

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
Τμήμα Πληροφορικής



Εργασία Μαθήματος **Ασφάλεια Πληροφοριακών Συστημάτων**

Άσκηση <<αριθμός άσκησης>>	Τελική εργασία μαθήματος 2021-22
Όνομα φοιτητή – Αρ. Μητρώου (όλων σε περίπτωση ομαδικής εργασίας)	Αθανασία Κομματίδου – Π18078
	Αναστασία Ιωάννα Μέξα – Π18101
	Βασιλική Πασιά – Π18123
Ημερομηνία παράδοσης	15/02/2022



Εκφώνηση της άσκησης

Να χρησιμοποιήσετε όποια γλώσσα προγραμματισμού επιθυμείτε (ενδεικτικά php, java, .net, python) και να αναπτύξετε μία διαδικτυακή εφαρμογή της επιλογής σας και να περιγράψετε τις βασικές επιχειρησιακές λειτουργίες της εφαρμογής αυτής. Ενδεικτικό παράδειγμα δίδεται στην 1η Άσκηση του μαθήματος. Μπορείτε να χρησιμοποιήσετε/τροποποιήσετε κάποια διαδικτυακή εφαρμογή που έχετε αναπτύξει στο πλαίσιο άλλου μαθήματος. Όλες οι απαιτούμενες τεχνολογίες (web, application, database server) θα είναι επίσης ελεύθερης επιλογής. Η εφαρμογή θα χρησιμοποιηθεί ως το περιβάλλον - υποδομή, ώστε να υλοποιήσετε τις παρακάτω υπηρεσίες ασφάλειας.

1. Μελέτη ασφάλειας ΠΣ. Σε αυτό το στάδιο θα πραγματοποιήσετε μία σύντομη μελέτη ασφάλειας ΠΣ, η οποία θα περιλαμβάνει: (α) Ανάλυση επικινδυνότητας και (β) Σχέδιο Πολιτικής Ασφάλειας. (Σημείωση: αυτό το βήμα αποτελεί επέκταση της 1ης άσκησης)
2. Να υλοποιήσετε κρυπτογράφηση ssl στον server. Η υπηρεσία σας να λειτουργεί μόνο σε https με τη χρήση πιστοποιητικού στον web server. (Σημείωση: αυτό το βήμα βασίζεται στην 3η άσκηση)
3. Να υλοποιείστε ένα μηχανισμό αυθεντικοποίησης (user authentication), πχ. username,password, one-time password, certificate based κτλ. και ελέγχου πρόσβασης (authorization), π.χ. LDAP-based, identity management, certificate based, group-based, role-based κτλ. (Σημείωση: αυτό το βήμα θα αποτελεί επέκταση της 4ης άσκησης)
4. Για τη λήψη δεδομένων εισόδου από τους χρήστες της εφαρμογής, να χρησιμοποιήσετε προγραμματιστικές συναρτήσεις, ανάλογα με τη γλώσσα προγραμματισμού που έχετε επιλέξει, οι οποίες να επιβάλουν input filtering και validation (δείτε ενδεικτικά: https://cheatsheetseries.owasp.org/cheatsheets/Input_Validation_Cheat_Sheet.html), ώστε να μην είναι εύκολο για έναν κακόβουλο χρήστη να υλοποιήσει επιθέσεις τύπου sql injection και XSS.
5. Να πραγματοποιήσετε αυτοματοποιημένο έλεγχο για την εύρεση ευπαθειών ασφάλειας (Σημείωση: αυτό το βήμα αποτελεί θα επέκταση της 6ης άσκησης)



ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

Τελική εργασία μαθήματος 2021-22	1
1. Μελέτη ασφάλειας ΠΣ.	4
1.1. Ανάλυση επικινδυνότητας	4
1.2. Σχέδιο Πολιτικής Ασφάλειας.....	6
2. Κρυπτογράφηση SSL.....	6
2.1. Προσθήκη της εφαρμογής στον IIS	6
2.2. Δημιουργία SSL certificate	8
3. Μηχανισμοί αυθεντικοποίησης και ελέγχου πρόσβασης	10
3.1. Μηχανισμός αυθεντικοποίησης (user authentication)	10
3.2. Κώδικας μηχανισμού αυθεντικοποίησης	16
3.3. Ελέγχου πρόσβασης (authorization).....	16
3.4. Κώδικας ελέγχου πρόσβασης	17
4. Input filtering και validation.....	18
4.1. Regular expressions.....	18
4.2. Email Address Validation.....	19
4.3. Date Validation.....	19
4.4. SQL και XSS injection	21
5. Εύρεση ευπαθειών ασφάλειας	22
5.1. Absence of Anti-CSRF Tokens	22
5.2. Cookie Without Secure Flag	23
5.3. Incomplete or No Cache-control Header Set	23
5.4. Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s).....	23
5.5. X-AspNet-Version Response Header	23
5.6. X-Content-Type-Options Header Missing	24
5.7. Timestamp Disclosure – Unix	24
5.8. Clickjacking X Frame Options Header Missing	25
6. Βιβλιογραφία.....	26



1. Μελέτη ασφάλειας ΠΣ.

