ΑΣΦΑΛΕΙΑ ΠΛΗΡΟΦΟΡΙΑΚΩΝ  
ΣΥΣΤΗΜΑΤΩΝ

ΕΡΓΑΣΙΑ 3

**Συντελεστές εργασίας**

Χριστοφορίδης Χαράλαμπος – Π19188

Γεωργιάδης Νικόλαος – π19032

Καρκάνης Ευστράτιος – Π19064

8 δεκεμβριου 2022

**Περιεχόμενα**

[1. Δημιουργία αρχής πιστοποίησης 2](#_Toc121328266)

[2. Δημιουργία και πιστοποίηση κλειδιών για server 4](#_Toc121328267)

[3. Δημιουργία, πιστοποίηση και ανάκληση κλειδιών 5](#_Toc121328268)

[4. Εισαγωγή πιστοποιητικού στον server 9](#_Toc121328269)

[5. Διαμόρφωση του server για διπλή αυθεντικοποίηση 13](#_Toc121328270)

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)

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

openSSL.cnf



openSSL.cnf (συνέχεια)

Στην συνέχεια, δημιουργήσαμε ένα αυτουπογεγραμμένο πιστοποιητικό 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.ser και CA.key, όπως φαίνονται παρακάτω:

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Κομμάτι από το αρχείο ca.key



Αρχείο ca.cer

1. **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 (έχει παρουσιαστεί και παραπάνω, στο ερώτημα 1):

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Ιδιωτικό κλειδί του server

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:

Εικόνα που περιέχει κείμενο, εφημερίδα, έγγραφο, στιγμιότυπο οθόνης

Περιγραφή που δημιουργήθηκε αυτόματα

Αρχείο server.req

Ωστόσο, έγιναν κάποιες διαμορφώσεις στο αρχείο διαμόρφωσης της ΑΠ, ώστε το πιστοποιητικό του server να περιλαμβάνει τα αντίστοιχα constraints (basic constraints, key usage, extended key usage) που αντιστοιχούν σε έναν server. Συγκεκριμένα, με τις παρακάτω προσθήκες στο αρχείο **openssl.cnf,** έχουμε τα σωστά αποτελέσματα:

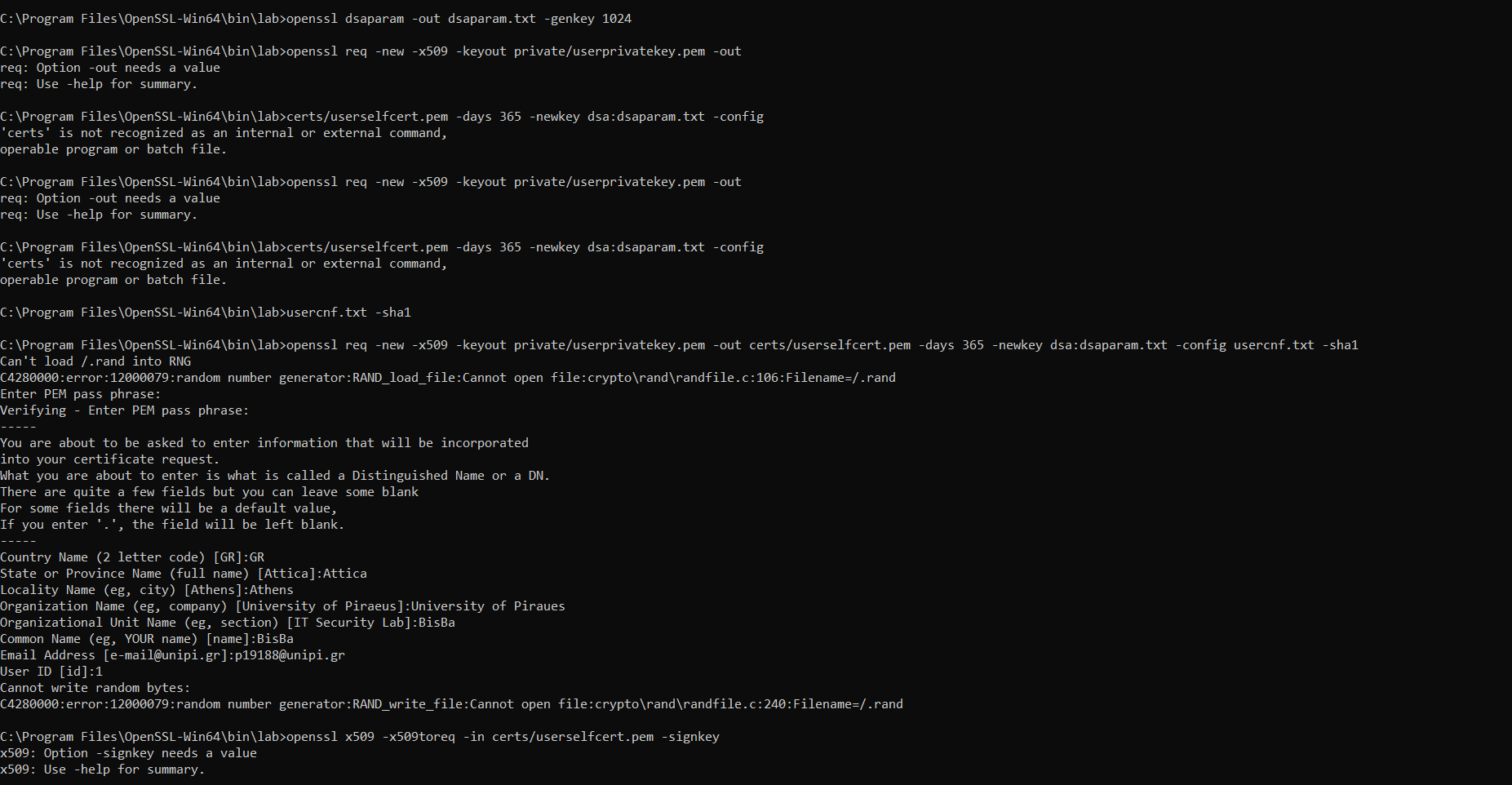
Εικόνα που περιέχει κείμενο

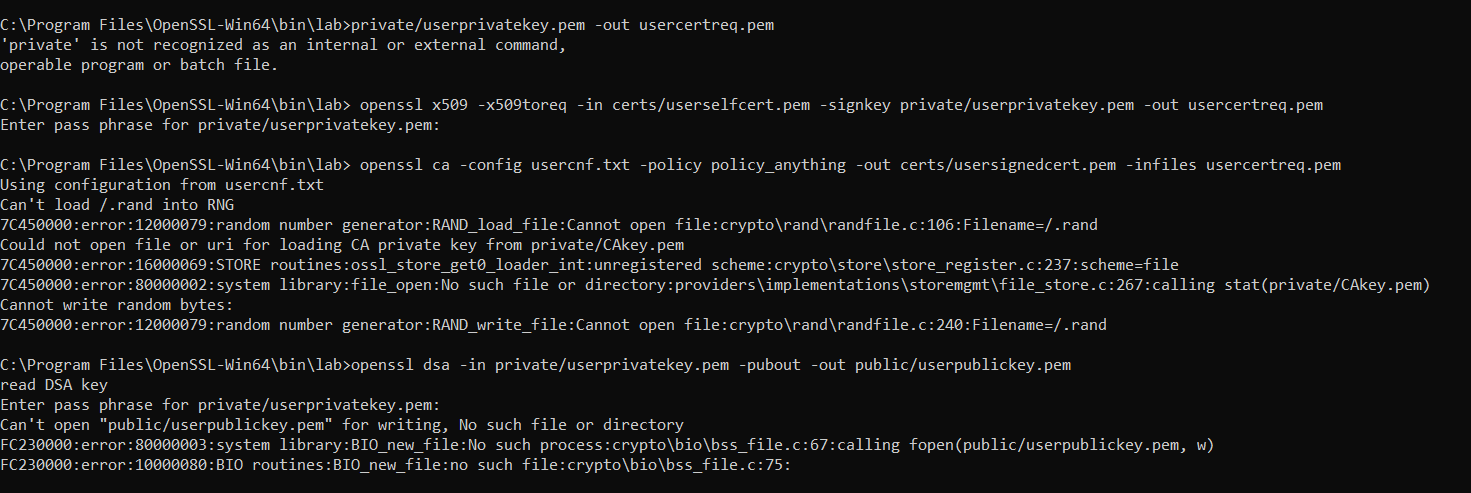
Περιγραφή που δημιουργήθηκε αυτόματα

**Σημείωση:** Οι παραπάνω εντολές έχουν γίνει για τον server Apache Tomcat

1. **Δημιουργία, πιστοποίηση και ανάκληση κλειδιών**

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

