



8 ΔΕΚΕΜΒΡΙΟΥ 2022

# ΑΣΦΑΛΕΙΑ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΕΡΓΑΣΙΑ 3

**Συντελεστές εργασίας**

Χριστοφορίδης Χαράλαμπος – Π19188

Γεωργιάδης Νικόλαος – Π19032

Καρκάνης Ευστράτιος – Π19064



# Περιεχόμενα

1. Δημιουργία αρχής πιστοποίησης .....	2
2. Δημιουργία και πιστοποίηση κλειδιών για server .....	4
3. Δημιουργία, πιστοποίηση και ανάκληση κλειδιών .....	5
4. Εισαγωγή πιστοποιητικού στον server .....	9
5. Διαμόρφωση του server για διπλή αυθεντικοποίηση.....	13

# 1. Δημιουργία αρχής πιστοποίησης

Σε αυτό το βήμα, θα δημιουργήσουμε μία αρχή πιστοποίησης, η οποία θα διαθέτει το δικό της αυτουπογεγραμμένο πιστοποιητικό. Για το σκοπό αυτό, χρησιμοποιήσαμε τις οδηγίες που δίνονται στο επόμενο link: <https://linuxconfig.org/apache-web-server-ssl-authentication>. Συγκεκριμένα, ολοκληρώσαμε τα βήματα 1 και 2 της σελίδας αυτής.

Το πρώτο βήμα είναι να διαμορφώσουμε το SSL πιστοποιητικό που θα περιέχει τις σωστές παραμέτρους για την ΑΠ, όπως φαίνεται στο αρχείο **openssl.cnf** (Configuration file)

```
[ req ]
default_md = sha1
distinguished_name = req_distinguished_name

[ req_distinguished_name ]
countryName = Country
countryName_default = SK
countryName_min = 2
countryName_max = 2
localityName = Locality
localityName_default = Bratislava
organizationName = Organization
organizationName_default = Linuxconfig Enterprises
commonName = Common Name
commonName_max = 64

[ certauth ]
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always,issuer:always
basicConstraints = CA:true
crlDistributionPoints = @crl

[ server ]
basicConstraints = CA:FALSE
keyUsage = digitalSignature, keyEncipherment, dataEncipherment
extendedKeyUsage = serverAuth
nsCertType = server
crlDistributionPoints = @crl

[ client ]
basicConstraints = CA:FALSE
keyUsage = digitalSignature, keyEncipherment, dataEncipherment
extendedKeyUsage = clientAuth
nsCertType = client
crlDistributionPoints = @crl
```

---

*openssl.cnf*

```
[ crl ]
URI=http://testca.local/ca.crl
```

*openssl.cnf (συνέχεια)*

Στην συνέχεια, δημιουργήσαμε ένα αυτουπογεγραμμένο πιστοποιητικό της ΑΠ εκτελώντας την επόμενη εντολή:

```
openssl req -config ./openssl.cnf -newkey rsa:2048 -nodes -
keyform PEM -keyout ca.key -x509 -days 3650 -extensions
certauth -outform PEM -out ca.cer
```

Επομένως, παράγονται τα αρχεία CA.cer και CA.key, όπως φαίνονται παρακάτω:

```
|-----BEGIN PRIVATE KEY-----
MIIEvAIBADANBgkqhkiG9w0BAQEFAASCCKYwggSiAgEAAoIBAQCn59a5yz4K/s
et
y4I9Nl4CulPyr1CGNG7s4xcferkYobPBYkcPPoiNg+
0fB4ZLIcy/vMMoA9BWRMRT
s2IYfavaS3dKjbHe9rmViTCnkBFiCVpLnIGNUb+AkYeQiaqz1rD/tHy00bt15A
50
ZJDUUttFt3CtNMDjlquX7o+IwOPCve1+Xcgx/tCrNV01QGApV795
+vXWaaYzf1b
ASBzyxTRF/e+R+HPL0Key5KMus8K66asLxWu2ti4oGqMrtPQkel9tKr90f8cGm
UW
WPv2h80bX8dKH3
+g7lUxC+x1GEQFz84ri8KbiiKMRZ7hogore4hkecVvAsho9MNx
hASHp8vJAgMBAAECggEAD6GUir5s5mQLXbzM6/ru50JxGe46nmEs6c5oXYFLfG
C4
FdWa4kSYPNMef52D4eRh0IB3tfbI7fG/9JTbbzE/5Et97bRhmJgP2z0m3j/vcQ
OI
-----END PRIVATE KEY-----
```