Στα πλαίσια αυτής της εργασίας αξιοποιήσαμε μια web εφαρμογή για κλείσιμο θέσης σε parking, την οποία είχαμε αναπτύξει σε προηγούμενο εξάμηνο. Έχει υλοποιηθεί στο περιβάλλον του Visual Studio σε γλώσσα C# χρησιμοποιώντας τεχνολογία ASP.NET. Η βάση δεδομένων που χρησιμοποιήθηκε είναι η PostgreSQL.

Η εφαρμογή μας παρέχει τις παρακάτω on-line υπηρεσίες:

- **Κλείσιμο θέσης parking:** Παρέχεται η δυνατότητα σε όλους τους χρήστες της εφαρμογής, να περιηγηθούν στις σελίδες του ιστοτόπου και να ενημερωθούν για τα συνεργαζόμενα Parking (περιοχή, φόρμα επικοινωνίας, ωράριο λειτουργίας, διαθεσιμότητα θέσεων). Συγκεκριμένα για τους εγγεγραμμένους χρήστες, μπορούν να επιλέξουν το parking της αρεσκείας τους, να διαλέξουν μία από τις διαθέσιμες θέσεις και να πραγματοποιήσουν την κράτηση.
- **Εγγραφή/Διαχείριση χρηστών:** Οι χρήστες εγγράφονται για τις υπηρεσίες της εφαρμογής μέσω web φόρμας, παρέχοντας στοιχεία όπως όνομα, επίθετο, username, password, email, αριθμό πινακίδας/ων κτλ. Για την αυθεντικοποίηση του χρήστη, στέλνεται αυτόματα ένα verification email με έναν εξαψήφιο κωδικό, τον οποίο έπειτα πρέπει να πληκτρολογήσει σε συγκεκριμένο πεδίο. Στην συνέχεια, ο χρήστης θα μπορεί να κάνει είσοδο στον λογαριασμό του, να προσθέσει οχήματα (μηχανές/αυτοκίνητα) και να δει το ιστορικό των κρατήσεών του. Ακόμα, υπάρχουν και οι διαχειριστές της εφαρμογής οι οποίοι μπορούν να διαγράψουν κάποιον χρήστη (μαζί με τις κρατήσεις του) ή/και να διαγράψουν κάποια συγκεκριμένη κράτηση.
- **Ηλεκτρονική πληρωμή:** Οι εγγεγραμμένοι χρήστες, μπορούν να πληρώσουν ηλεκτρονικά για να εξασφαλίσουν την ολοκλήρωση της κράτησής τους. Τα στοιχεία της χρεωστικής/πιστωτικής κάρτας τους, αποθηκεύονται στη βάση σε κρυπτογραφημένη μορφή.

1.1. Ανάλυση επικινδυνότητας

Με τον όρο ανάλυση επικινδυνότητας αναφερόμαστε στη διαδικασία εντοπισμού, αξιολόγησης και ιεράρχηση των κινδύνων των ευάλωτων σημείων ενός Π.Σ. και προσδιορισμού των επιμέρους κινδύνων οι οποίοι σε περίπτωση εκδήλωσης θα είχαν αρνητικές συνέπειες για τον υπό μελέτη οργανισμό.

Για να πραγματοποιηθεί η ανάλυση επικινδυνότητας, είναι σημαντική η απογραφή των απειλών, των κινδύνων και των συνεπειών, όπως φαίνεται στον παρακάτω πίνακα.



Risk Assessment			
Απειλές	Κίνδυνοι	Βαθμός κινδύνου	Συνέπειες
Μη εξουσιοδοτημένη πρόσβαση στο σύστημα (Unauthorized Access)	Αν οι μη αυθεντικοποιημένοι χρήστες αποκτήσουν πρόσβαση με δικαιώματα εγγεγραμμένων χρηστών ή διαχειριστών, μπορούν να έχουν πρόσβαση στα προσωπικά δεδομένα άλλων χρηστών και να πραγματοποιήσουν αλλοιώσεις στα δεδομένα του ΠΣ. (πχ. δημιουργία εικονικών κρατήσεων, διαγραφή χρηστών και κρατήσεων)	Υψηλός	Αναξιοπιστία, Δυσφήμιση, Άμεσες οικονομικές απώλειες, Νομικές κυρώσεις.
Μη εξουσιοδοτημένη εκτέλεση κώδικα (Code Injection)	Η εκτέλεση κώδικα τρίτου με συνέπειες τόσο για τον χρήστη όσο και για το ΠΣ. Σε περίπτωση sql injection μπορεί να επέλθει αλλοίωση της βάσης δεδομένων, ενώ σε περίπτωση XSS οδηγεί τον χρήστη σε ανεπιθύμητες ενέργειες και επιπτώσεις.	Υψηλός	Αναξιοπιστία, Δυσφήμιση, Άμεσες οικονομικές απώλειες, Νομικές κυρώσεις, Παρεμπόδιση λειτουργιών.
Άρνηση υπηρεσίας (Denial of Service)	Μη ανταπόκριση του web server στα αιτήματα των χρηστών.	Μέτριος	Δυσφήμιση, Άμεσες οικονομικές απώλειες, Παρεμπόδιση λειτουργιών.
Υποκλοπή δεδομένων από την βάση (Data security breach)	Η παραβίαση της ασφάλειας της βάσης δεδομένων, αποκαλύπτει εμπιστευτικά δεδομένα.	Υψηλός	Αναξιοπιστία, Δυσφήμιση, Νομικές κυρώσεις, Παρεμπόδιση λειτουργιών.



