



**Πανεπιστήμιο Δυτικής Αττικής
Σχολή Μηχανικών
Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών**

**ΕΡΓΑΣΤΗΡΙΟ ΑΣΦΑΛΕΙΑ ΣΤΗΝ ΤΕΧΝΟΛΟΓΙΑ ΤΗΣ ΠΛΗΡΟΦΟΡΙΑΣ
ΑΣΦ05 - Εργασία 4 (PKI)
ΦΙΛΙΠΠΟΣ ΠΑΠΑΓΕΩΡΓΙΟΥ - 21390174**

ΗΜΕΡΟΜΗΝΙΑ ΠΑΡΑΔΟΣΗΣ:

**Κυριακή 16 Ιουνίου 2024 - 11:55 μ.μ.
(απομένουν 18 ημέρες 2 ώρες 14 λεπτά)**

ΟΜΑΔΑ ΕΡΓΑΣΤΗΡΙΟΥ:

ΑΣΦ05 - ΤΕΤΑΡΤΗ 11:00 - 13:00

Υπευθύνος Ομάδας:

ΓΕΩΡΓΟΥΛΑΣ ΑΓΓΕΛΟΣ

ΤΙ ΘΑ ΔΟΥΜΕ ΣΤΗΝ ΕΡΓΑΣΙΑ

Σε αυτή την εργασία θα εξετάσουμε την έννοια και τη λειτουργία των Υποδομών Δημόσιου Κλειδιού (Public Key Infrastructure - PKI), οι οποίες αποτελούν το θεμέλιο της ασφαλούς επικοινωνίας στο διαδίκτυο. Συγκεκριμένα, θα μελετήσουμε και θα εφαρμόσουμε τις διαδικασίες δημιουργίας μιας Αρχής Πιστοποίησης (Certification Authority - CA), την έκδοση ψηφιακών πιστοποιητικών, και τη χρήση αυτών των πιστοποιητικών για τη διασφάλιση της επικοινωνίας σε έναν HTTPS server. Επιπλέον, θα προσομοιώσουμε επιθέσεις τύπου Man-In-The-Middle (MITM) για να κατανοήσουμε πώς οι PKI μπορούν να αποτρέψουν τέτοιες επιθέσεις και να διερευνήσουμε τις συνέπειες μιας παραβίασης μιας CA. Μέσα από τις δραστηριότητες και τις ερωτήσεις της άσκησης, θα αποκτήσουμε πρακτική εμπειρία στην υλοποίηση και χρήση των PKI.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΣΕΝΑΡΙΟ.....	4
--------------	---

Δραστηριότητα 1: Δημιουργία Αρχής Πιστοποίησης	5
Δραστηριότητα 2: Έκδοση πιστοποιητικού για πελάτη	7
Δραστηριότητα 3: Χρήση του πιστοποιητικού σε δοκιμαστικό HTTPS Server	10
Δραστηριότητα 4: Χρήση του πιστοποιητικού σε Apache HTTPS Web Server	14
Δραστηριότητα 5: Επίθεση τύπου Man-In-The-Middle	16

ΣΕΝΑΡΙΟ

Το σενάριο της άσκησης προϋποθέτει τη δημιουργία μιας Αρχής Πιστοποίησης η οποία θα εκδώσει πιστοποιητικό για έναν πελάτη (website). Στη συνέχεια ο πελάτης θα χρησιμοποιήσει το πιστοποιητικό αυτό για να δημιουργήσει έναν ασφαλή ιστότοπο. Τα στοιχεία των δυο αυτών οντοτήτων που χρησιμοποιούμε στην επίδειξη των δραστηριοτήτων της άσκησης στην παρούσα εκφώνηση είναι τα κάτωθι:

Αρχή Πιστοποίησης	
C (Country):	GR
ST (State):	ATTICA
L (Locality):	AIGALEO
O (Organization):	UNIWA
OU (Organizational Unit):	ICE
CN (Common Name):	ice21390174.uniwa.gr
EMAIL (Email Address):	Ice21390174@uniwa.gr

Πελάτης	
C (Country):	GR
ST (State):	ATTICA
L (Locality):	ATHENS
O (Organization):	TAILORMADE
OU (Organizational Unit):	TAILOR
CN (Common Name):	tailormade.com
EMAIL (Email Address):	tailormade@info.com

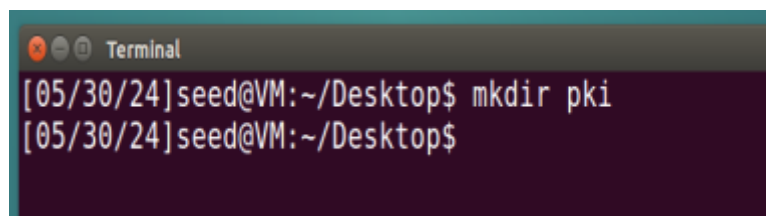
Δραστηριότητα 1: Δημιουργία Αρχής Πιστοποίησης

Σε αυτή τη δραστηριότητα, θα δημιουργήσουμε τη δική μας Αρχή Πιστοποίησης (Certification Authority - CA) χρησιμοποιώντας το OpenSSL. Η Αρχή Πιστοποίησης είναι μια έμπιστη οντότητα που εκδίδει ψηφιακά πιστοποιητικά, τα οποία χρησιμοποιούνται για την επαλήθευση της ταυτότητας των χρηστών και των υπηρεσιών στο διαδίκτυο.

Συγκεκριμένα, θα ακολουθήσουμε τα εξής βήματα:

1. Δημιουργία Καταλόγου Root για την ΑΠ: Θα δημιουργήσουμε τη δομή καταλόγων και αρχείων που απαιτείται για την ΑΠ.
2. Διαμόρφωση της ΑΠ: Θα ρυθμίσουμε το αρχείο διαμόρφωσης του OpenSSL για τη σωστή λειτουργία της ΑΠ.
3. Κατασκευή Αυτό-υπογεγραμμένου Πιστοποιητικού της ΑΠ: Θα δημιουργήσουμε ένα αυτό-υπογεγραμμένο πιστοποιητικό για την ΑΠ μας, το οποίο θα χρησιμοποιηθεί ως το ριζικό πιστοποιητικό.

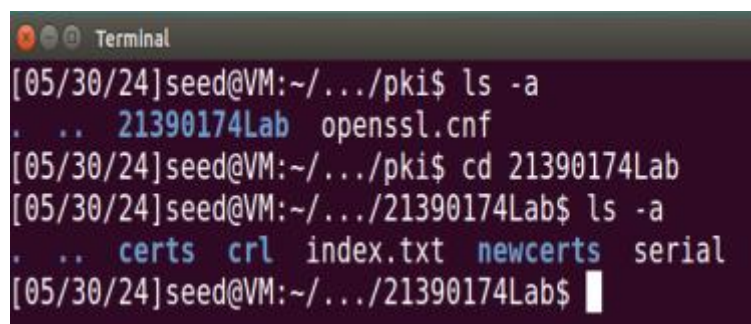
Βήμα 1: Δημιουργία καταλόγου root για την ΑΠ



```
Terminal
[05/30/24]seed@VM:~/Desktop$ mkdir pki
[05/30/24]seed@VM:~/Desktop$
```

1.Είκονα δημιουργία φάκελου pki

Βήμα 2: Διαμόρφωση της ΑΠ



```
Terminal
[05/30/24]seed@VM:~/.../pki$ ls -a
.  ..  21390174Lab  openssl.cnf
[05/30/24]seed@VM:~/.../pki$ cd 21390174Lab
[05/30/24]seed@VM:~/.../21390174Lab$ ls -a
.  ..  certs  crl  index.txt  newcerts  serial
[05/30/24]seed@VM:~/.../21390174Lab$
```