*Κομμάτι από το αρχείο ca.key*

```
-----BEGIN CERTIFICATE-----
MIIDgzCCAmugAwIBAgIUVoB8zwICJvQb0MPMt/fuSbjCvHgWdQYJKoZIhvcNAQEF
BQAwHjELMAkGA1UEBhMCRR1Ix DzANBgNVBAMMB1Rlc3RDQTAeFw0yMjE0
MzVaFw0zMjE0MDE0MzVaMB4xCzAJBgNVBAYTAkdSMQ8wDQYDVQQDDAZUZXR0
Q0EwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCn59a5yz4K/sety4I9
N14CulPyr1CGNG7s4xcfcrcYobPBYkcPPoiNg+0fB4ZLIcy/vMMoA9BWRMRTs2IY
favaS3dKjbHe9rmViTCnkBFiCVpLnIGNUb+AkYeQiaqz1rD/tHy00bt15A5OZJDU
UttfFt3CtNMDj1quX7o+IwOPCcve1+Xcgx/tCrNV01QGApV795+vXWaaYzf1bASBz
yxTRF/e+R+HPL0Key5KMus8K66asLxWu2ti4oGqMrtpQke19tKr90f8cGmUWWPv2
h80bX8dKH3+g71UxC+x1GEQFz84ri8KbiiKMRZ7hogore4hkecVvAsho9MNxhASH
p8vJAgMBAAGjgbgwbUwHQYDVR0OBBYEFDAUFLW4aRLBsnWzFcLj+n48RUmBMFkG
A1UdIwRSMFCAFDAUFLW4aRLBsnWzFcLj+n48RUmBoSKkIDAeMQswCQYDVQQGEwJH
UjEPMA0GA1UEAwGVGVzdENBghRWgHzNYgIm9BvQw8y39+5JuMK8eDAMBgNVHRME
BTADAQH/MC sGA1UdHwQkMCiWIKAoByGGmh0dHA6Ly90ZXN0Y2EubG9jYyYwvY2Eu
Y3J3sMA0GCSqGSIb3DQEBBQUAA4IBAQBQw5WfgcBY544qHMutLGzzIC3qYpjdxxG
qiWaIE6a0Cs2Vt1p6co5c8zzB2W0uUGMp/TFkeGQ5dqXH0A2DhMNZwb1MNZH+UFs
qTyQd1WPVmvx2zR2RmhWTow1cgpOPXaZUNYTsV7xUPKvhsR762AozTn1FxHLLG20
W2BX1t1I81c5IvMzoeL8u2g88bXDXpIT9Wg8gNE8FEbuo3kogRN/v1Z80em+ogCq
uJqKJmXfsQgOLwmbHjzT/54rK7xq2vPnzWTy/k/SvMhzXa1bjgr+x19QVPDaYAw8
WY1taMtOZi9Iigmu1uwL+xQQLMyLJivPy9TYka87QkUD8GJIHjY5
-----END CERTIFICATE-----
```

*Αρχείο ca.cer*

## 2. Δημιουργία και πιστοποίηση κλειδιών για server

Αφού ολοκληρωθεί με επιτυχία το ερώτημα 1 και έχοντας φτιάξει το πιστοποιητικό της ΑΠ, θα δημιουργήσουμε ένα ζεύγος κλειδιών για το **web server**, εκτελώντας διαδοχικά τα βήματα 3 έως και 5 του παρακάτω link: <https://linuxconfig.org/apache-web-server-ssl-authentication>

Συγκεκριμένα, εκτελέσαμε τις επόμενες εντολές:

- Εντολή για δημιουργία ιδιωτικού SSL κλειδιού του server

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

και το αποτέλεσμα φαίνεται στην επόμενη εικόνα :

```

-----BEGIN PRIVATE KEY-----
MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBCwggSjAgEAAoIBAQD0Vfw4lr+p6o
ET
H8t5TD7IhE05+
8Dv7GdTdwm/Ew1Oa4S50OXrv+EpkK352jr7t5yU8AHH6rWhkbrW
9FlIyXCHwFGBrWf46f3uKI9Bd/GWixrmYGvaPwLuP2OCrMcyVewldqqStR/Gw8
H3
hDCfPqAzbsTT5LxreR0qdy7q+HnjeKlI1GJog9n8l4ZuwAcbslHnCvjjiSSG1/
rg|
KrEgGyAyn3PaKy2zXuXkL7LftjobpcPkVoWgLfdCc9w0P9x/5uyMvy8hRtfsKr
11
sg1FjEoGJTFvGe3SITFG1DVC+
3vgGbKXFWiWIFYZBHdfLnuGf40rFsN1dryX7B3q
O3cd3G7pAgMBAAECggEAFekZxIHdFvNUu3aAhQcLPdSUwrBZVsRP3wrz5wpmVo
b/
9p4o87nxiVKwNFSglIKDKb4aOSK92tD3mCB/M0T3HPNhLNxwN1YYswstn2Ygg2
UT
Z3ZUj57pFf3vxAYFuiCcv4a5GBBvd7mtXdl/FD+
8ts6uhIv7d3IvGbb4qP7RYDDc

```

*Ιδιωτικό κλειδί του server (server.key)*

Στη συνέχεια φτιάξαμε ένα αίτημα **certificate signing request (csr)** προς την δοκιμαστική ΑΠ ώστε να υπογράψει το πιστοποιητικό του server. Για να γίνει αυτό, εκτελέσαμε το βήμα 4 του link, δηλαδή την εντολή:

```

openssl req -config ./openssl.cnf -new -key server.key -out
server.req

```

και το αρχείο που παρήχθη είναι το ακόλουθο:

```

-----BEGIN CERTIFICATE REQUEST-----
MIICZjCCAU4CAQAwITELMAkGA1UEBhMCU0sxExJABGNVBAAMCwxxY2FsaG9zdDCC
ASIWdQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAPRUXDiWv6nqgRMfy3lMPsiE
TTn7w0/sZ1N3Cb8TDU5rhLnQ5eu/4SmQr-fna0vu3nJTAcfqtaGRutb0WUjJcIfA
UYGtZ/jp/e4oj0F38ZaLGuzGa9o/Au4/Y4KsxzJV7CV2qpK1H8bDwfeEMJ8+oDNu
xNPkvGt5HSp3Lur4eeN4qUjUYmiD2fyXhm7ABxuyUecK+00JJiAX+uAqsSAbIDKf
c9orLbNe5eQvst+20hulw+RWhaAt90Jz3DQ/3H/m7Iy/LyFG1+wqvXWyDUWMSgY1
N9UZ7dIHmUaUNUL7e+AZspcVYiJYVhkEd18ue4Z/jSsWw3V2vJfsHeo7dx3cbukC
AwEAAaAAMA0GCSqGSIb3DQEBBQUAA4IBAQBZhuBzKmIBWjwrglGzbkP3tdB1Lw5+
UGppMIX2CgqPETW0gBI797D4TndNDzoIS5UUpHQugbiqJa0o/ZhZ8kmURwo/iXFJ
X8SHibonD00QFBs+VWmpPw9D0rFrTjxqAGIAPip4pgIMNS0aGX1Av6bubpfHjSeI
C+cB56mImp4W1+XbBQxdTW/HQn9s2TGmq68dommVR3dlzr5AoybpzcsiHXFT1KG5
XSII1CHCZVS2IV30V5UImoWpP/4IurRRkEjBwCTBb2y+exg8jk6QobFmntlmEYhk
evL1EFPFrDPqS1F1v8FnFhU8S9qUV/jGhf2DTHfHN08BWNgSg2bkS61N
-----END CERTIFICATE REQUEST-----