1.2. Σχέδιο Πολιτικής Ασφάλειας

Το σχέδιο πολιτικής ασφάλειας περιγράφει τους στόχους της ασφάλειας και τις αντίστοιχες διαδικασίες που πρέπει να ακολουθούνται ώστε να επιτευχθούν αυτοί οι στόχοι. Οι βασικές αρχές ασφάλειας είναι η διαθεσιμότητα των δεδομένων, η εμπιστευτικότητα και η ακεραιότητα. Αναλυτικότερα:

- Διαθεσιμότητα (Availability): Διασφάλιση της προσπελασιμότητας της πληροφορίας σε εξουσιοδοτημένους χρήστες όποτε απαιτείται
- Εμπιστευτικότητα (Confidentiality): Διασφάλιση της προσπελασιμότητας της πληροφορίας μόνον από όσους έχουν τα απαραίτητα δικαιώματα
- Ακεραιότητα (Integrity): Διαφύλαξη της ακρίβειας και της πληρότητας της πληροφορίας και των μεθόδων επεξεργασίας αυτής.

Λαμβάνοντας υπόψη την ανάλυση επικινδυνότητας που αναλύθηκε προηγουμένως, υλοποιήθηκαν τα ακόλουθα μέτρα ασφάλειας στην προσπάθεια αντιμετώπισης των αντίστοιχων κινδύνων.

Αρχικά, προκειμένου ένας χρήστης να χρησιμοποιήσει τις υπηρεσίες που του παρέχει η διαδικτυακή εφαρμογή επιβάλλεται να δημιουργήσει προσωπικό λογαριασμό ή να συνδεθεί σε έναν ήδη υπάρχων. Συνεπώς, προκύπτει η άμεση ανάγκη ανάπτυξης ενός μηχανισμού αυθεντικοποίησης και ελέγχου πρόσβασης.

Αντίστοιχα, ζωτικής σημασίας είναι η εξασφάλιση μιας ασφαλούς σύνδεσης μεταξύ δύο συσκευών που ανταλλάζουν δεδομένα μέσω του διαδικτύου. Για το λόγο αυτό χρησιμοποιείται το πρωτόκολλο SSL, το οποίο προστατεύει τα δεδομένα των χρηστών με ένα επίπεδο ισχυρής κρυπτογράφησης και δημιουργεί ένα ασφαλές περιβάλλον τόσο για τους επισκέπτες, όσο και για τους ιδιοκτήτες της εφαρμογής.

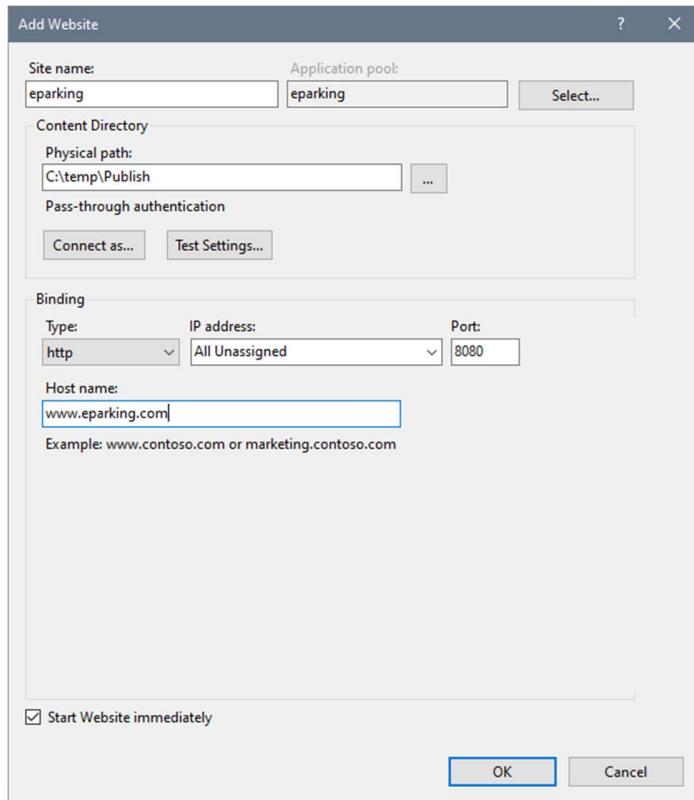
Επιπροσθέτως, για την διασφάλιση της ομαλής λειτουργίας της εφαρμογής είναι απαραίτητος ο έλεγχος και το φιλτράρισμα του input του χρήστη. Ειδάλλως, η λανθασμένη (εσκεμμένα ή μη) καταχώρηση στοιχείων μπορεί να οδηγήσει σε injection attacks, memory leakage και compromised systems.

Τέλος, θα πρέπει να γίνεται συχνός έλεγχος για την εύρεση ευπαθειών ασφάλειας, πιθανώς μέσω αυτοματοποιημένων συστημάτων.

2. Κρυπτογράφηση SSL

2.1. Προσθήκη της εφαρμογής στον IIS

Καταρχάς, για να εγκαταστήσουμε την εφαρμογή μας στον IIS την κάναμε Publish μέσα από το περιβάλλον του Visual Studio. Στη συνέχεια, κάναμε προσθήκη της ιστοσελίδας μας μέσω του IIS Manager με τις παρακάτω ρυθμίσεις.



