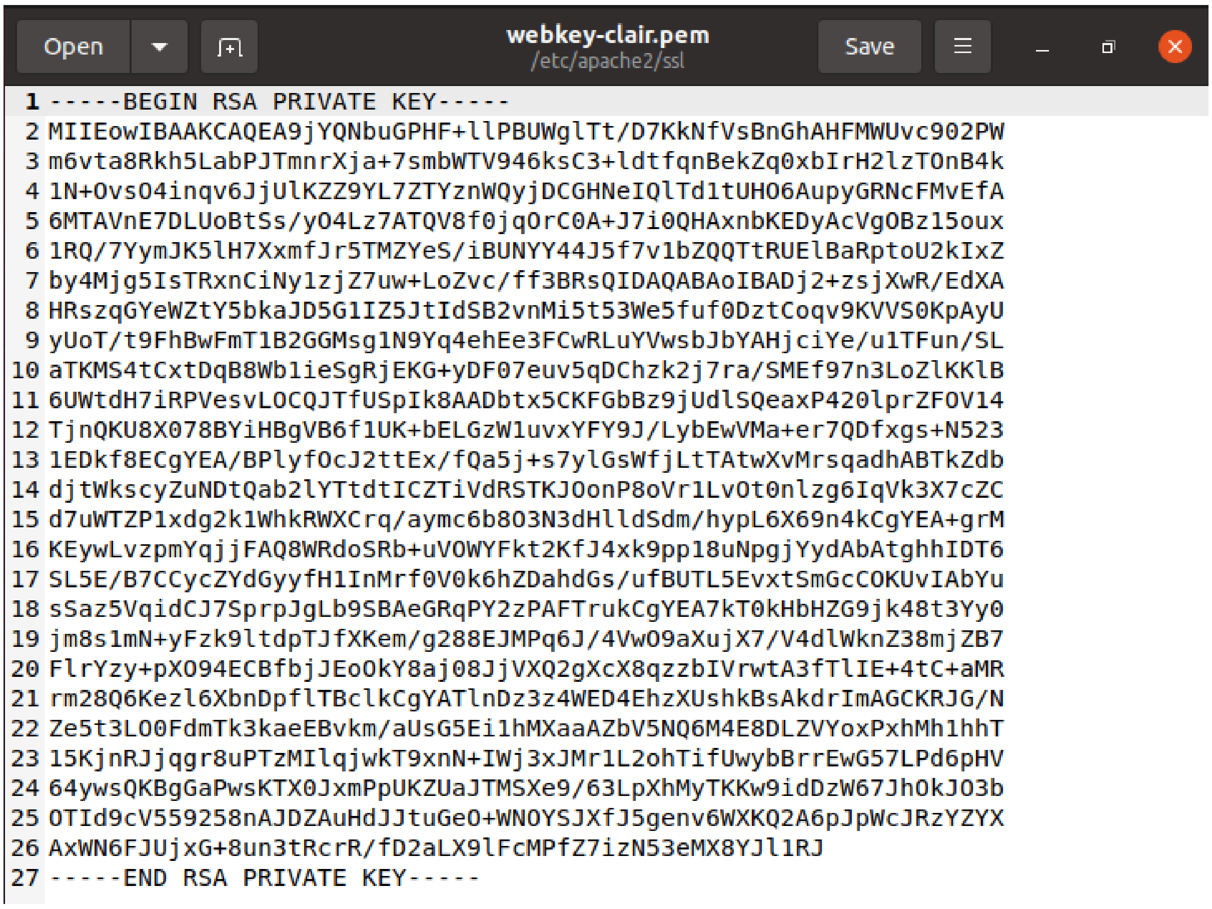
a) Open the openssl configuration file:
```
# nano /root/tp-openssl/openssl.cnf
```
b) Change dir = ./demoCA to dir = /root/tp-openssl. Close the file and save it.

c) Use OpenSSL to sign a Certificate Signing Request (CSR) and issue a certificate using an existing Certification Authority (CA):
```
openssl ca -config ./openssl.cnf -policy policy_anything -out
certs/webcert.pem -infiles certs/newreq.pem
```

d) Verification (OK):
```
openssl verify -CAfile cacert.pem certs/webcert.pem
```
e) Copy webcert from CA machine to Server machine:
```
openssl verify -CAfile cacert.pem certs/webcert.pem
```