```

*Αρχείο server.req*

Με βάση κάποιες παραμέτρους στο αρχείο διαμόρφωσης της ΑΠ, το πιστοποιητικό του server περιλαμβάνει τα αντίστοιχα constraints (basic constraints, key usage, extended key usage) που αντιστοιχούν σε έναν server. Συγκεκριμένα, στο αρχείο **openssl.cnf**:

```
[ server ]
basicConstraints = CA:FALSE
keyUsage = digitalSignature, keyEncipherment, dataEncipherment
extendedKeyUsage = serverAuth
nsCertType = server
crlDistributionPoints = @crl
```

Το πιστοποιητικό του server που δημιουργείται μετά την υπογραφή του αιτήματος από την ΑΠ φαίνεται στην επόμενη εικόνα, εκτελώντας την εντολή του βήματος 5 από το link:

```
openssl x509 -req -in server.req -CA ca.cer -CAkey ca.key
-set_serial 100 -extfile openssl.cnf -extensions server -
days 365 -outform PEM -out server.cer
```

```
|-----BEGIN CERTIFICATE-----
MIIDazCCA10gAwIBAgIBZDANBgkqhkiG9w0BAQsFADAeMQswCQYDVQQGEwJHUjEP
MA0GA1UEAwVGVVzdENBMB4XDTEyMTIwNDExMjUzOFoXDTEyMTIwNDExMjUzOFow
ITELMAkGA1UEBhMCU0sxETABBgNVBAMMCWxvY2FsaG9zdDCCASIwDQYJKoZIhvcN
AQEBBQADggEPADCCAQoCggEBAPRUXDiWv6nqgRMfy3lMPsiETTn7wO/sZ1N3Cb8T
DU5rhLnQ5eu/4SmQr-fna0vu3nJTwAc-fqtaGRutb0WUjJcIfAUYGtZ/jp/e4oj0F3
8ZaLGuZga9o/Au4/Y4KsxxJV7CV2qpK1H8bDwfeEMJ8+oDNuxNPkvGt5HSp3Lur4
eeN4qUjUYmiD2fyXhm7ABxuyUecK+00JJIAx+uAqsSAbIDKfc9orLbNe5eQvst+2
Ohulw+RWhaAt90Jz3DQ/3H/m7Iy/LyFG1+wqvXWYDUWMSgY1N9UZ7dIhMUaUNUL7
e+AZspcVYiJYVhkEd18ue4Z/jSsWw3V2vJfsHeo7dx3cbukCAwEAaA0B5DCBrTAJ
BgNVHRMEAjAAMAsGA1UdDwQEAwIEsDATBgNVHSEEDAKBggrBgEFBQcDATARBglg
hkgBhvhCAQEEBAMCBkAwKwYDVR0fBCQwIjAgoB6gHIYaaHR0cDovL3Rlc3RjYS5s
b2NhbC9jY55jcmwwHQYDVR0fBBYEFDDXXbssDhJva0zAW9RzByVX4oXuMB8GA1Ud
IwQYMBaAFDafUFLW4aRLBsnWzFcLj+n48RUMBMA0GCSqGSIb3DQEBCwUAA4IBAQA0
UPdH50UrP2w3ET2/vkKd4mRaZCMS3nsasqOcPeRFgJRi5Lv72z8L1P1TX9yEOfQR
uykzgPt2VT3C+Gru7d5Ut/aRRo+BPZGPdaBIEQZ7yGwrDyCwz79PbgDdu0KsDDdg
ONLuX88u3gq6vsNXZLGxQDFdNnF0ad1/azvt5SaKL8TX+/miH20Z6Q8N12x3JzSv
g3tdWQJmQYbGeagHXbA/7ttvVvfSkiOPssD0P0CDwpU7Dof06cw63BisEvvUpiBz
8vJ3wstA/vI2fXE20FONQDXFowWLYODHNTNzeGjFOYmIGBTmjGHgZGHubB930120
3ox/VITrAuCIDkyte6rB
-----END CERTIFICATE-----
```

*Αρχείο server.cer*



**Σημείωση:** Οι εντολές που αφορούν τη δημιουργία του αιτήματος υπογραφής πιστοποιητικού του server από τον CA, έχουν κατάληξη .req στη συγκεκριμένη υλοποίηση και όχι .csr.