Με σκοπό να πληκτρολογούμε για την είσοδο στην εφαρμογή το domain name που ορίσαμε αντί για «localhost», τροποποιήσαμε το αρχείο hosts που βρίσκεται στο path C:\Windows\System32\drivers\etc\, όπως φαίνεται στο παρακάτω screenshot.

```
# Copyright (c) 1993-2009 Microsoft Corp.  
#  
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.  
#  
# This file contains the mappings of IP addresses to host names. Each  
# entry should be kept on an individual line. The IP address should  
# be placed in the first column followed by the corresponding host name.  
# The IP address and the host name should be separated by at least one  
# space.  
#  
# Additionally, comments (such as these) may be inserted on individual  
# lines or following the machine name denoted by a '#' symbol.  
#  
# For example:  
#  
#      102.54.94.97      rhino.acme.com          # source server  
#      38.25.63.10      x.acme.com            # x client host  
  
# localhost name resolution is handled within DNS itself.  
#      127.0.0.1        localhost  
#      ::1              localhost  
  
127.0.0.1    www.eparking.com
```



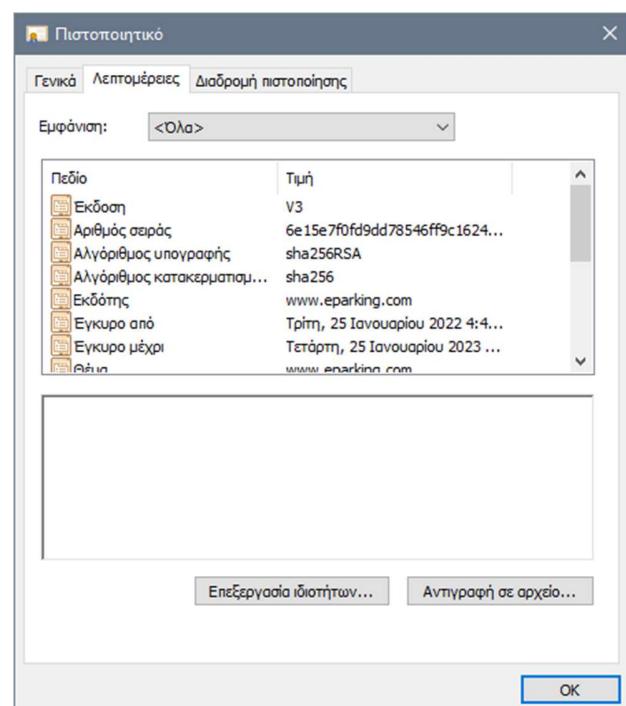
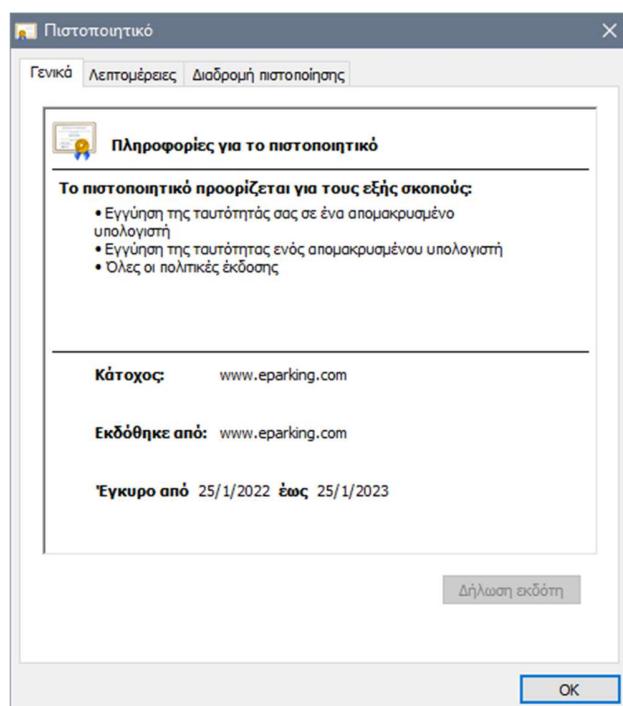
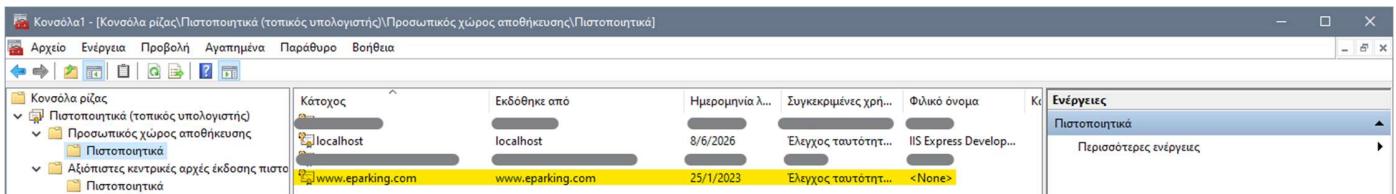
2.2. Δημιουργία SSL certificate

Για την δημιουργία ενός SSL πιστοποιητικού τρέξαμε την παρακάτω εντολή στο PowerShell με δικαιώματα διαχειριστή.

```
Administrator: Windows PowerShell (x86)
PS C:\WINDOWS\system32> New-SelfSignedCertificate -DnsName "www.eparking.com" -CertStoreLocation "cert:\LocalMachine\My"
PSParentPath: Microsoft.PowerShell.Security\Certificate::LocalMachine\My
Thumbprint Subject
----- -----
538A19B57D7D0AD3269F2055099F70E2A5D889FD CN=www.eparking.com

PS C:\WINDOWS\system32>
```

Το αποτέλεσμα της εντολής ήταν η δημιουργία ενός αυτό-υπογεγραμμένου πιστοποιητικού, το οποίο μπορούμε να το εντοπίσουμε μέσω της εφαρμογής *mmc.exe*. Πιο συγκεκριμένα βρίσκεται στον υπο-φάκελο «Πιστοποιητικά», του φακέλου «Προσωπικός χώρος αποθήκευσης».