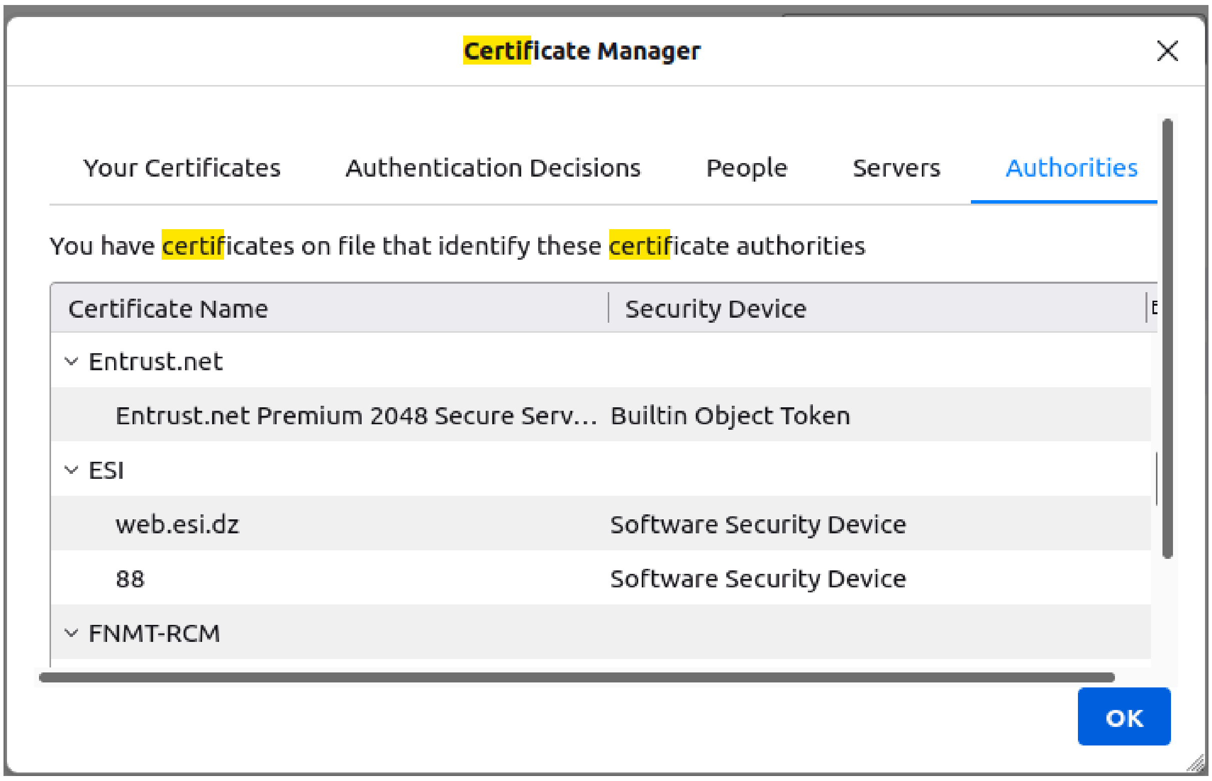
4) Return to Apache Server machine:

a) Private key conversion:
```
# openssl rsa -in private/webkey.pem -out
private/webkey-clair.pem
```
b) Create a directory for Apache SSL certificates:
```
# mkdir /etc/apache2/ssl
```
c) Copy certificate to Apache SSL directory:
```
# cp certs/webcert.pem /etc/apache2/ssl
```
d) Move private key to Apache SSL directory:
```
# mv private/webkey-clair.pem /etc/apache2/ssl
```
e) Change private key permissions:
```
# chmod 400 /etc/apache2/ssl/webkey-clair.pem
```
f) Enable Apache SSL module:
```
# a2enmod ssl
```
g) Restart Apache service:
```
# systemctl restart apache2
```
h) Create configuration file for default HTTPS site:
```
# touch /etc/apache2/sites-available/default-ssl
```
i) Activate default HTTPS site:
```
# a2ensite default-ssl
```
j) Reload Apache configuration:
```
# /etc/init.d/apache2 reload
```
k) Modify default HTTPS site configuration file:
```
# nano /etc/apache2/sites-available/default-ssl.conf
```
l) Check Apache configuration:
```
# apache2ctl configtest
```

```
root@ubuntu:~# cd tp-openssl
root@ubuntu:~/tp-openssl# ls
certs  crl  index.txt  openssl.cnf  private  serial
root@ubuntu:~/tp-openssl# cd certs
root@ubuntu:~/tp-openssl/certs# ls
newreq.pem  webcert.pem
```

```
root@ubuntu:/etc/apache2/ssl# ls
webcert.pem  webkey-clair.pem
```



```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEA9jYQNbuGPHF+llPBUWglTt/D7KkNfVsBnGhAHFMWUvc902PW
m6vta8Rkh5LabPJTmnrXja+7smbWTV946ksC3+ldtfqnBekZq0xbIrH2lzTOnB4k
1N+Ovs04inqv6JjUlKZZ9YL7ZTYznWQyjDCGHNeIQlTd1tUHO6AupyGRNcFMvEfA
6MTAVnE7DLUoBtSs/yO4Lz7ATQV8f0jqOrC0A+J7i0QHAxnbKEDyAcVgOBz15oux
1RQ/7YymJK5lH7XxmfJr5TMZYeS/iBUNYY44J5f7v1bZQQTtRUElBaRptoU2kIxZ
by4Mjg5IsTRxnCiNy1zjZ7uw+LoZvc/ff3BRsQIDAQABAoIBADj2+zsjXwR/EdXA
HRszqGYeWZtY5bkaJD5G1IZ5JtIdSB2vnMi5t53We5fuf0DztCoqv9KVVS0KpAyU
yUoT/t9FhBwFmT1B2GGMsg1N9Yq4ehEe3FCwRLuYVwsbJbYAHjciYe/u1TFun/SL
aTKMS4tCxtDqB8Wb1ieSgRjEKG+yDF07euv5qDChzk2j7ra/SMEf97n3LoZlKKlB
6UWtdH7iRPVesvLOCQJTfUSpIk8AADbtx5CKFGbBz9jUdlSQeaxP420lprZFOV14
TjnQKU8X078BYiHBgVB6f1UK+bELGzW1uvxYFY9J/LybEwVMa+er7QDfxgs+N523
1EDkf8ECgYEA/BPlyfOcJ2ttEx/fQa5j+s7ylGsWfjLtTAtwXvMrsqadhABTkZdb
djtWkscyZuNDtQab2lYTtdtICZTiVdRSTKJOonP8oVr1LvOt0nlzg6IqVk3X7cZC
d7uWTZP1xdg2k1WhkRWXCrq/aymc6b8O3N3dHlldSdm/hypL6X69n4kCgYEA+grM
KEywLvzpmYqjjFAQ8WRdoSRb+uVOWYFkt2KfJ4xk9pp18uNpgjYydAbAtghhIDT6
SL5E/B7CCycZYdGyyfH1InMrf0V0k6hZDahdGs/ufBUTL5EvxtSmGcCOKUvIAbYu
sSaz5VqidCJ7SprpJgLb9SBAeGRqPY2zPAFTrukCgYEA7kT0kHbHZG9jk48t3Yy0
jm8s1mN+yFzk9ltdpTJfXKem/g288EJMPq6J/4VwO9aXujX7/V4dlWknZ38mjZB7
FlrYzy+pXO94ECBfbjJEoOkY8aj08JjVXQ2gXcX8qzzbIVrwtA3fTlIE+4tC+aMR
rm28Q6Kezl6XbnDpflTBclkCgYATlnDz3z4WED4EhzXUshkBsAkdrImAGCKRJG/N
Ze5t3LO0FdmTk3kaeEBvkm/aUsG5Ei1hMXaaAZbV5NQ6M4E8DLZVYoxPxhMh1hhT
15KjnRJjqgr8uPTzMIlqjwkT9xnN+IWj3xJMr1L2ohTifUwybBrrEwG57LPd6pHV
64ywsQKBgGaPwsKTX0JxmPpUKZUaJTMSXe9/63LpXhMyTKKw9idDzW67JhOkJO3b
OTId9cV559258nAJDZAuHdJJtuGeO+WNOYSJXfJ5genv6WXKQ2A6pJpWcJRzYZYX
AxWN6FJUjxG+8un3tRcrR/fD2aLX9lFcMPfZ7izN53eMX8YJl1RJ
-----END RSA PRIVATE KEY-----
```

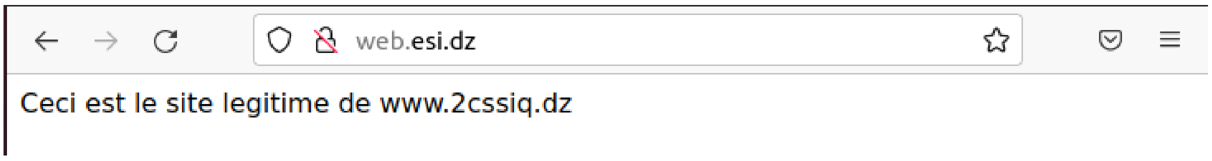
5) **Client machine:** Add the authority certificate to the client machine browser.