2.Είκονα δημιουργία φάκελων μέσα pki

```

# For the CA policy
policy_anything ]
countryName = match
stateOrProvinceName = match
organizationName = match
organizationalUnitName = optional
commonName = supplied
emailAddress = optional

# For the 'anything' policy
# At this point in time, you must list all ac
# types.
[ policy_anything ]
countryName = optional
stateOrProvinceName = optional
localityName = optional
organizationName = optional
organizationalUnitName = optional
commonName = supplied
emailAddress = optional

```

3.Εικόνα διαμόρφωση openssl.cnf file

Βήμα 3: Κατασκευή αυτό-υπογεγραμμένου πιστοποιητικού της ΑΠ

```

days 365 -config openssl.cnf
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to 'ca.key'
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) [AU]:GR
State or Province Name (full name) [Some-State]:ATTICA
Locality Name (eg, city) []:AIGALEO
Organization Name (eg, company) [Internet Widgits Pty Ltd]:UNIWA
Organizational Unit Name (eg, section) []:ICE
Common Name (e.g. server FQDN or YOUR name) []:ice21390174.uniwa.gr
Email Address []:ice21390174@uniwa.gr
[05/30/24]seed@VM:~/.../pki$

```

4.Εικόνα Δημιουργία αυτό-υπογεγραμμένου πιστοποιητικού

Δραστηριότητα 2: Έκδοση πιστοποιητικού για πελάτη

Σε αυτή τη δραστηριότητα, θα εκδώσουμε και θα υπογράψουμε ψηφιακά πιστοποιητικά για πελάτες χρησιμοποιώντας την Αρχή Πιστοποίησης (CA) που δημιουργήσαμε στην προηγούμενη δραστηριότητα. Ο πελάτης θα είναι ένας ιστότοπος με το όνομα `firstname-lastname.gr`, όπου `firstname` είναι το όνομά σας και `lastname` το επώνυμό μας (και τα δύο με λατινικούς χαρακτήρες). Ακολουθούμε τα εξής βήματα:

Βήμα 1: Δημιουργία ζεύγους δημόσιου/ιδιωτικού κλειδιού για τον πελάτη.

```
05/30/24]seed@VM:~/.../pki$ ls
1390174Lab ca.crt ca.key openssl.cnf
05/30/24]seed@VM:~/.../pki$ openssl genrsa -aes128 -out server.key 1024
Generating RSA private key, 1024 bit long modulus
.....++++++
++++++
e is 65537 (0x10001)
Enter pass phrase for server.key:
Verifying - Enter pass phrase for server.key:
05/30/24]seed@VM:~/.../pki$ ls
1390174Lab ca.crt ca.key openssl.cnf server.key
05/30/24]seed@VM:~/.../pki$
```

5.Εύκονα Δημιουργία ζεύγους κλειδιών RSA για τον πελάτη

```
[05/30/24]seed@VM:~/.../pki$ openssl rsa -in server.key -text
Enter pass phrase for server.key:
Private-Key: (1024 bit)
modulus:
 00:d4:7f:42:1b:05:ce:ac:a5:7d:6f:56:c9:67:13:
 96:c1:ea:54:d1:54:c7:59:50:54:86:63:c0:93:af:
 7d:2a:a0:32:93:79:f1:c6:b3:3d:ba:42:b7:87:71:
 86:bf:b8:2a:b5:e4:9f:da:e7:93:d8:ce:51:7d:84:
 bf:7e:2f:d0:5f:6f:59:07:c6:a1:58:5a:cf:27:92:
 1d:fb:b5:c3:29:63:a2:2f:ce:b5:26:4f:55:ff:cb:
 3a:2e:f6:95:fe:15:29:92:92:0d:dc:f0:4c:53:01:
 b3:57:ce:c3:b7:0d:07:cf:c2:be:dc:01:2a:b6:74:
 b3:47:77:67:0f:95:16:10:55
publicExponent: 65537 (0x10001)
privateExponent:
 00:88:0b:69:61:2f:98:a0:03:db:88:ba:c1:7a:d2:
 99:c4:50:a8:38:d4:d4:44:24:1f:d9:62:67:da:a5:
 91:b7:06:95:1a:c2:09:be:b2:e6:a4:27:2e:a4:f2:
 53:d8:ce:a9:d9:86:23:a9:dc:75:55:6a:d1:d5:50:
 75:e0:ed:ef:b2:df:19:45:38:d5:d0:d2:da:ab:ea:
 90:b1:b4:ce:96:ff:97:d5:f3:b9:df:26:0a:96:8d:
 34:a3:0e:91:b2:45:a9:4c:1b:8e:66:3d:82:19:ce:
 23:c5:88:2a:b7:98:31:cf:5a:3f:f4:d5:80:7b:58:
```

6.Εύκονα Έλεγχος δημιουργίας ζεύγους κλειδιών RSA για τον πελάτη

Βήμα 2: Δημιουργία αιτήματος υπογραφής πιστοποιητικού (CSR).

```
[05/30/24]seed@VM:~/.../pki$ openssl req -new -key server.key -out server.csr -c
onfig openssl.cnf
Enter pass phrase for server.key:
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]:GR
State or Province Name (full name) [Some-State]:ATTICA
Locality Name (eg, city) []:ATHENS
Organization Name (eg, company) [Internet Widgits Pty Ltd]:TAILORMADE
Organizational Unit Name (eg, section) []:TAILOR
Common Name (e.g. server FQDN or YOUR name) []:TailorMadeRoaster
Email Address []:Tailormade@info.gr

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:filippos
An optional company name []:TailorMade
[05/30/24]seed@VM:~/.../pki$
```

7.Είκονα Δημιουργία του CSR

Σημείωση 1:

Η εντολή για τη δημιουργία του αιτήματος υπογραφής πιστοποιητικού (CSR) είναι παρόμοια με την εντολή για τη δημιουργία του αυτο-υπογεγραμμένου πιστοποιητικού για την ΑΠ. Η κύρια διαφορά είναι η επιλογή `-x509`. Χωρίς αυτήν, η εντολή δημιουργεί ένα αίτημα (CSR), ενώ με αυτήν δημιουργεί ένα αυτο-υπογεγραμμένο πιστοποιητικό.

Σημείωση 2:

Κατά τη δημιουργία ενός CSR, ζητούνται επιπλέον πεδία όπως το "challenge password" και το "optional company name". Το "challenge password" δεν είναι το ίδιο με το passphrase για την κρυπτογράφηση του ιδιωτικού κλειδιού. Είναι μια κοινή γνωστή φράση μεταξύ του πελάτη και της ΑΠ για επιπλέον έλεγχο ταυτότητας.


```
[05/30/24]seed@VM:~/.../pki$ openssl req -text -noout -verify -in server.csr
verify OK
Certificate Request:
Data:
  Version: 0 (0x0)
  Subject: C=GR, ST=ATTICA, L=ATHENS, O=TAILORMADE, OU=TAILOR, CN=TaylorMadeRoaster/emailAddress=Tailormade@info.gr
  Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
    Public-Key: (1024 bit)
    Modulus:
      00:d4:7f:42:1b:05:ce:ac:a5:7d:6f:56:c9:67:13:
      96:c1:ea:54:d1:54:c7:59:50:54:86:63:c0:93:af:
      7d:2a:a0:32:93:79:f1:c6:b3:3d:ba:42:b7:87:71:
      86:bf:b8:2a:b5:e4:9f:da:e7:93:d8:ce:51:7d:84:
      bf:7e:2f:d0:5f:6f:59:07:c6:a1:58:5a:cf:27:92:
      1d:fb:b5:c3:29:63:a2:2f:ce:b5:26:4f:55:ff:cb:
      3a:2e:f6:95:fe:15:29:92:92:0d:dc:f0:4c:53:01:
      b3:57:ce:c3:b7:0d:07:cf:c2:be:dc:01:2a:b6:74:
      b3:47:77:67:0f:95:16:10:55
    Exponent: 65537 (0x10001)
  Attributes:
    challengePassword      :unable to print attribute
    unstructuredName       :unable to print attribute
  Signature Algorithm: sha256WithRSAEncryption
  6e:a0:4a:7d:18:78:57:62:ab:4a:26:45:f9:7e:4e:4b:59:ed:
  43:10:97:16:68:76:38:e3:e0:ad:ec:d3:61:60:f6:93:ce:92:
  a5:8a:27:a3:3e:fe:f4:a4:91:e9:e6:8e:58:54:ba:f6:d2:99:
  22:7b:ab:b5:d6:cc:2d:55:1b:97:3d:33:f7:71:1e:14:98:a1:
  e7:c4:c4:1d:5f:d0:70:d0:10:77:64:ff:5a:c1:5d:e3:bb:e2:
  32:f9:7a:52:b2:5f:41:3b:ab:1e:51:05:1e:15:78:22:30:f1:
  52:5a:8a:b1:58:66:ad:d2:f0:1f:48:15:8c:10:03:96:a5:92:
```