Παρόλα αυτά, το πιστοποιητικό μας δεν θεωρείται έγκυρο αν δεν εισαχθεί στην λίστα «Αξιόπιστες κεντρικές αρχές έκδοσης πιστοποιητικών» στο mmc.exe. Αφού εισαχθεί, θα θεωρείται πλέον έγκυρο από τον browser.

Κονσόλα1 - [Κονσόλα ρίζας\Πιστοποιητικά (τοπικός υπολογιστής)\Αξιόπιστες κεντρικές αρχές έκδοσης πιστοποιητικών\Πιστοποιητικά]

Αρχείο Ενέργεια Προβολή Αγοραμένα Παράθυρο Βοήθεια

Κάτοχος Εκδόθηκε από Ημερομηνία λ... Συγκεκριμένες χρή... Φύλικό όνομα

Ενέργειες Πιστοποιητικά Περισσότερες ενέργειες

Κάτοχος	Εκδόθηκε από	Ημερομηνία λ...	Συγκεκριμένες χρή...	Φύλικό όνομα
Microsoft Authenticode(tm) Ro...	Microsoft Authenticode(tm) Root...	1/1/2000	Αισιοδότης ηλεκτρον...	Microsoft Authentico...
Microsoft ECC Product Root Ce...	Microsoft ECC Product Root Certi...	27/2/2043	<Όλα>	Microsoft ECC Prod...
Microsoft ECC TS Root Certifica...	Microsoft ECC TS Root Certificate ...	27/2/2043	<Όλα>	Microsoft ECC TS R...
Microsoft Root Authority	Microsoft Root Authority	31/12/2020	<Όλα>	Microsoft Root Aut...
Microsoft Root Certificate Auth...	Microsoft Root Certificate Authori...	10/5/2021	<Όλα>	Microsoft Root Cert...
Microsoft Root Certificate Auth...	Microsoft Root Certificate Authori...	24/6/2035	<Όλα>	Microsoft Root Cert...
Microsoft Root Certificate Auth...	Microsoft Root Certificate Authori...	23/3/2036	<Όλα>	Microsoft Root Cert...
Microsoft Time Stamp Root Cer...	Microsoft Time Stamp Root Certif...	23/10/2039	<Όλα>	Microsoft Time Sta...
NO LIABILITY ACCEPTED, (c97 Ve...		8/1/2004	Χρονική σήμανση	VeriSign Time Stam...
QuoVadis Root CA 2	QuoVadis Root CA 2	24/11/2031	Ελεγχός ταυτότη...	QuoVadis Root CA 2
QuoVadis Root CA 2 G3	QuoVadis Root CA 2 G3	12/1/2042	Ελεγχός ταυτότη...	QuoVadis Root CA ...
QuoVadis Root Certification Au...	QuoVadis Root Certification Auth...	17/3/2021	Ελεγχός ταυτότη...	QuoVadis Root Cert...
Razer Chroma SDK	Razer Chroma SDK	11/5/2028	Ελεγχός ταυτότη...	<None>
SecureTrust CA	SecureTrust CA	31/12/2029	Ελεγχός ταυτότη...	Trustwave
Security Communication Root...	Security Communication RootCA1	30/9/2023	Ελεγχός ταυτότη...	SECOM Trust Syste...
Starfield Class 2 Certification A...	Starfield Class 2 Certification Auth...	29/6/2034	Ελεγχός ταυτότη...	Starfield Class 2 Cer...
Starfield Root Certificate Autho...	Starfield Root Certificate Authorit...	1/1/2038	Ελεγχός ταυτότη...	Starfield Root Certif...
Starfield Services Root Certificat...	Starfield Services Root Certificate ...	1/1/2030	Ελεγχός ταυτότη...	Starfield Technologi...
StartCom Certification Authority	StartCom Certification Authority	17/9/2036	Ελεγχός ταυτότη...	StartCom Certificati...
Symantec Enterprise Mobile Root ...	Symantec Enterprise Mobile Root ...	15/3/2032	Υπογραφή κύβικη	<None>
thawte Primary Root CA	thawte Primary Root CA	17/7/2036	Ελεγχός ταυτότη...	thawte
thawte Primary Root CA - G3	thawte Primary Root CA - G3	2/12/2037	Ελεγχός ταυτότη...	thawte Primary Roo...
Thawte Timestamping CA	Thawte Timestamping CA	1/1/2021	Χρονική σήμανση	Thawte Timestampin...
T-TeleSec GlobalRoot Class 2	T-TeleSec GlobalRoot Class 2	2/10/2033	Ελεγχός ταυτότη...	T-TeleSec GlobalRo...
TWCA Root Certification Authority	TWCA Root Certification Authority	31/12/2030	Ελεγχός ταυτότη...	TWCA Root Certific...
USERTrust ECC Certification Aut...	USERTrust ECC Certification Auth...	19/1/2038	Ελεγχός ταυτότη...	Sectigo ECC
USERTrust RSA Certification Aut...	USERTrust RSA Certification Auth...	19/1/2038	Ελεγχός ταυτότη...	Sectigo
UTN-USERFirst-Object	UTN-USERFirst-Object	9/7/2019	Σύστημα οργών κ...	Sectigo (UTN Object)
VeriSign Class 3 Public Primary ...	VeriSign Class 3 Public Primary Ce...	17/7/2036	Ελεγχός ταυτότη...	VeriSign
VeriSign Universal Root Certific...	VeriSign Universal Root Certificati...	2/12/2037	Ελεγχός ταυτότη...	VeriSign Universal R...
www.eparking.com	www.eparking.com	25/1/2023	Ελεγχός ταυτότη...	<None>