### 3. Δημιουργία, πιστοποίηση και ανάκληση κλειδιών

Αρχικά θα δημιουργήσουμε ένα νέο ζευγάρι κλειδιών και θα τα πιστοποιήσουμε μέσω της ΑΠ όπως μας υποδεικνύεται από την εκφώνηση (private.pem και public.pem) (με DSA):

```
C:\Program Files\OpenSSL-Win64\bin\lab>openssl dsaparam -out dsaparam.txt -genkey 1024

C:\Program Files\OpenSSL-Win64\bin\lab>openssl req -new -x509 -keyout private/userprivatekey.pem -out
req: Option -out needs a value
req: Use -help for summary.

C:\Program Files\OpenSSL-Win64\bin\lab>certs/userselfcert.pem -days 365 -newkey dsa:dsaparam.txt -config
'certs' is not recognized as an internal or external command,
operable program or batch file.

C:\Program Files\OpenSSL-Win64\bin\lab>openssl req -new -x509 -keyout private/userprivatekey.pem -out
req: Option -out needs a value
req: Use -help for summary.

C:\Program Files\OpenSSL-Win64\bin\lab>certs/userselfcert.pem -days 365 -newkey dsa:dsaparam.txt -config
'certs' is not recognized as an internal or external command,
operable program or batch file.

C:\Program Files\OpenSSL-Win64\bin\lab>usercnf.txt -sha1

C:\Program Files\OpenSSL-Win64\bin\lab>openssl req -new -x509 -keyout private/userprivatekey.pem -out certs/userselfcert.pem -days 365 -newkey dsa:dsaparam.txt -config usercnf.txt -sha1
Can't load ./rand into RNG
4280000:error:12000079:random number generator:RAND_load_file:Cannot open file:crypto\rand\randfile.c:106:Filename=./rand
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
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) [GB]:GR
State or Province Name (full name) [Attica]:Attica
Locality Name (eg, city) [Athens]:Athens
Organization Name (eg, company) [University of Piraeus]:University of Piraeus
Organizational Unit Name (eg, section) [IT Security Lab]:BisBa
Common Name (eg, YOUR name) [name]:BisBa
Email Address [e-mail@unipi.gr]:pi918@unipi.gr
User ID [id]:1
Cannot write random bytes:
4280000:error:12000079:random number generator:RAND_write_file:Cannot open file:crypto\rand\randfile.c:240:Filename=./rand

C:\Program Files\OpenSSL-Win64\bin\lab>openssl x509 -x509toreq -in certs/userselfcert.pem -signkey
x509: Option -signkey needs a value
x509: Use -help for summary.

C:\Program Files\OpenSSL-Win64\bin\lab>private/userprivatekey.pem -out usercertreq.pem
'private' is not recognized as an internal or external command,
operable program or batch file.

C:\Program Files\OpenSSL-Win64\bin\lab>openssl x509 -x509toreq -in certs/userselfcert.pem -signkey private/userprivatekey.pem -out usercertreq.pem
Enter pass phrase for private/userprivatekey.pem:

C:\Program Files\OpenSSL-Win64\bin\lab>openssl ca -config usercnf.txt -policy policy_anything -out certs/usersignedcert.pem -infiles usercertreq.pem
Using configuration from usercnf.txt
Can't load ./rand into RNG
7C450000:error:12000079:random number generator:RAND_load_file:Cannot open file:crypto\rand\randfile.c:106:Filename=./rand
Could not open file or uri for loading CA private key from private/CAkey.pem
7C450000:error:16000069:STORE routines:ossl_store_get0_loader_int:unregistered scheme:crypto\store\store_register.c:237:scheme=file
7C450000:error:80000002:system library:file_open:No such file or directory:providers\implementations\storemgmt\file_store.c:267:calling stat(private/CAkey.pem)
Cannot write random bytes:
7C450000:error:12000079:random number generator:RAND_write_file:Cannot open file:crypto\rand\randfile.c:240:Filename=./rand

C:\Program Files\OpenSSL-Win64\bin\lab>openssl dsa -in private/userprivatekey.pem -pubout -out public/userpublickey.pem
read DSA key
Enter pass phrase for private/userprivatekey.pem:
Can't open "public/userpublickey.pem" for writing. No such file or directory
FC230000:error:80000003:system library:BIO_new_file:No such process:crypto\bio\bss_file.c:67:calling fopen(public/userpublickey.pem, w)
FC230000:error:10000080:BIO routines:BIO_new_file:no such file:crypto\bio\bss_file.c:75:
```



Αρχεία που παράγονται:

 userprivatekey.pem	12/7/2022 14:51	CMS (S/MIME) File	1 KB
--	-----------------	-------------------	------

 userpublickey.pem	12/7/2022 15:18	CMS (S/MIME) File	1 KB
---	-----------------	-------------------	------

(ακολουθήσαμε τις διαφάνειες από το αρχείο lab-openssl σελίδες 18-19)

Ύστερα μέσω της ΑΠ ανακαλούμε το πιστοποιητικό:

 usercertreq.pem	12/7/2022 14:53	CMS (S/MIME) File	2 KB
---	-----------------	-------------------	------

Και θα το προσθέσουμε στη λίστα (certificate revocation list/CRL):