8.Είкона Έλεγχος δημιουργίας CSR

Βήμα 3: Έκδοση πιστοποιητικού για τον πελάτη

```
Signature ok
Certificate Details:
  Serial Number: 1 (0x1)
  Validity
    Not Before: May 30 10:18:55 2024 GMT
    Not After : May 30 10:18:55 2025 GMT
  Subject:
    countryName           = GR
    stateOrProvinceName   = ATTICA
    localityName          = ATHENS
    organizationName       = TAILORMADE
    organizationalUnitName = TAILOR
    commonName             = TaylorMadeRoaster
    emailAddress           = Tailormade@info.gr
  X509v3 extensions:
    X509v3 Basic Constraints:
      CA:FALSE
    Netscape Comment:
      OpenSSL Generated Certificate
    X509v3 Subject Key Identifier:
      D9:A2:86:39:40:F9:AD:14:5B:D4:79:80:B3:58:8A:73:89:B2:DF:50
    X509v3 Authority Key Identifier:
      keyid:C5:6B:FD:83:F9:68:E9:BA:F8:F7:BD:AB:4A:3A:33:D7:70:0E:57:70

Certificate is to be certified until May 30 10:18:55 2025 GMT (365 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
[05/30/24]seed@VM:~/.../pki$
```

9.Είкона Υπογραφή του CSR από την ΑΠ

```
[05/30/24]seed@VM:~/../pki$ openssl x509 -in server.crt -text -noout
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1 (0x1)
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=GR, ST=ATTICA, L=AIGALEO, O=UNIWA, OU=ICE, CN=ice21390174.uniwa.gr/emailAddress=ice21390174@uniwa.gr
        Validity
            Not Before: May 30 10:18:55 2024 GMT
            Not After : May 30 10:18:55 2025 GMT
        Subject: C=GR, ST=ATTICA, L=ATHENS, O=TAILORMADE, OU=TAILOR, CN=TailorMadeRoaster/emailAddress=Tailormade@info.gr
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            Public-Key: (1024 bit)
            Modulus:
                00:d4:7f:42:1b:05:ce:ac:a5:7d:6f:56:c9:67:13:
                96:c1:ea:54:d1:54:c7:59:50:54:86:63:c0:93:af:
                7d:2a:a0:32:93:79:f1:c6:b3:3d:ba:42:b7:87:71:
                86:bf:b8:2a:b5:e4:9f:da:e7:93:d8:ce:51:7d:84:
                bf:7e:2f:d0:5f:6f:59:07:c6:a1:58:5a:cf:27:92:
                1d:fb:b5:c3:29:63:a2:2f:ce:b5:26:4f:55:ff:cb:
                3a:2e:f6:95:fe:15:29:92:92:0d:dc:f0:4c:53:01:
                b3:57:ce:c3:b7:0d:07:cf:c2:be:dc:01:2a:b6:74:
                b3:47:77:67:0f:95:16:10:55
            Exponent: 65537 (0x10001)
    X509v3 extensions:
```

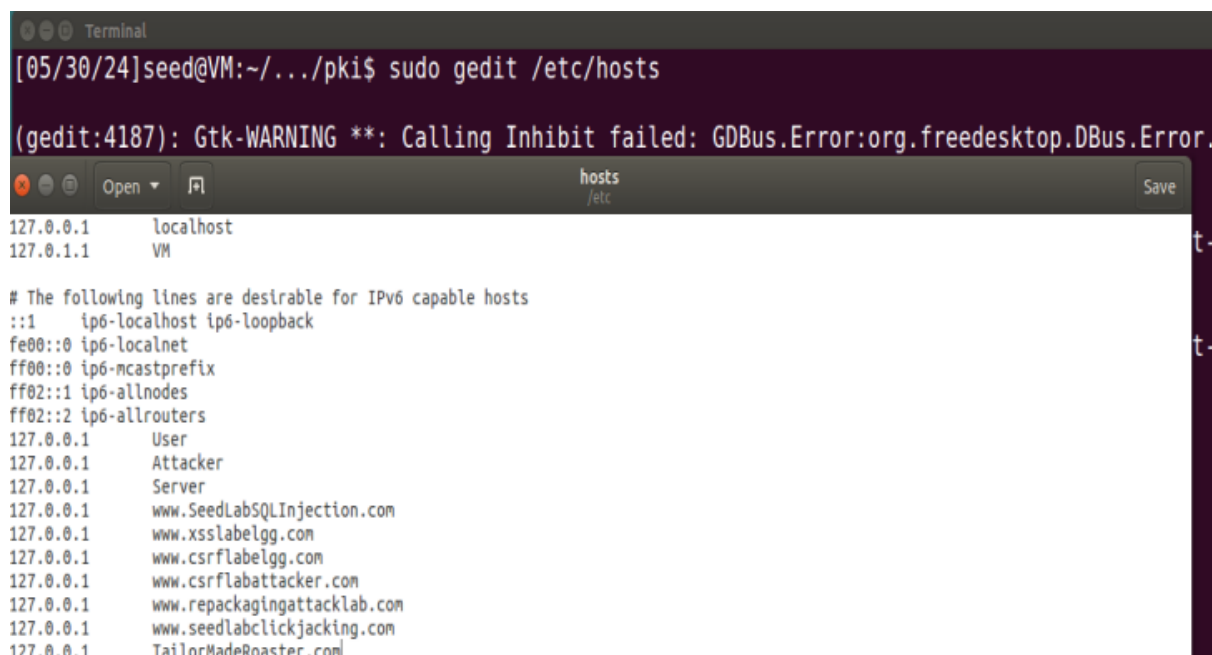
10.Είκονα

Έλεγχος δημιουργίας πιστοποιητικού πελάτη

Δραστηριότητα 3:
Χρήση του πιστοποιητικού σε δοκιμαστικό HTTPS Server
Σε αυτή τη δραστηριότητα, θα διερευνήσουμε πώς χρησιμοποιούνται στην

πράξη τα πιστοποιητικά από τους ιστότοπους (websites) ώστε να προσφέρουν ασφαλή περιήγηση στους χρήστες τους. Θα χρησιμοποιήσουμε τον ενσωματωμένο HTTPS web server που παρέχει το OpenSSL για τις ανάγκες της δοκιμής.