Ο χώρος αποθήκευσης Αξιόπιστες κεντρικές αρχές έκδοσης πιστοποιητικών περιέχει 67 πιστοποιητικά.

Το τελικό βήμα είναι να προσθέσουμε ένα https binding στην ιστοσελίδα μας μέσω του IIS.

Edit Site Binding

Type: https IP address: All Unassigned Port: 443

Host name:

Require Server Name Indication

Disable TLS 1.3 over TCP Disable QUIC

Disable Legacy TLS Disable HTTP/2

Disable OCSP Stapling

SSL certificate:

www.eparking.com

OK Cancel

Site Bindings

Type	Host Name	Port	IP Address	Binding Information
http	www.eparking.c...	8080	*	
https	www.eparking.com	443	*	

Add... Edit... Remove Browse Close



Το αποτέλεσμα όλων των παραπάνω είναι όταν πληκτρολογούμε στον browser την διεύθυνση www.eparking.com, να συνδεόμαστε στην ιστοσελίδα μας με ασφάλεια, όπως φαίνεται παρακάτω.

The screenshot shows a web browser window with the URL <http://www.eparking.com/HomePage.aspx>. The page features a large image of a multi-level parking garage with several cars parked. Overlaid on the image is the text "MAKE YOUR RESERVATION" in large, bold, white letters. Below this, smaller text reads "Welcome to the best online parking reservation platform. Book your place now." A prominent "BOOK NOW" button is centered at the bottom of the main image area. At the top of the page, there is a navigation bar with links for "Book Now", "Parkings", "Contact Us", "Log in", and "Sign up". The browser's address bar and various tabs are visible at the top.

3. Μηχανισμοί αυθεντικοποίησης και ελέγχου πρόσβασης

3.1. Μηχανισμός αυθεντικοποίησης (user authentication)

Έχουμε υλοποιήσει τους μηχανισμούς username, password και one-time password (μέσω email) για την αυθεντικοποίηση του χρήστη. Να σημειωθεί ότι το password του χρήστη στην βάση κρυπτογραφημένο με την χρήση salt και hash.

Data Output									Explain	Messages	Notifications
	first_name	last_name	email	username	hashed_password	salt	admin				
1	Anastasia	Mexa	ranger.kataang@gmail.com	stacy	z/Aep9a05OJ7RAy08AxKLu...	BZlbOGtXqcQSRxDYhZouUc...	true				
2	Nancy	Kommatidou	nancygianna11@gmail.com	nancy11	DyZvpCOMufLxInsANtD/kja...	NPWJtKxthOTlskgtpggokUD...	false				



Ο χρήστης έχει την επιλογή να κάνει εγγραφή (Sign Up) ή να συνδεθεί με τα στοιχεία του (Log In), πατώντας τα αντίστοιχα κουμπιά του μενού.

The screenshot shows the homepage of the eParking website. At the top, there is a navigation bar with links for "Book Now", "Parkings", "Contact Us", "Log in" (which is underlined in red), and "Sign up" (which is also underlined in red). The main visual is a photograph of a modern parking garage with red and white striped bollards and several cars parked. Overlaid on the image is a large, bold text "MAKE YOUR RESERVATION". Below this, smaller text reads "Welcome to the best online parking reservation platform. Book your place now." A prominent "BOOK NOW" button is centered at the bottom of the main image area.

Ξεκινώντας με την διαδικασία εγγραφής, ο χρήστης μεταφέρεται στην σελίδα Sign Up όπου καλείται να συμπληρώσει τα στοιχεία του. Αρχικά συμπληρώνει το όνομα, το επώνυμο, το email του, το username, τον κωδικό του δύο φορές για λόγους επαλήθευσης και τέλος πρέπει να συμφωνήσει με τους όρους και τις προϋποθέσεις της εφαρμογής. Στην συνέχεια πατάει το κουμπί «Submit» για να προχωρήσει την διαδικασία εγγραφής.

The screenshot shows the "SIGN UP" registration form overlaid on a photograph of a parking garage. The form contains fields for "First Name" (with placeholder "Enter your First Name"), "Last Name" (with placeholder "Enter your Last Name"), "Email" (with placeholder "Enter your Email"), "Username" (with placeholder "Enter your Username"), "Password" (with placeholder "Enter your Password"), and "Repeat Password" (with placeholder "Repeat Password"). There is also a checkbox labeled "I accept the Terms of Service" and a "SUBMIT" button at the bottom. The background image shows a modern parking garage with red and white striped bollards and a sign indicating "2G" and "3F".