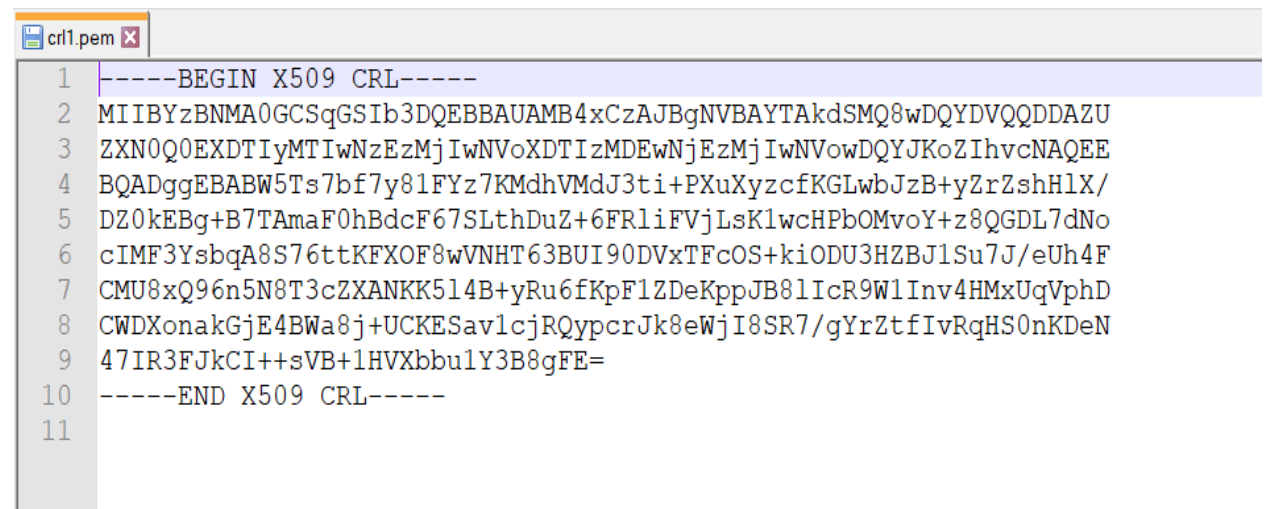
Δημιουργία λίστας και προβολή:

```
C:\Program Files\OpenSSL-Win64\bin\lab> openssl ca -gencrl -out crl/crl1.pem -config CA.cnf.txt
Using configuration from CA.cnf.txt

C:\Program Files\OpenSSL-Win64\bin\lab> openssl crl -in crl/crl1.pem -text
Certificate Revocation List (CRL):
  Version 1 (0x0)
  Signature Algorithm: md5WithRSAEncryption
  Issuer: C = GR, CN = TestCA
  Last Update: Dec  7 13:22:05 2022 GMT
  Next Update: Jan  6 13:22:05 2023 GMT
No Revoked Certificates.
  Signature Algorithm: md5WithRSAEncryption
  Signature Value:
    15:b9:4e:ce:db:7f:bc:bc:d4:56:33:ec:a3:1d:85:53:1d:27:
    7b:62:f8:f5:ee:5f:2c:dc:7c:a1:8b:c1:b2:73:07:ec:99:ad:
    9b:21:1e:55:ff:0d:9d:24:10:18:3e:07:b4:c0:99:a1:74:84:
    17:5c:17:ae:d2:2e:d8:43:b9:9f:ba:15:19:62:15:58:cb:b0:
    ad:70:70:73:db:38:cb:e8:63:ec:fc:40:60:cb:ed:d3:68:70:
    83:05:dd:8b:1b:a8:0f:12:ef:ab:6d:28:55:ce:17:cc:15:34:
    74:fa:dc:15:08:f7:40:d5:c5:31:5c:39:2f:a4:88:e0:d4:dc:
    76:41:27:54:ae:ec:9f:de:52:1e:05:08:c5:3c:c5:0f:7a:9f:
    93:7c:4f:77:19:5c:03:4a:2b:99:78:07:ec:91:bb:a7:ca:a4:
    5d:59:0d:e2:a9:a4:90:7c:94:87:11:f5:6d:48:9e:fe:07:33:
    15:2a:56:98:43:09:60:d7:a2:76:a4:1a:31:38:05:66:bc:8f:
    e5:02:28:44:9a:bf:57:23:45:0c:a9:72:b2:64:f1:e5:a3:23:
    c4:91:ef:f8:18:ad:9b:5f:22:f4:6a:1d:2d:27:28:37:8d:e3:
    b2:11:dc:52:64:08:8f:be:b1:50:7e:d4:75:57:6d:bb:b5:63:
    70:7c:80:51
-----BEGIN X509 CRL-----
MIIBYzBNMA0GCSqGSIb3DQEBAUAMB4xCzAJBgNVBAYTAkdSMQ8wDQYDVQDDAUZU
ZXN0Q0EXDTEyMTIwNzEzMjIwNVoXDTEzMDUwNjEzMjIwNVoDQYJKoZIhvcNAQEE
BQADggEBABW5Ts7bF7y81FYz7KMDhVMDJ3ti+PXuXyzcfKGLwbJzB+yZrZshHlX/
DZ0kEBg+B7TAmaF0hBdcF67SLthDuZ+6FR1iFVjLsK1wcHPbOMvoY+z8QGDL7dNo
cIMF3YsbqA8S76ttKFx0F8wVNHT63BUI90DVxTFcOS+kiODU3HZBJ1Su7J/eUh4F
CMU8xQ96n5N8T3cZXANKK514B+yRu6fKpF1ZDeKppJB81IcR9W1Inv4HMxUqVphD
CWDXonakGjE4Bwa8j+UCKESav1cjRQypcrJk8eWjI8SR7/gYrZtfIvRqHS0nKDeN
47IR3FJkCI++sVB+1HVXbbu1Y3B8gFE=
-----END X509 CRL-----
```

Ανάκληση και προσθήκη πιστοποιητικού στην SRL λίστα:

```
C:\Program Files\OpenSSL-Win64\bin\lab> openssl ca -gencrl -revoke certs/userselfcert.pem -config CAcnf.txt
Using configuration from CAcnf.txt
-----BEGIN X509 CRL-----
MIIBYzBNMA0GCSqGSIb3DQEBAUAMB4xCzAJBgNVBAYTAkdSMQ8wDQYDVQQDDAZU
ZXN0Q0EXDTEyMTIwNzEzMjIwNVoxDTIzMDEwNjEzMjIwNVowDQYJKoZIhvcNAQEE
BQADggEBABW5Ts7bf7y81FYz7KMDhVmdJ3ti+PXuXyzcfKGLwbJzB+yZrZshHlX/
DZ0kEBg+B7TAmAF0hBdcF67SLthDuZ+6FRliFVjLsKlwcHPbOMvoY+z8QGDL7dNo
cIMF3YsbqA8S76ttKFXOF8wVNHT63BUI90DVxTFcOS+kiODU3HZBJ1Su7J/eUh4F
CMU8xQ96n5N8T3cZXANKK514B+yRu6fKpF1ZDeKppJB81IcR9W1Inv4HMxUqVphD
CWDXonakGjE4BWA8j+UCKESavlCjRQypcrJk8eWjI8SR7/gYrZtfIvRqHS0nKDeN
47IR3FJkCI++sVB+1HVXbbulY3B8gFE=
-----END X509 CRL-----
Adding Entry with serial number 0F3B8BDC5D6ADA051A45D2D2C055451D04CE2E87 to DB for /C=GR/ST=Attica/L=Athens/O=University of Piraeus/OU=BisBa/CN=BisBa/emailAddress=p19188@unipi.gr/uid=1
Revoking Certificate 0F3B8BDC5D6ADA051A45D2D2C055451D04CE2E87.
Data Base Updated
```



```
crl1.pem
1 -----BEGIN X509 CRL-----
2 MIIBYzBNMA0GCSqGSIb3DQEBAUAMB4xCzAJBgNVBAYTAkdSMQ8wDQYDVQQDDAZU
3 ZXN0Q0EXDTEyMTIwNzEzMjIwNVoxDTIzMDEwNjEzMjIwNVowDQYJKoZIhvcNAQEE
4 BQADggEBABW5Ts7bf7y81FYz7KMDhVmdJ3ti+PXuXyzcfKGLwbJzB+yZrZshHlX/
5 DZ0kEBg+B7TAmAF0hBdcF67SLthDuZ+6FRliFVjLsKlwcHPbOMvoY+z8QGDL7dNo
6 cIMF3YsbqA8S76ttKFXOF8wVNHT63BUI90DVxTFcOS+kiODU3HZBJ1Su7J/eUh4F
7 CMU8xQ96n5N8T3cZXANKK514B+yRu6fKpF1ZDeKppJB81IcR9W1Inv4HMxUqVphD
8 CWDXonakGjE4BWA8j+UCKESavlCjRQypcrJk8eWjI8SR7/gYrZtfIvRqHS0nKDeN
9 47IR3FJkCI++sVB+1HVXbbulY3B8gFE=
10 -----END X509 CRL-----
11
```