Βήμα 1: Διαμόρφωση DNS



```
[05/30/24]seed@VM:~/../pki$ sudo gedit /etc/hosts
(gedit:4187): Gtk-WARNING **: Calling Inhibit failed: GDBus.Error:org.freedesktop.DBus.Error.

hosts
/etc

127.0.0.1    localhost
127.0.1.1    VM

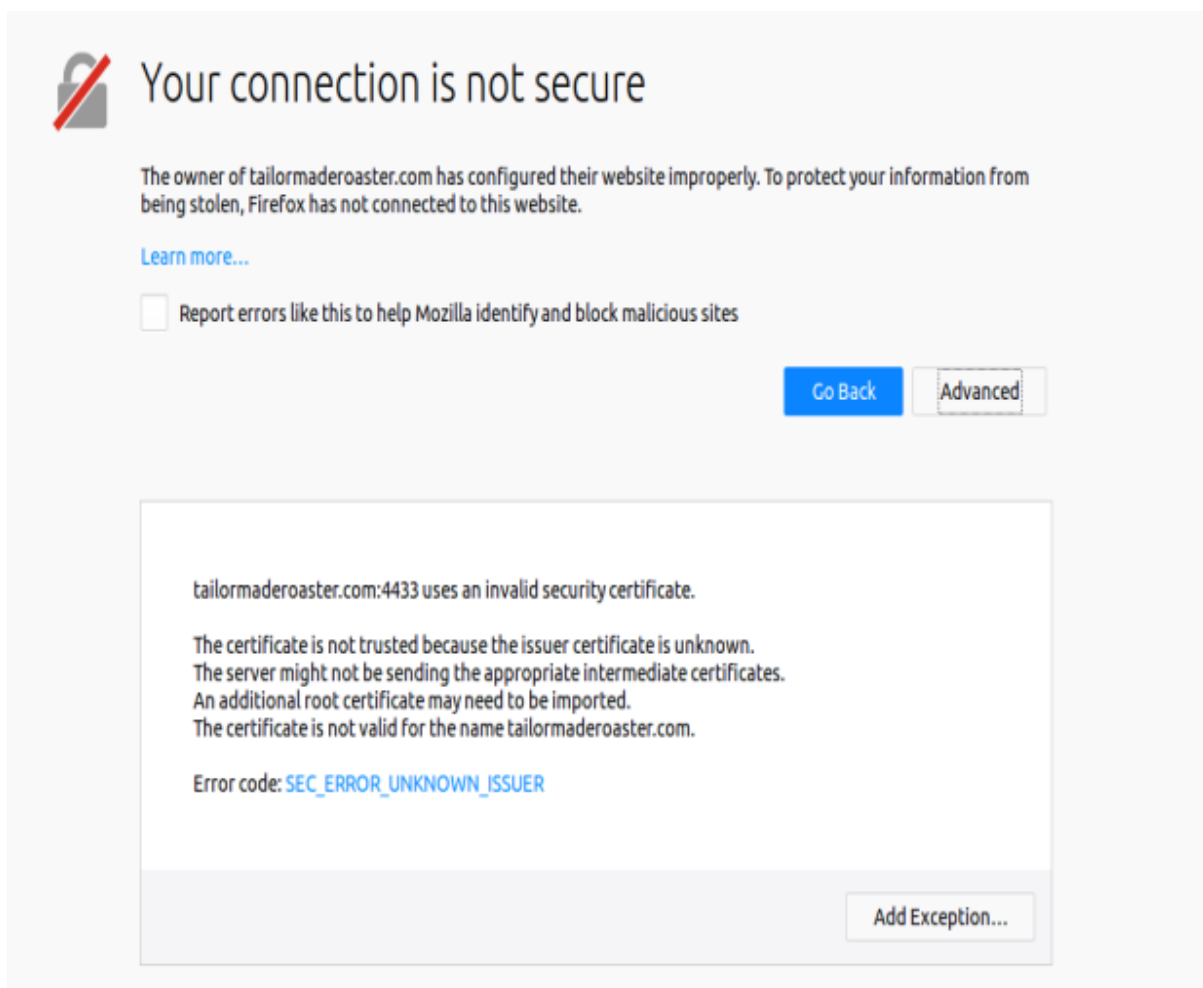
# The following lines are desirable for IPv6 capable hosts
::1         ip6-localhost ip6-loopback
fe00::0     ip6-localnet
ff00::0     ip6-mcastprefix
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters
127.0.0.1   User
127.0.0.1   Attacker
127.0.0.1   Server
127.0.0.1   www.SeedLabSQLInjection.com
127.0.0.1   www.xsslabelgg.com
127.0.0.1   www.csrflabelgg.com
127.0.0.1   www.csrflabattacker.com
127.0.0.1   www.repackagingattacklab.com
127.0.0.1   www.seedlabclickjacking.com
127.0.0.1   TailorMadeRoaster.com
```

11.Είκονα Διαμόρφωση του DNS

Βήμα 2: Διαμόρφωση του web server και δοκιμή

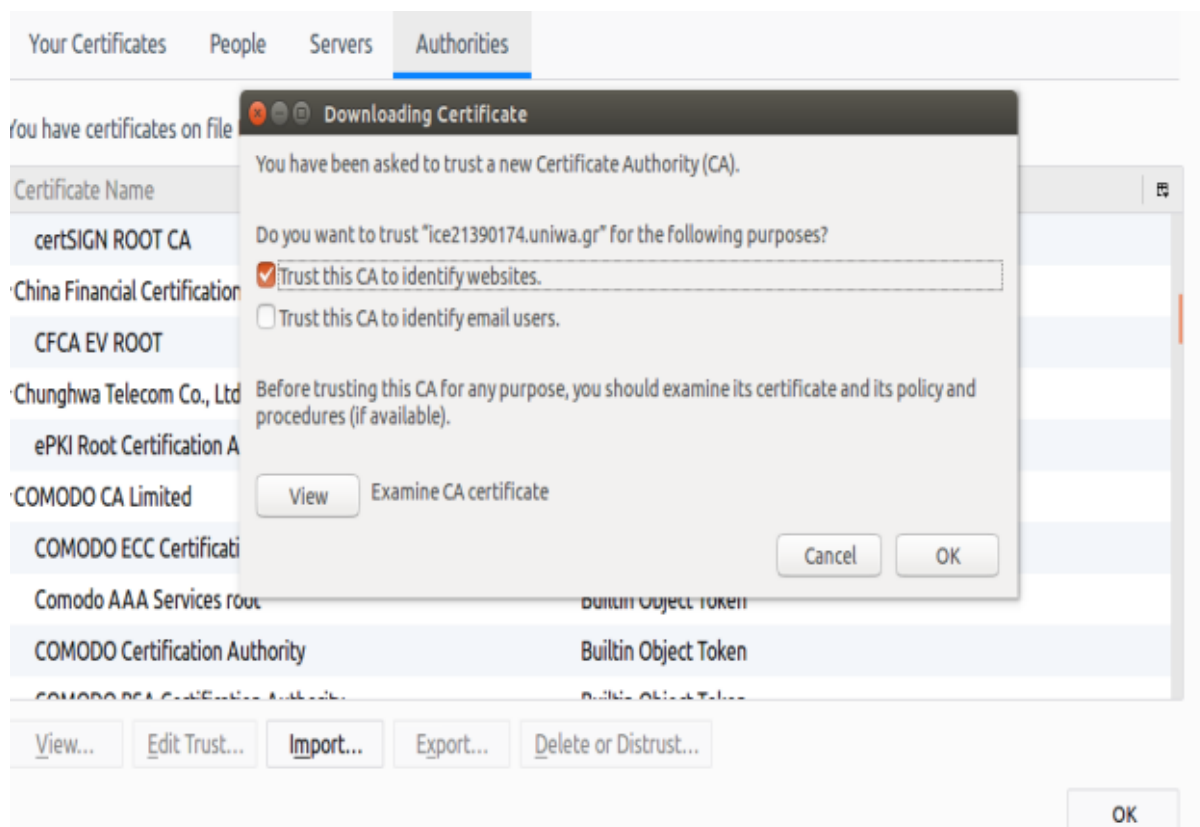
```
05/30/24]seed@VM:~/.../pki$ cp server.key server.pem
05/30/24]seed@VM:~/.../pki$ cat server.crt >> server.pem
05/30/24]seed@VM:~/.../pki$ openssl s_server -cert server.pem -www
Enter pass phrase for server.pem:
Using default temp DH parameters
ACCEPT
```

12.Είκονα Εκκίνηση του OpenSSL web server



13.Είκονα Μη αποδεκτό πιστοποιητικό στον Firefox

Βήμα 3: Αποδοχή του πιστοποιητικού της ΑΠ από το πρόγραμμα περιήγησης.