You have certificates on file that identify these certificate authorities

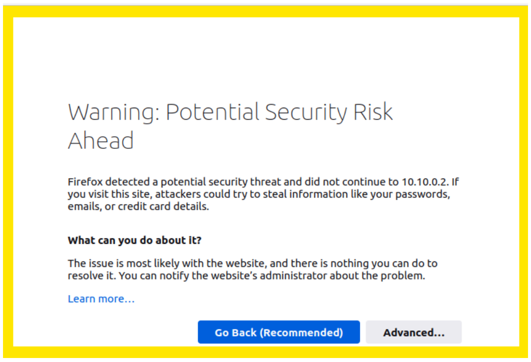
Certificate Name	Security Device
Entrust.net	
Entrust.net Premium 2048 Secure Serv…	Builtin Object Token
ESI	
web.esi.dz	Software Security Device
88	Software Security Device
FNMT-RCM	

Access http://web.esi.dz via the client machine:



Ceci est le site legitime de www.2cssiq.dz

Access https://web.esi.dz via the client machine:

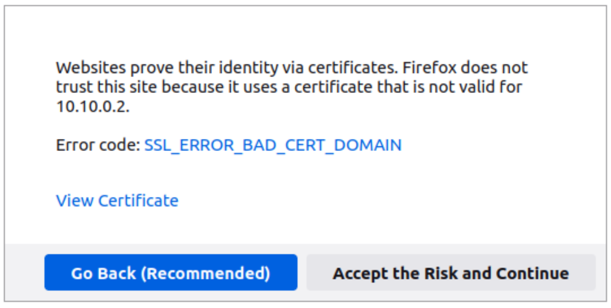


Warning: Potential Security Risk Ahead

Firefox detected a potential security threat and did not continue to 10.10.0.2. If you visit this site, attackers could try to steal information like your passwords, emails, or credit card details.

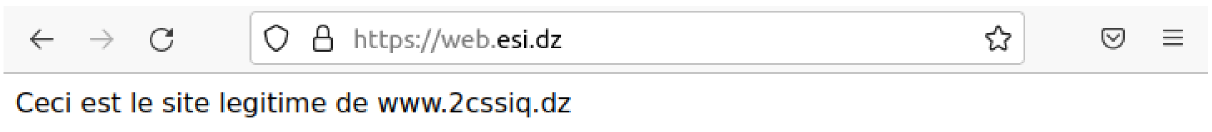
What can you do about it?

The issue is most likely with the website, and there is nothing you can do to resolve it. You can notify the website's administrator about the problem.