(ακολουθήσαμε τις διαφάνειες από το αρχείο lab-openssl σελίδα 20)







## 4. Εισαγωγή πιστοποιητικού στον server

Σε αυτό το ερώτημα θα χρησιμοποιήσουμε τα αρχεία **server.cer**, **server.key** και **ca.cer**, για να φτιάξουμε ένα αρχείο keystore που περιέχει το πιστοποιητικό του server και του CA (.jks). Το τελευταίο, είναι αυτό που πρέπει να εισάγουμε στον server. Οι εντολές για την δημιουργία του αρχείου .jks βρέθηκαν εδώ:

<https://www.tothenew.com/blog/convert-apache-x509-cert-ssl-certificate-to-tomcat-keystore/>

```
openssl pkcs12 -export -in server.cer -inkey server.key -
certfile ca.cer -out serverkeystore.p12
```

```
keytool -importkeystore -srckeystore serverkeystore.p12
-srcstoretype PKCS12 -destkeystore serverkeystore.jks
```

 ca.cer	4/12/2022 5:24 μμ	Security Certificate	2 KB
 ca.key	4/12/2022 5:23 μμ	Registration Entries	2 KB
 openssl.cnf	4/12/2022 4:34 μμ	CNF File	1 KB
 server.cer	4/12/2022 5:25 μμ	Security Certificate	2 KB
 server.key	4/12/2022 5:24 μμ	Registration Entries	2 KB
 serverkeystore.jks	4/12/2022 5:30 μμ	JKS File	4 KB

Αρχείο .jks (export password = 123456)

Στη συνέχεια μεταβαίνουμε στο φάκελο του Tomcat->conf. Τοποθετούμε το αρχείο serverkeystore.jks και κάνουμε τις παρακάτω αλλαγές στον connector του αρχείου server.xml:

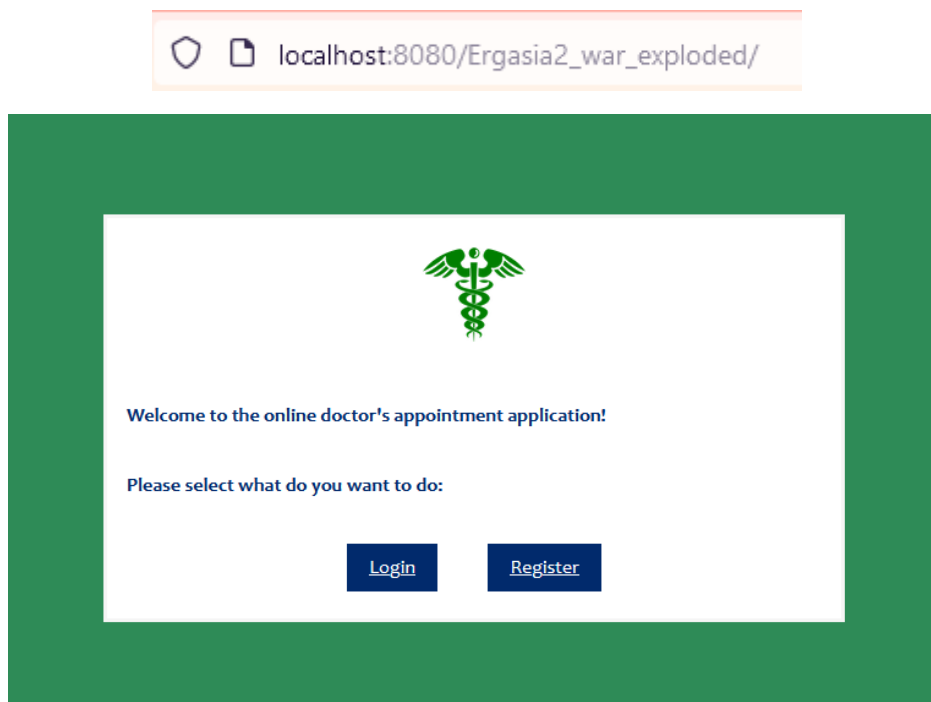
```
<!-- Define an SSL/TLS HTTP/1.1 Connector on port 8443
This connector uses the NIO implementation. The default
SSLImplementation will depend on the presence of the APR/native
library and the useOpenSSL attribute of the
AprLifecycleListener.
Either JSSE or OpenSSL style configuration may be used regardless of
the SSLImplementation selected. JSSE style configuration is used below.
-->

<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
maxThreads="150" scheme="https" secure="true"
keystoreFile="conf\serverkeystore.jks"
keystorePass="123456"
clientAuth="false"/>
```

Αλλαγές στον connector. Πηγή:

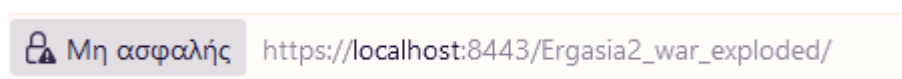
<https://www.tencentcloud.com/document/product/1007/43804>

Μεταβαίνουμε στον φυλλομετρητή αφού εκκινήσουμε τον Tomcat:



*http σύνδεση*

Δοκιμάζουμε την https σύνδεση:



Προειδοποίηση: Πιθανός κίνδυνος ασφαλείας

Το Firefox ανίχνευσε μια πιθανή απειλή ασφαλείας και δεν συνέχισε στο **localhost**. Αν επισκεφθείτε αυτόν τον ιστότοπο, οι εισβολείς ενδέχεται να προσπαθήσουν να υποκλέψουν πληροφορίες, όπως τους κωδικούς πρόσβασης, τα email ή τα στοιχεία των πιστωτικών καρτών σας.

**Τι μπορείτε να κάνετε γι' αυτό;**

Το ζήτημα οφείλεται κατά πάσα πιθανότητα στην ιστοσελίδα και δεν μπορείτε να κάνετε τίποτα για να το διορθώσετε.

Αν είστε σε εταιρικό δίκτυο ή χρησιμοποιείτε λογισμικό anti-virus, μπορείτε να απευθυνθείτε στις ομάδες υποστήριξης για βοήθεια. Μπορείτε επίσης να ειδοποιήσετε το διαχειριστή της ιστοσελίδας για το πρόβλημα.

[Μάθετε περισσότερα...](#)

Επιστροφή (Προτείνεται)

Σύνθετα...

*https σύνδεση*


Πατάμε Σύνθετα --> Προβολή πιστοποιητικού:

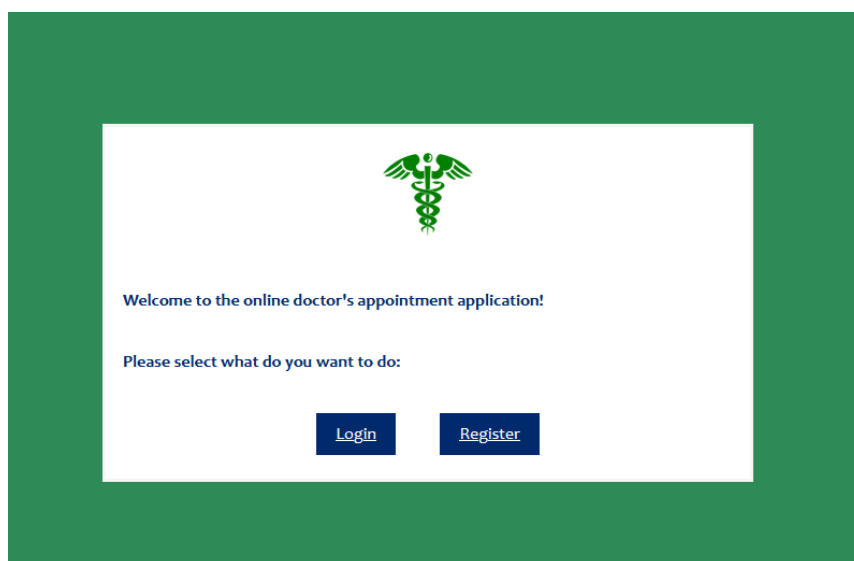
localhost	TestCA
<b>Όνομα θέματος</b>	
Χώρα	SK
Κοινό όνομα	localhost
<b>Όνομα εκδότη</b>	
Χώρα	GR
Κοινό όνομα	TestCA

localhost	TestCA
<b>Όνομα θέματος</b>	
Χώρα	GR
Κοινό όνομα	TestCA
<b>Όνομα εκδότη</b>	
Χώρα	GR
Κοινό όνομα	TestCA

Πιστοποιητικό server και πιστοποιητικό εκδότη CA (Τα χαρακτηριστικά είναι ενδεικτικά)

Επιλέγουμε «αποδοχή κινδύνου και συνέχεια» για να συνεχίσουμε





 [https://localhost:8443/Ergasia2\\_war\\_exploded/](https://localhost:8443/Ergasia2_war_exploded/)



## 5. Διαμόρφωση του server για διπλή αυθεντικοποίηση

Εδώ ακολουθήσαμε τα βήματα 7 έως 10 από την ιστοσελίδα: <https://linuxconfig.org/apache-web-server-ssl-authentication> στην ενότητα «Issuing OpenSSL certificates» για να κάνουμε τα εξής:

- Δημιουργία ιδιωτικού κλειδιού client (αρχείο client.key)
- Δημιουργία αίτησης πιστοποιητικού client προς το CA (αρχείο client.req)
- Υπογραφή της αίτησης του πιστοποιητικού client από το CA και έκδοση του πιστοποιητικού του client (αρχείο client.cer)
- Δημιουργία αρχείου PKCS12 από τα client.key και client.cer.

 client.cer	4/12/2022 8:40 μμ
 client.key	4/12/2022 8:38 μμ
 client.p12	4/12/2022 8:41 μμ
 client.req	4/12/2022 8:39 μμ

**Σημείωση:** Τα προαναφερθέντα αρχεία που δημιουργούνται κατά την εκτέλεση των βημάτων 7 έως 10 από τον σύνδεσμο: <https://linuxconfig.org/apache-web-server-ssl-authentication>.