14.Είκονα Χειροκίνητη προσθήκη πιστοποιητικού στον Firefox

```
tailormade.com:4433/ x +
https://tailormade.com:4433

$ server -cert server.pem -www
Secure Renegotiation IS supported
Ciphers supported in s server binary
TLsv1/SSLv3: ECDHE-RSA-AES256-GCM-SHA384 TLsv1/SSLv3: ECDHE-ECDSA-AES256-GCM-SHA384
TLsv1/SSLv3: ECDHE-RSA-AES256-SHA384 TLsv1/SSLv3: ECDHE-ECDSA-AES256-SHA384
TLsv1/SSLv3: ECDHE-RSA-AES256-SHA TLsv1/SSLv3: ECDHE-ECDSA-AES256-SHA
TLsv1/SSLv3: SRP-DSS-AES-256-CBC-SHA TLsv1/SSLv3: SRP-RSA-AES-256-CBC-SHA
TLsv1/SSLv3: DHE-DSS-AES256-GCM-SHA384 TLsv1/SSLv3: DH-RSA-AES256-GCM-SHA384
TLsv1/SSLv3: DHE-RSA-AES256-GCM-SHA384 TLsv1/SSLv3: DHE-RSA-AES256-SHA256
TLsv1/SSLv3: DHE-DSS-AES256-SHA256 TLsv1/SSLv3: DH-RSA-AES256-SHA256
TLsv1/SSLv3: DH-DSS-AES256-SHA256 TLsv1/SSLv3: DHE-RSA-AES256-SHA
TLsv1/SSLv3: DHE-DSS-AES256-SHA TLsv1/SSLv3: DH-RSA-AES256-SHA
TLsv1/SSLv3: DHE-DSS-AES256-SHA TLsv1/SSLv3: DHE-RSA-CAMELLIA256-SHA
TLsv1/SSLv3: DHE-DSS-CAMELLIA256-SHA TLsv1/SSLv3: DH-RSA-CAMELLIA256-SHA
TLsv1/SSLv3: ECDH-ECDSA-AES256-GCM-SHA384 TLsv1/SSLv3: ECDH-RSA-AES256-GCM-SHA384
TLsv1/SSLv3: ECDH-ECDSA-AES256-SHA384 TLsv1/SSLv3: ECDH-RSA-AES256-SHA384
TLsv1/SSLv3: ECDH-ECDSA-AES256-SHA TLsv1/SSLv3: ECDH-RSA-AES256-SHA
TLsv1/SSLv3: AES256-SHA TLsv1/SSLv3: AES256-GCM-SHA384
TLsv1/SSLv3: CAMELLIA256-SHA TLsv1/SSLv3: PSK-AES256-CBC-SHA
TLsv1/SSLv3: ECDHE-RSA-AES128-GCM-SHA256 TLsv1/SSLv3: ECDHE-ECDSA-AES128-GCM-SHA256
TLsv1/SSLv3: ECDHE-RSA-AES128-SHA256 TLsv1/SSLv3: ECDHE-ECDSA-AES128-SHA256
TLsv1/SSLv3: ECDHE-RSA-AES128-SHA TLsv1/SSLv3: ECDHE-ECDSA-AES128-SHA
TLsv1/SSLv3: SRP-DSS-AES-128-CBC-SHA TLsv1/SSLv3: SRP-RSA-AES-128-CBC-SHA
TLsv1/SSLv3: SRP-AES-128-CBC-SHA TLsv1/SSLv3: DH-DSS-AES128-GCM-SHA256
TLsv1/SSLv3: DHE-DSS-AES128-GCM-SHA256 TLsv1/SSLv3: DH-RSA-AES128-GCM-SHA256
TLsv1/SSLv3: DHE-RSA-AES128-GCM-SHA256 TLsv1/SSLv3: DHE-RSA-AES128-SHA256
TLsv1/SSLv3: DHE-DSS-AES128-SHA256 TLsv1/SSLv3: DH-RSA-AES128-SHA256
TLsv1/SSLv3: DH-DSS-AES128-SHA256 TLsv1/SSLv3: DHE-RSA-AES128-SHA
TLsv1/SSLv3: DH-DSS-AES128-SHA TLsv1/SSLv3: DH-RSA-AES128-SHA
TLsv1/SSLv3: DH-DSS-AES128-SHA TLsv1/SSLv3: DHE-RSA-SEED-SHA
TLsv1/SSLv3: DHE-DSS-SEED-SHA TLsv1/SSLv3: DH-RSA-SEED-SHA
TLsv1/SSLv3: DH-DSS-SEED-SHA TLsv1/SSLv3: DHE-RSA-CAMELLIA128-SHA
TLsv1/SSLv3: DH-DSS-CAMELLIA128-SHA TLsv1/SSLv3: DH-RSA-CAMELLIA128-SHA
TLsv1/SSLv3: DH-DSS-CAMELLIA128-SHA TLsv1/SSLv3: ECDH-RSA-AES128-GCM-SHA256
TLsv1/SSLv3: ECDH-ECDSA-AES128-GCM-SHA256 TLsv1/SSLv3: ECDH-RSA-AES128-SHA256
TLsv1/SSLv3: ECDH-ECDSA-AES128-SHA256 TLsv1/SSLv3: ECDH-RSA-AES128-SHA
TLsv1/SSLv3: ECDH-ECDSA-AES128-SHA TLsv1/SSLv3: AES128-GCM-SHA256
TLsv1/SSLv3: AES128-SHA256 TLsv1/SSLv3: AES128-SHA
TLsv1/SSLv3: SEED-SHA TLsv1/SSLv3: CAMELLIA128-SHA
TLsv1/SSLv3: PSK-AES128-CBC-SHA TLsv1/SSLv3: ECDHE-RSA-RC4-SHA
TLsv1/SSLv3: ECDHE-ECDSA-RC4-SHA TLsv1/SSLv3: ECDH-RSA-RC4-SHA
TLsv1/SSLv3: ECDH-ECDSA-RC4-SHA TLsv1/SSLv3: RC4-SHA
TLsv1/SSLv3: RC4-MD5 TLsv1/SSLv3: PSK-RC4-SHA
TLsv1/SSLv3: ECDHE-RSA-DES-CBC3-SHA TLsv1/SSLv3: ECDHE-ECDSA-DES-CBC3-SHA
TLsv1/SSLv3: SRP-DSS-3DES-EDE-CBC-SHA TLsv1/SSLv3: SRP-RSA-3DES-EDE-CBC-SHA
TLsv1/SSLv3: SRP-3DES-EDE-CBC-SHA TLsv1/SSLv3: EDH-RSA-DES-CBC3-SHA
TLsv1/SSLv3: EDH-DSS-DES-CBC3-SHA TLsv1/SSLv3: DH-RSA-DES-CBC3-SHA
TLsv1/SSLv3: DH-DSS-DES-CBC3-SHA TLsv1/SSLv3: ECDH-RSA-DES-CBC3-SHA
TLsv1/SSLv3: ECDH-ECDSA-DES-CBC3-SHA TLsv1/SSLv3: DES-CBC3-SHA
TLsv1/SSLv3: PSK-3DES-EDE-CBC-SHA
...
Ciphers common between both SSL end points:
ECDHE-ECDSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-GCM-SHA256 ECDHE-ECDSA-AES256-GCM-SHA384
ECDHE-RSA-AES256-GCM-SHA384 ECDHE-RSA-AES128-SHA ECDHE-RSA-AES256-SHA
AES128-SHA AES256-SHA DES-CBC3-SHA
Signature Algorithms: ECDSA+SHA256:ECDSA+SHA384:ECDSA+SHA512:0x04+0x08:0x05+0x08:0x06+0x08:RSA+SHA256:RSA+SHA384:RSA+SHA512:ECDSA+SHA1:RSA+SHA1
Shared Signature Algorithms: ECDSA+SHA256:ECDSA+SHA384:ECDSA+SHA512:RSA+SHA256:RSA+SHA384:RSA+SHA512:ECDSA+SHA1:RSA+SHA1
Supported Elliptic Curves: 0x001D:P-256:P-384:P-521:0x0100:0x0101
Shared Elliptic curves: P-256:P-384:P-521
...
New, TLsv1/SSLv3, Cipher is ECDHE-RSA-AES128-GCM-SHA256
SSL-Session:
Protocol : TLSv1.2
Cipher : ECDHE-RSA-AES128-GCM-SHA256
Session-ID:
*****
```