Μόλις πατήσει το κουμπί, θα του ζητηθεί από την εφαρμογή να εισάγει το verification code που έχει λάβει ο χρήστης στο email του.

eparking confirmation email Εισερχόμενα X



eparking.papei2021@gmail.com
προς εγώ ▾

3:36 μ.μ. (πριν από 0 λεπτά)



Αγγλικά ▾ > Ελληνικά ▾ Μετάφραση μηνύματος

Απενεργοποίηση για: Αγγλικά X

Hi nancy,

Here is your 5-digit code to verify your email address: 92570

Confirmed!

Yes, I confirm.

I haven't received it yet.

◀ Απάντηση

▶ Προώθηση



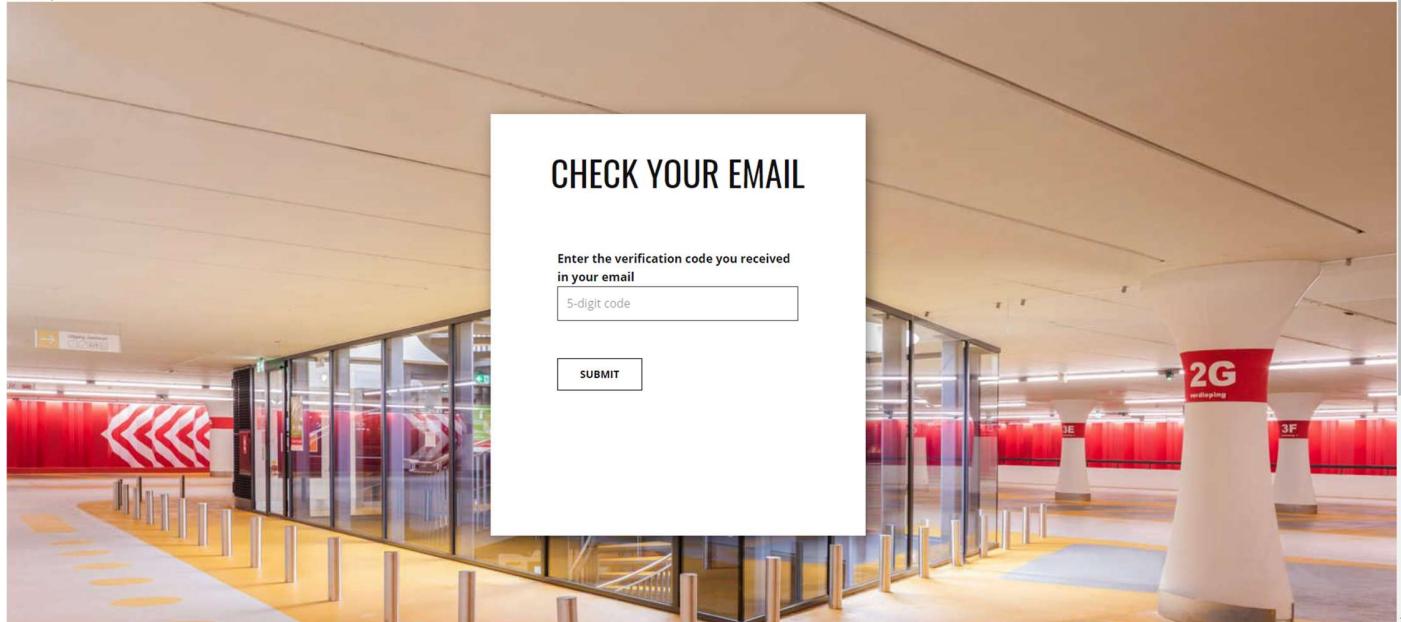
Book Now

Parkings

Contact Us

Log in

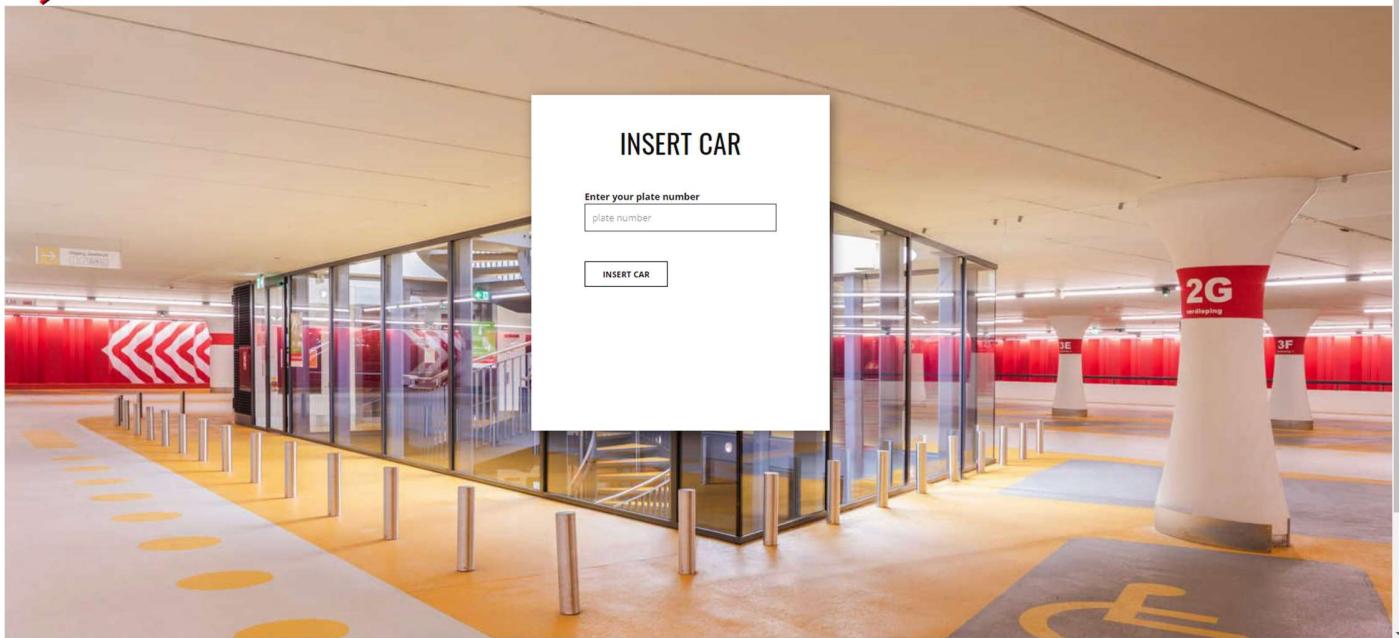
Sign up