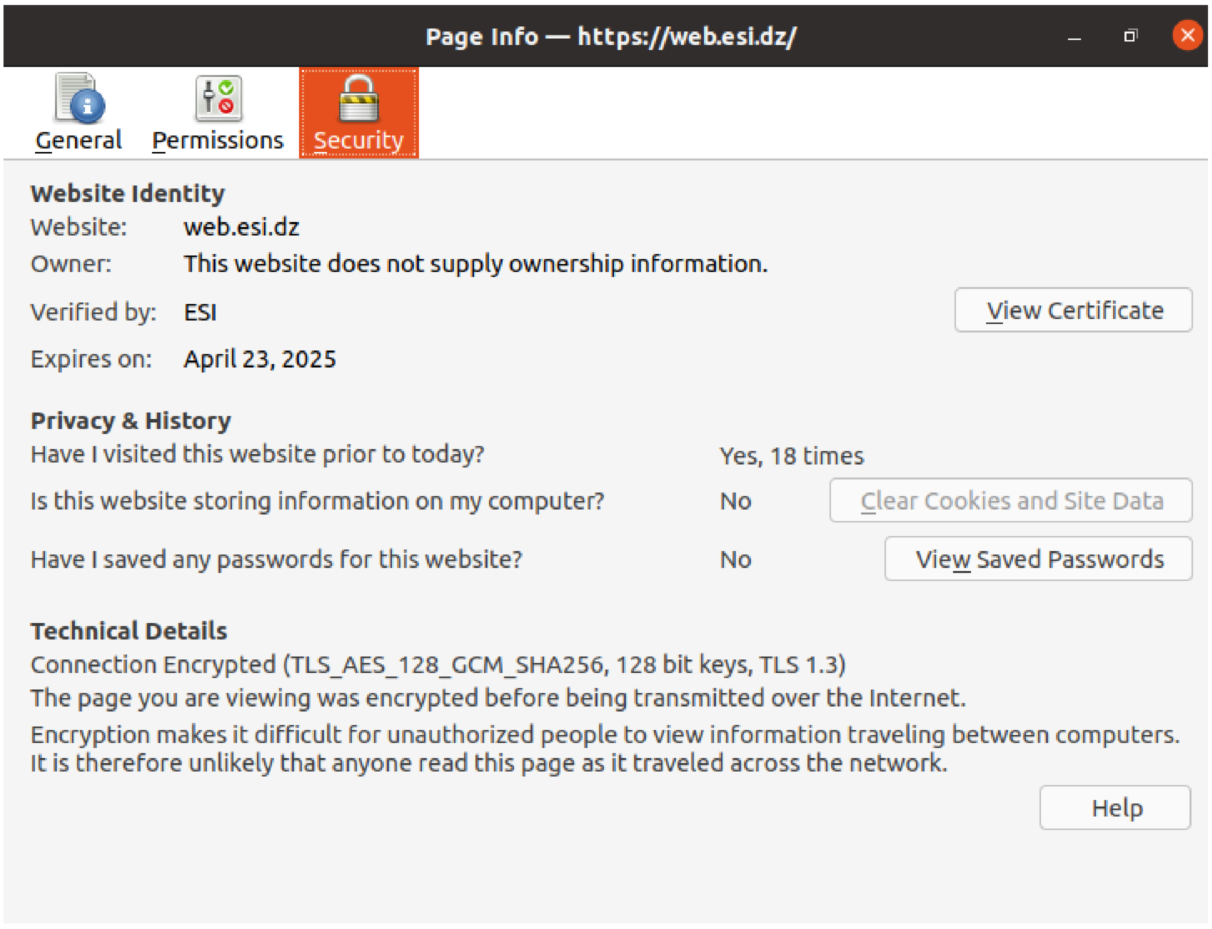


Websites prove their identity via certificates. Firefox does not trust this site because it uses a certificate that is not valid for 10.10.0.2.

Error code: SSL_ERROR_BAD_CERT_DOMAIN



Ceci est le site legitime de www.2cssiq.dz



Website Identity

Website:	web.esi.dz
Owner:	This website does not supply ownership information.
Verified by:	ESI
Expires on:	April 23, 2025

Privacy & History

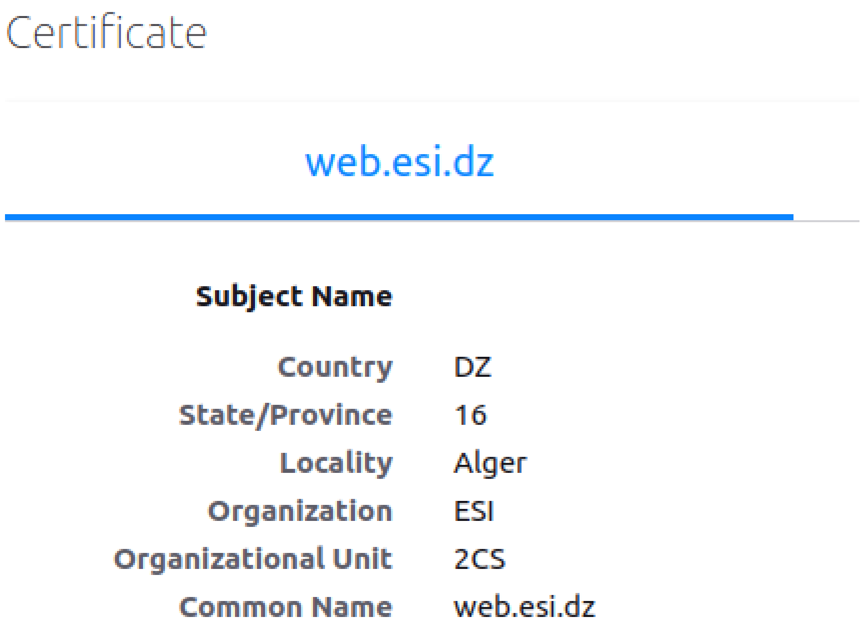
Have I visited this website prior to today?	Yes, 18 times
Is this website storing information on my computer?	No
Have I saved any passwords for this website?	No

Technical Details

Connection Encrypted (TLS_AES_128_GCM_SHA256, 128 bit keys, TLS 1.3)

The page you are viewing was encrypted before being transmitted over the Internet.

Encryption makes it difficult for unauthorized people to view information traveling between computers. It is therefore unlikely that anyone read this page as it traveled across the network.



Certificate

web.esi.dz

Subject Name

Country	DZ
State/Province	16
Locality	Alger
Organization	ESI
Organizational Unit	2CS
Common Name	web.esi.dz