15.Είкона Επίτυχια πιστοποίησης σελίδας.

Με αυτά τα βήματα, θα έχετε δημιουργήσαμε και δοκιμάσαμε έναν ασφαλή ιστότοπο χρησιμοποιώντας το πιστοποιητικό που εκδόσαμε για τον πελάτη μας.

Δραστηριότητα 4: Χρήση του πιστοποιητικού σε Apache HTTPS Web Server

Σε αυτή τη δραστηριότητα, θα φιλοξενήσετε τον ιστότοπο του πελάτη σας σε έναν πραγματικό web server που βασίζεται στο Apache. Ο Apache HTTP Server, ο οποίος είναι ήδη εγκατεστημένος στην εικονική μηχανή, υποστηρίζει το πρωτόκολλο HTTPS. Για να εξυπηρετήσει έναν ιστότοπο που θα χρησιμοποιεί HTTPS, πρέπει να διαμορφωθεί κατάλληλα ώστε να γνωρίζει πού να βρει το ιδιωτικό κλειδί και το πιστοποιητικό του ιστότοπου.

Βήμα 1: Μεταφορά των αρχείων του website στον Apache server

```
[05/30/24]seed@VM:~/.../pki$ sudo mkdir /var/www/tailormade  
[05/30/24]seed@VM:~/.../pki$
```

16.Εικόνα δημιουργήσαμε έναν κατάλογο όνομα tailormade.

```
[05/30/24]seed@VM:~/.../21390174Lab$ sudo touch index.html /var/www/tailormade/  
[05/30/24]seed@VM:~/.../21390174Lab$
```

17.Εικόνα δημιουργήσαμε το index html στο φακελό.

Βήμα 2: Ρύθμιση του Apache για HTTPS



18.Εικόνα ρυθμίση apache

```
SSLEngine On  
SSLCertificateFile /pki/server.crt  
SSLCertificateKeyFile /pki/server.key  
</VirtualHost>
```

19.Εικόνα ρυθμίση apache

```

[05/30/24]seed@VM:~/.../pki$ sudo apachectl configtest
AH00112: Warning: DocumentRoot [/var/www/seedlabclickjacking] does not exist
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 1
27.0.1.1. Set the 'ServerName' directive globally to suppress this message
Syntax OK
[05/30/24]seed@VM:~/.../pki$ sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
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 certi
ficates.
To activate the new configuration, you need to run:
    service apache2 restart
[05/30/24]seed@VM:~/.../pki$ sudo a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
    service apache2 reload
[05/30/24]seed@VM:~/.../pki$ sudo service apache2 restart
Enter passphrase for SSL/TLS keys for tailormade.com:443 (DSA): *****
[05/30/24]seed@VM:~/.../pki$

```

20.Εύκονα Ενεργοποίηση του SSL στον Apache

```

<!DOCTYPE html>
<html>
  <head>
    <title>Tailormade</title>
    <meta charset="UTF-8">
  </head>
  <body>
    <h1>TailorMade</h1>
    <h2>Owner</h2><hr>
    <p>Lastname: Papageorgiou</p>
    <p>firstname:filippos</p>
    <p>AM:21390174</p>
    <p>DATE:30/5/2024</p>
  </body>
</html>

```

21.Εύκονα σελίδα html



Εικόνα 21. Ενεργοποίηση του SSL στον Apache

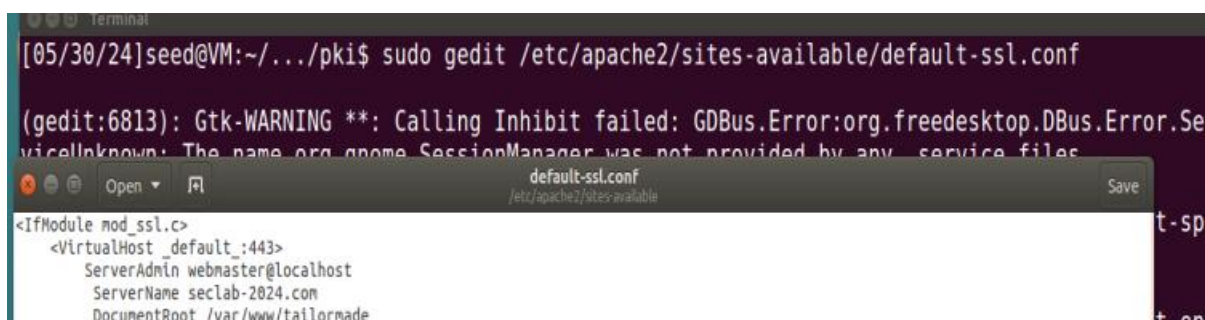
Δραστηριότητα 5: Επίθεση τύπου Man-In-The-Middle

Σε αυτή τη δραστηριότητα, θα επιδείξουμε πώς μια Υποδομή Δημοσίου Κλειδιού (PKI) μπορεί να αποτρέψει τις επιθέσεις τύπου Man-In-The-Middle (MITM). Θα προσομοιώσουμε μια επίθεση MITM για να κατανοήσουμε καλύτερα τις λειτουργίες και τις αδυναμίες της PKI.

Σενάριο της επίθεσης

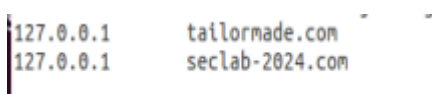
Υποθέτουμε ότι υπάρχει ένας νόμιμος ιστότοπος: `https://seclab-2024.com`. Ο επιτιθέμενος θέλει να παραπλανήσει τους χρήστες και να τους εκτρέψει σε έναν δικό του ιστότοπο `https://tailormade.gr`, ο οποίος υποτίθεται ότι προσομοιάζει οπτικά με τον αυθεντικό.

Βήμα 1: Ρύθμιση του κακόβουλου ιστότοπου.