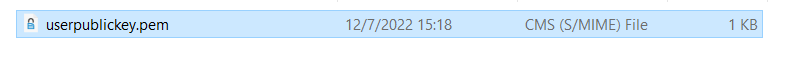




Αρχεία που παράγονται:

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα



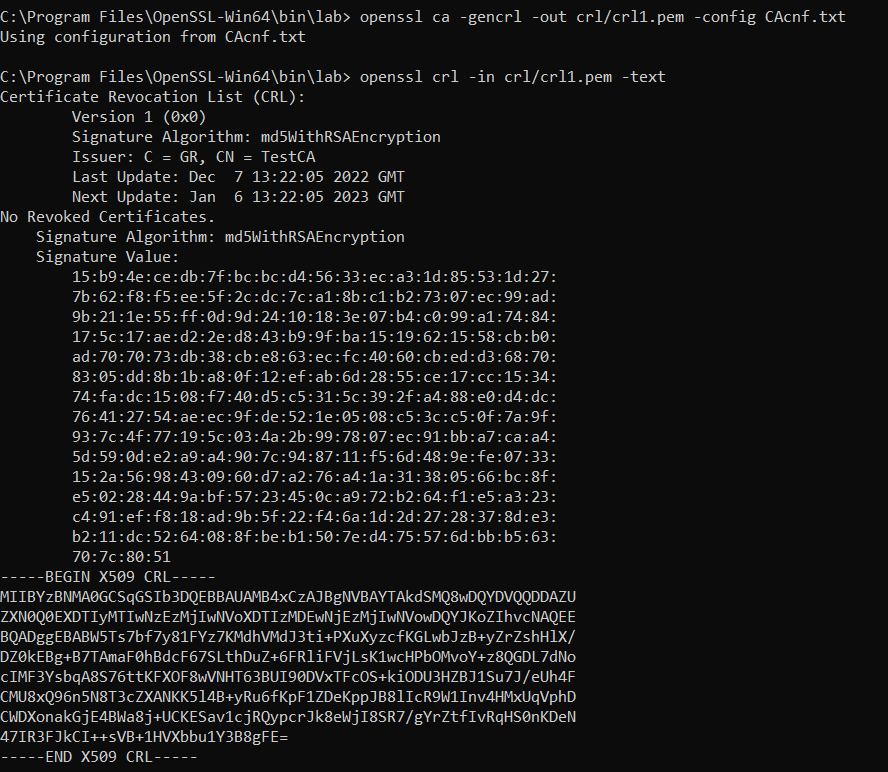
(ακολουθήσαμε τις διαφάνειες από το αρχείο lab-openssl σελίδες 18-19)

Ύστερα μέσω της ΑΠ ανακαλούμε το πιστοποιητικό:

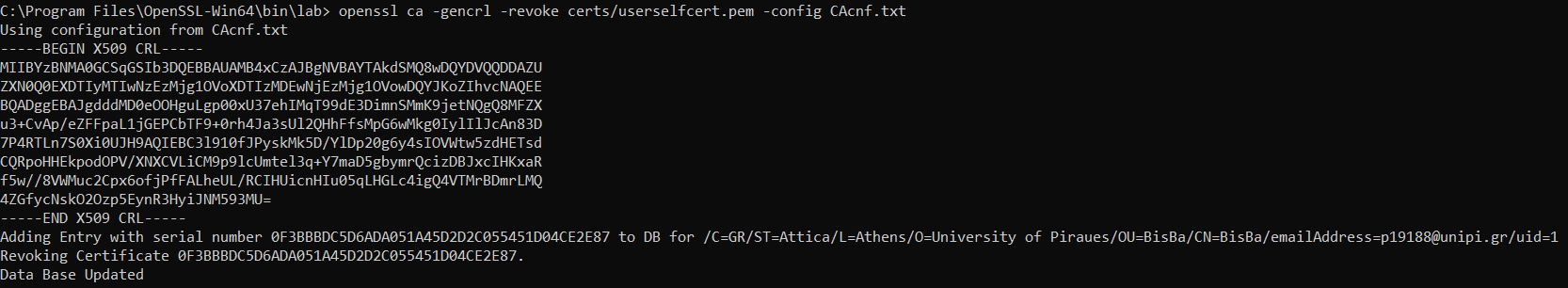


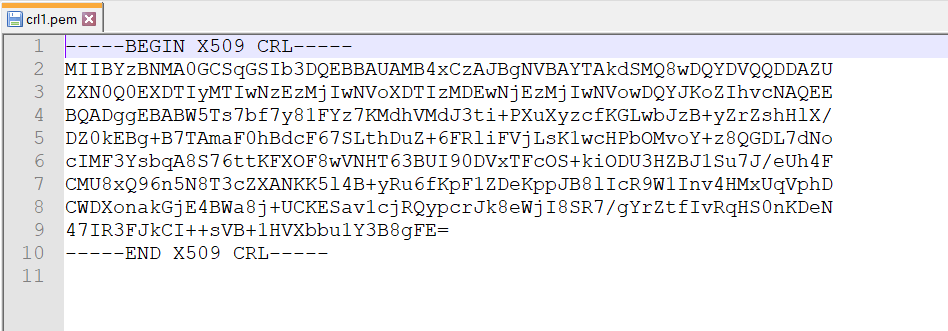
Και θα το προσθέσουμε στη λίστα (certificate revocation list/CRL):

Δημιουργία λίστας και προβολή:



Ανάκληση και προσθήκη πιστοποιητικού στην SRL λίστα:





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

1. **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**

Graphical user interface, text, application

Description automatically generated

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:

Text, letter

Description automatically generated

Αλλαγές στον connector. Πηγή: https://www.tencentcloud.com/document/product/1007/43804

Μεταβαίνουμε στον φυλλομετρητή αφού εκκινήσουμε τον Tomcat:



Graphical user interface, application

Description automatically generated

http σύνδεση

Δοκιμάζουμε την https σύνδεση:



Graphical user interface, text, application, email

Description automatically generated

https σύνδεση

Πατάμε Σύνθετα --> Προβολή πιστοποιητικού:

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

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

Επιλέγουμε «αποδοχή κινδύνου και συνέχεια» για να συνεχίσουμε



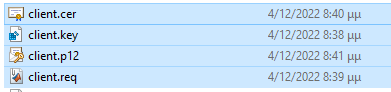
Graphical user interface, application

Description automatically generated

1. **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.



**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

Tρέχουμε 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:

Text, letter

Description automatically generated

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:

Graphical user interface, text, application, email

Description automatically generated

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

Graphical user interface, text, application, email

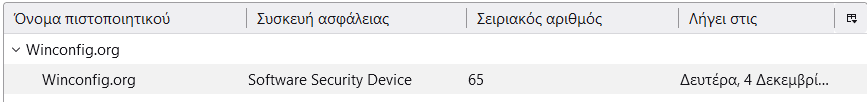
Description automatically generated

Παράθυρο διαχείρισης πιστοποιητικών

Επιλέγουμε εισαγωγή και εισάγουμε το πιστοποιητικό client.p12:

Graphical user interface, application

Description automatically generated



Επιτυχής εισαγωγή πιστοποιητικού client μετά την συμπλήρωση του password



Graphical user interface, application

Description automatically generated

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.