Για τα επόμενα δύο βήματα αντλήθηκαν πληροφορίες από:  
<https://stackoverflow.com/questions/1552345/tomcat-client-authentication-using-ssl>

Τρέχουμε την εντολή:

```
keytool -import -alias CertAuth -keystore catrustore.jks -file ca.cer,
```

για να δημιουργήσουμε το αρχείο catrustore.jks που θα χρειαστεί να έχει πρόσβαση ο Tomcat Server (το μετακινούμε στη διαδρομή TomcatHome->conf). Έπειτα, τροποποιούμε ξανά τον connector στο αρχείο server.xml του Tomcat για να επιτύχουμε και την αυθεντικοποίηση χρήστη κατά την πρόσβαση στην ιστοσελίδα:

```

<!-- Define an SSL/TLS HTTP/1.1 Connector on port 8443
      This connector uses the NIO implementation. The default
      SSLImplementation will depend on the presence of the APR/native
      library and the useOpenSSL attribute of the
      AprLifecycleListener.
      Either JSSE or OpenSSL style configuration may be used regardless of
      the SSLImplementation selected. JSSE style configuration is used below.
-->

<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
      SSLEnabled="true"
      maxThreads="150" scheme="https" secure="true"
      keystoreFile="conf\serverkeystore.jks"
      keystorePass="123456"
      truststoreFile="conf\catrustore.jks"
      truststorePass="123456"
      clientAuth="true"
      sslProtocol="TLS"
/>

```

*Αλλαγή του server.xml. Πλέον έχουμε αυθεντικοποίηση χρήστη  
ορίζοντας clientAuth="true".*

Για να συνδεθούμε επιτυχώς με τον server (σε https), πρέπει να εγκαταστήσουμε το πιστοποιητικό client.p12 στον browser διαφορετικά έχουμε το παρακάτω αποτέλεσμα:

### Αποτυχία ασφαλούς σύνδεσης

Προέκυψε σφάλμα κατά την σύνδεση στο localhost:8443.  
SSL\_ERROR\_RX\_CERTIFICATE\_REQUIRED\_ALERT

Κωδικός σφάλματος: SSL\_ERROR\_RX\_CERTIFICATE\_REQUIRED\_ALERT

- Η σελίδα που προσπαθείτε να δείτε δεν μπορεί να εμφανιστεί επειδή δεν ήταν δυνατή η επαλήθευση των ληφθέντων δεδομένων.
- Παρακαλώ επικοινωνήστε με τους ιδιοκτήτες του ιστοτόπου για να τους ενημερώσετε σχετικά με αυτό το πρόβλημα.

[Μάθετε περισσότερα...](#)

Δοκιμή ξανά



Σε firefox, μεταβαίνουμε στις Ρυθμίσεις→Απόρρητο και ασφάλεια→Πιστοποιητικά→Προβολή πιστοποιητικών:

Διαχείριση πιστοποιητικών ✕

Τα πιστοποιητικά σας Αποφάσεις ταυτοποίησης Άτομα Διακομιστές Αρχές

Έχετε πιστοποιητικά από αυτούς τους οργανισμούς που σας ταυτοποιούν


Όνομα πιστοποιητικού	Συσκευή ασφάλειας	Σειριακός αριθμός	Λήγει στις	✎
----------------------	-------------------	-------------------	------------	---

Προβολή... Αντίγραφο ασφαλείας... Αντίγραφο όλων... Εισαγωγή... Διαγραφή...

OK

### Παράθυρο διαχείρισης πιστοποιητικών

Επιλέγουμε εισαγωγή και εισάγουμε το πιστοποιητικό client.p12:

Name	Date modified	Type	Size
 client.p12	4/12/2022 8:41 μμ	Personal Informati...	3 KB


ne: client.p12

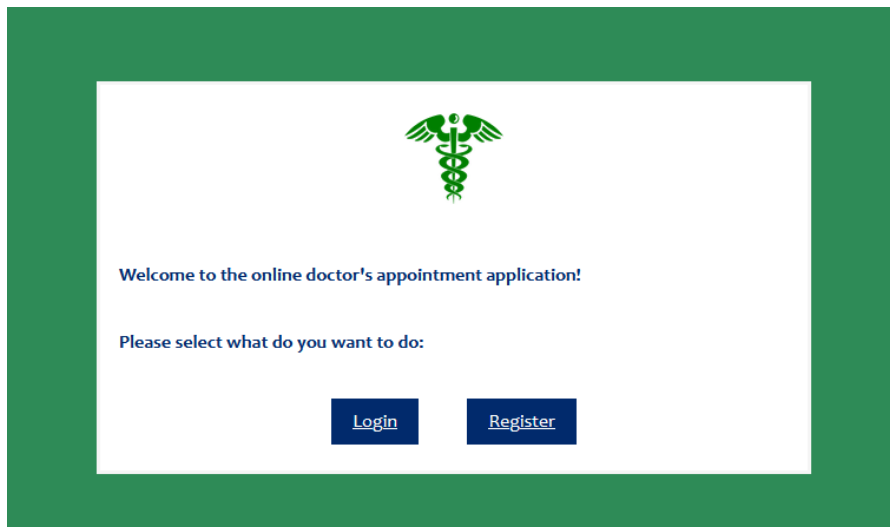
Αρχείο PKCS12 (\*.p12;\*.pfx)

Open Cancel

Όνομα πιστοποιητικού	Συσκευή ασφάλειας	Σειριακός αριθμός	Λήγει στις	✎
Winconfig.org	Software Security Device	65	Δευτέρα, 4 Δεκεμβρί...	

Επιτυχής εισαγωγή πιστοποιητικού client μετά την συμπλήρωση του password

  [https://localhost:8443/Ergasia2\\_war\\_exploded/](https://localhost:8443/Ergasia2_war_exploded/)



*Επιτυχής πρόσβαση μετά την ανανέωση της σελίδας*

Κατά τη σύνδεση ο client λαμβάνει το πιστοποιητικό του server και το επαληθεύει. Αφού το επαληθεύσει, ο client στέλνει το δικό του πιστοποιητικό στον server για να το επαληθεύσει. Μόλις γίνει αμοιβαία επαλήθευση μετά την ανταλλαγή πιστοποιητικών(χειραψία) ο client και ο server ανταλλάζουν πακέτα μέσω ενός κρυπτογραφημένου καναλιού.