Εικόνα 22. Ρύθμιση του κακόβουλου ιστότοπου

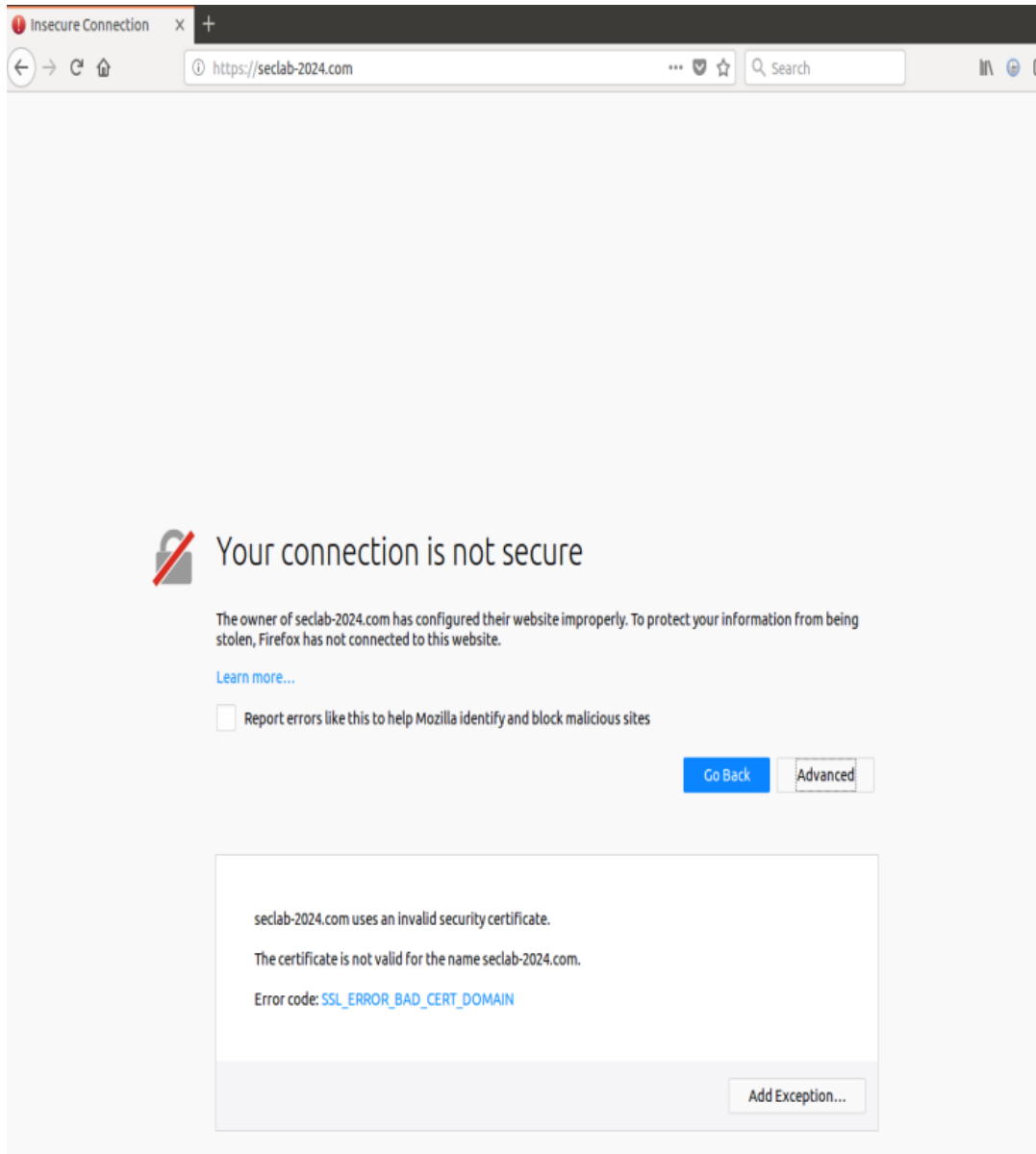
Βήμα 2: Ανακατεύθυνση του θύματος



Εικόνα 23. Προσομοίωση επίθεσης DNS

```
[05/30/24]seed@VM:~/.../pki$ sudo service apache2 restart
Enter passphrase for SSL/TLS keys for seclab-2024.com:443 (DSA): *****
[05/30/24]seed@VM:~/.../pki$
```

Εικόνα 24. Προσομοίωση επίθεσης DNS



Εικόνα 25. Απέριψη από browser

Παρατηρήσεις από την επίθεση τύπου Man-In-The-Middle

1. Προσπάθεια πρόσβασης στον κακόβουλο ιστότοπο:

- Κατά την προσπάθεια πρόσβασης στη διεύθυνση `https://seclab-2024.com`, ο browser πιθανότατα θα εμφανίσει ένα μήνυμα προειδοποίησης σχετικά με το πιστοποιητικό ασφαλείας.

2. Προειδοποιήσεις από τον browser:

- Ο browser ενδέχεται να εμφανίσει προειδοποιήσεις όπως "Your connection is not private" ή "This site's security certificate is not trusted".

- Αυτές οι προειδοποιήσεις συμβαίνουν επειδή το πιστοποιητικό που παρουσιάζεται από τον κακόβουλο ιστότοπο (`https://tailormade.com`) δεν είναι έγκυρο για το domain `https://seclab-2024.com`.

3. Αιτία αποτυχίας της επίθεσης:

- Η επίθεση αποτυγχάνει επειδή το πιστοποιητικό που χρησιμοποιείται από τον κακόβουλο ιστότοπο δεν ταιριάζει με το domain `seclab-2024.com`. Ο browser εντοπίζει αυτή τη δυσαρμονία και παρουσιάζει μια προειδοποίηση.

- Η PKI αποτρέπει τέτοιου είδους επιθέσεις επειδή το πιστοποιητικό που υπογράφεται από μια έγκυρη Αρχή Πιστοποίησης (CA) περιλαμβάνει το όνομα του domain για το οποίο έχει εκδοθεί. Εάν το domain δεν ταιριάζει, ο browser δεν θεωρεί το πιστοποιητικό έγκυρο.

4. Προστασία από την PKI:

- Η χρήση της PKI βοηθά στην αποτροπή επιθέσεων τύπου MITM, καθώς οι browsers είναι σχεδιασμένοι να ελέγχουν την εγκυρότητα των πιστοποιητικών και να προειδοποιούν τους χρήστες σε περίπτωση που το πιστοποιητικό δεν ταιριάζει με το domain του ιστότοπου.

- Οι χρήστες πρέπει να εμπιστεύονται μόνο τους ιστότοπους που παρουσιάζουν έγκυρα πιστοποιητικά από αξιόπιστες CA. Αυτό μειώνει σημαντικά την πιθανότητα επιτυχημένης επίθεσης MITM.

5. Συμπεράσματα:

- Οι επιθέσεις τύπου Man-In-The-Middle μπορούν να αποτραπούν αποτελεσματικά με τη χρήση της PKI και έγκυρων ψηφιακών πιστοποιητικών.

- Είναι σημαντικό οι χρήστες να δίνουν προσοχή στις προειδοποιήσεις ασφαλείας του browser τους και να μην αγνοούν τέτοια μηνύματα.

- Η αποτυχία της επίθεσης καταδεικνύει την αξία της PKI στην προστασία των ηλεκτρονικών επικοινωνιών και στην εξασφάλιση της ακεραιότητας και της εμπιστευτικότητας των

Δραστηριότητα 6: Επίθεση τύπου Man-In-The-Middle σε περίπτωση παραβιασμένης ΑΠ

/ Για αυτήν την δραστηριότητα χρησιμοποιήθηκαν πηγές από το διαδίκτυο για το πώς να κάνω fake κλειδιά . Και οι προηγούμενες δραστηριότητες, */*

Σε αυτή τη δραστηριότητα, θα εξετάσουμε τις συνέπειες μιας παραβίασης της Αρχής Πιστοποίησης (CA) και πώς αυτό μπορεί να επιτρέψει επιθέσεις τύπου Man-In-The-Middle (MITM). Θα υποθέσουμε ότι η πρωταρχική ΑΠ που κατασκευάσαμε στην Δραστηριότητα 1 έχει παραβιαστεί και το ιδιωτικό της κλειδί έχει κλαπεί από έναν εισβολέα.

Σχεδιασμός και Εκτέλεση Ενεργειών

1. Δημιουργία πλαστού πιστοποιητικού:

Ο εισβολέας θα δημιουργήσει ένα νέο πιστοποιητικό χρησιμοποιώντας το κλεμμένο ιδιωτικό κλειδί της ΑΠ.

```
/* openssl req -new -key ca.key -out fake_csr.csr -  
subj  
"/C=GR/ST=Attica/L=Athens/O=FakeCompany/OU=IT/CN=seclab-2024.com" */
```

```
[05/30/24]seed@VM:~/.../pki$ openssl req -new -key ca.key -out fake_csr.csr -subj "/C=GR/ST=Attica/L=Athens/O=FakeCompany/OU=IT/CN=seclab-2024.com"  
Enter pass phrase for ca.key:  
[05/30/24]seed@VM:~/.../pki$
```

Υπογραφή του πλαστού πιστοποιητικού με το κλεμμένο ιδιωτικό κλειδί της ΑΠ:

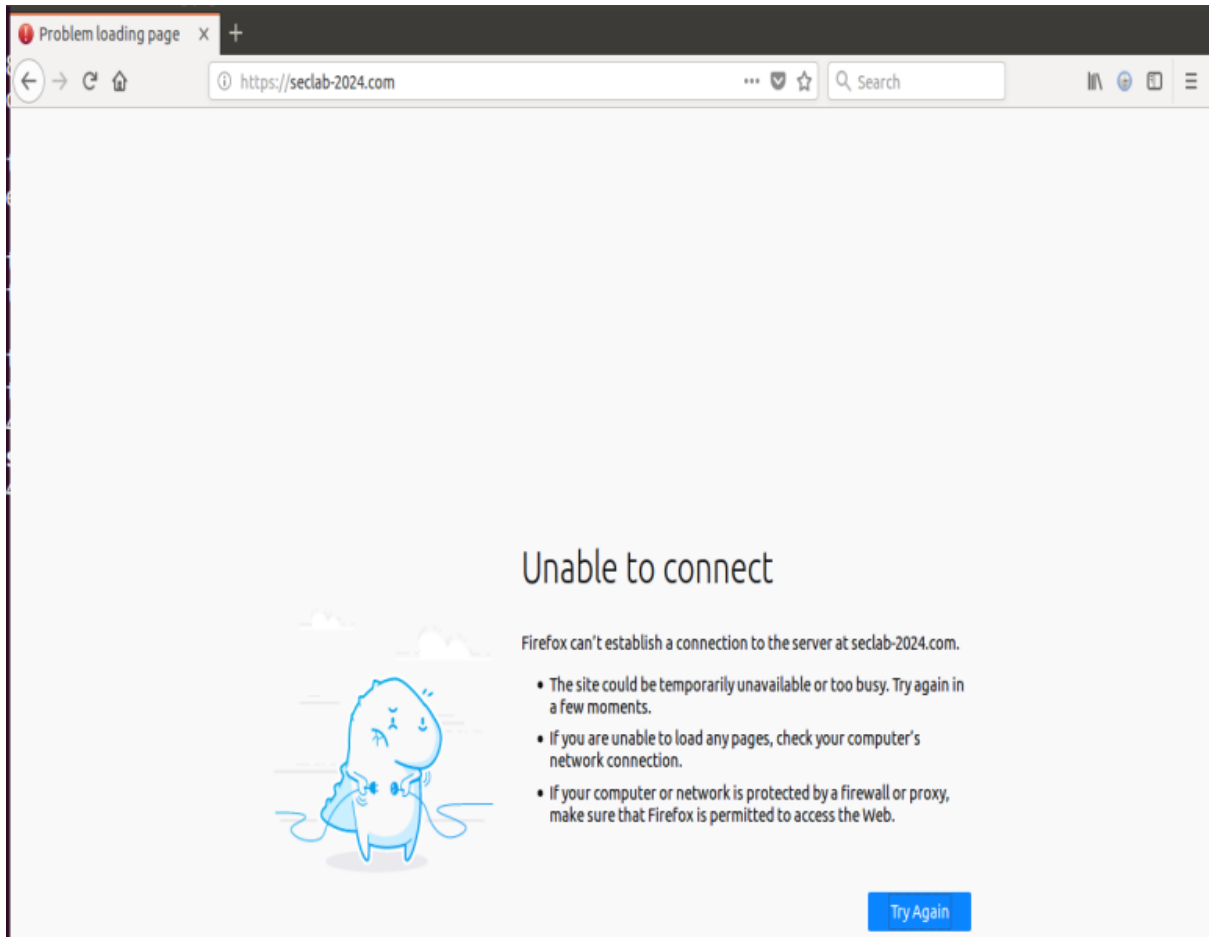
```
[05/30/24]seed@VM:~/.../pki$ openssl x509 -req -in fake_csr.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out fake_cert.crt -days 365 -sha256  
Signature ok  
subject=/C=GR/ST=Attica/L=Athens/O=FakeCompany/OU=IT/CN=seclab-2024.com  
Getting CA Private Key  
Enter pass phrase for ca.key:  
[05/30/24]seed@VM:~/.../pki$
```

2. Ρύθμιση του κακόβουλου ιστότοπου με το πλαστό πιστοποιητικό:

```
SSL Engine On  
SSLCertificateFile /home/seed/Desktop/pki/fake_cert.crt  
SSLCertificateKeyFile /home/seed/Desktop/pki/server.pem
```

```
[05/30/24]seed@VM:~/.../pki$ sudo service apache2 restart  
Enter passphrase for SSL/TLS keys for seclab-2024.com:443 (DSA): *****  
[05/30/24]seed@VM:~/.../pki$
```

3. Προσομοίωση της Επίθεσης MITM



Δύστηχος δεν μπόρεσα να λειτουργήσω το seclab-2024.com με ψευτικό πιστοποιητικό άλλα οι παρατηρήσεις βασή αυτήν την εργασία είναι οι παραλάτω.

Παρατηρήσεις και Ανάλυση

1. Συμπεράσματα:

- Επιτυχία της Επίθεσης: Εάν ο browser αποδεχτεί το πλαστό πιστοποιητικό χωρίς προειδοποιήσεις, αυτό σημαίνει ότι η επίθεση MITM είναι επιτυχής, υποδεικνύοντας την ευπάθεια που δημιουργείται όταν η ΑΠ παραβιαστεί.

- Σημασία της Ασφάλειας της ΑΠ: Αυτή η δραστηριότητα καταδεικνύει τη σημασία της ασφάλειας της ΑΠ. Η παραβίαση της ΑΠ μπορεί να έχει σοβαρές συνέπειες, καθώς επιτρέπει σε κακόβουλους παράγοντες να υποκλέπτουν και να παραποιούν την επικοινωνία.

2. Μέτρα Προστασίας:

- Ενίσχυση της Ασφάλειας της ΑΠ: Είναι κρίσιμο οι οργανισμοί να διασφαλίζουν την προστασία του ιδιωτικού κλειδιού της ΑΠ μέσω μέτρων όπως η χρήση Hardware Security Modules (HSM), ισχυρών κωδικών πρόσβασης και τακτικών ελέγχων ασφαλείας.

- Ανάκληση Πιστοποιητικών: Σε περίπτωση παραβίασης, τα παραβιασμένα πιστοποιητικά πρέπει να ανακληθούν αμέσως και να ενημερωθούν τα συστήματα ελέγχου πιστοποιητικών (CRL, OCSP) για να αποτρέψουν τη χρήση τους.