Στην συνέχεια, αφού έχει εισάγει σωστό κωδικό, ζητείται από τον χρήστη να εισάγει τον αριθμό πινακίδας/ων από τα οχήματα που θέλει να έχει πρόσβαση η εφαρμογή. Μπορεί να εισάγει και πινακίδες αυτοκινήτων και πινακίδες μηχανών.



Book Now Parkings Contact Us My Profile Log Out



Όταν ολοκληρώσει την εγγραφή του στην εφαρμογή, αυτόματα μεταφέρεται στην σελίδα του My Profile και εκεί φαίνονται σε πίνακες των οχημάτων του και οι κρατήσεις που έχει πραγματοποιήσει. Εννοείται ότι, αν ο χρήστης μόλις εγγράφτηκε δεν θα υπάρχει καμία κράτηση παρά μόνο τα οχήματά του.



Book Now Parkings Contact Us My Profile Log Out Hi nancy



My Vehicles	
Type	Plate Number
car	IOY2341
motorcycle	IOY234

ADD MORE

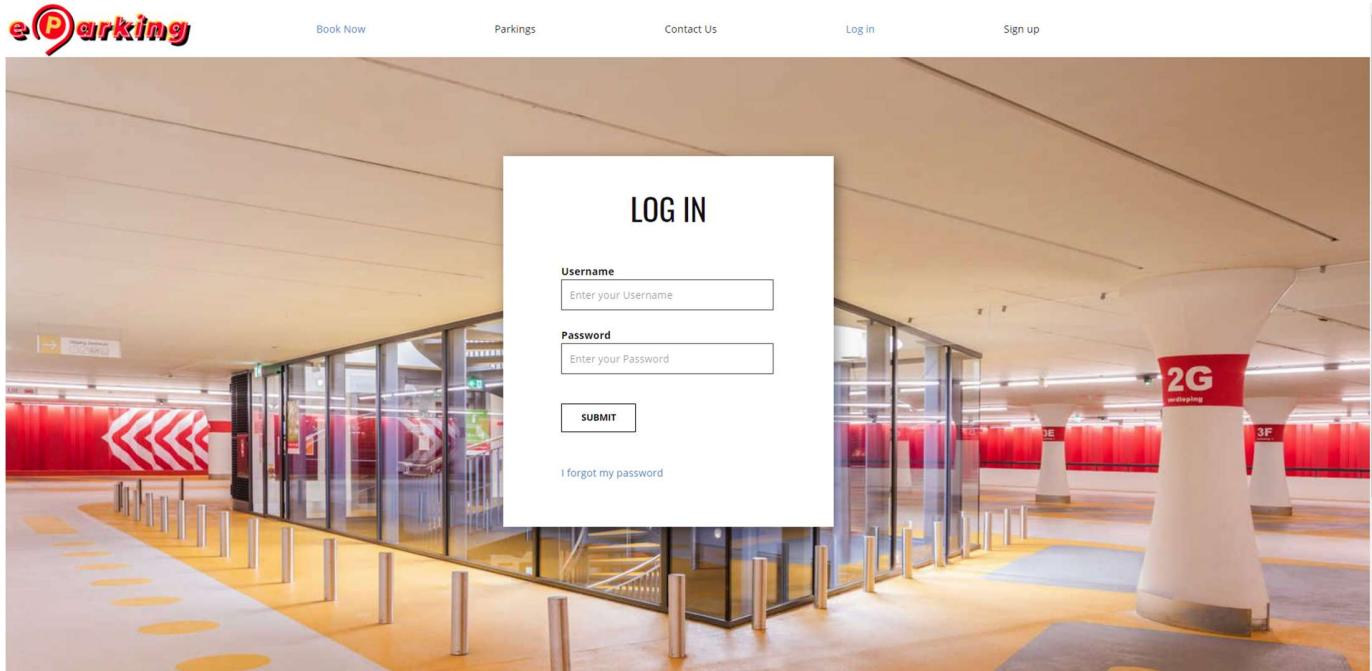
BOOK NOW

My Reservations

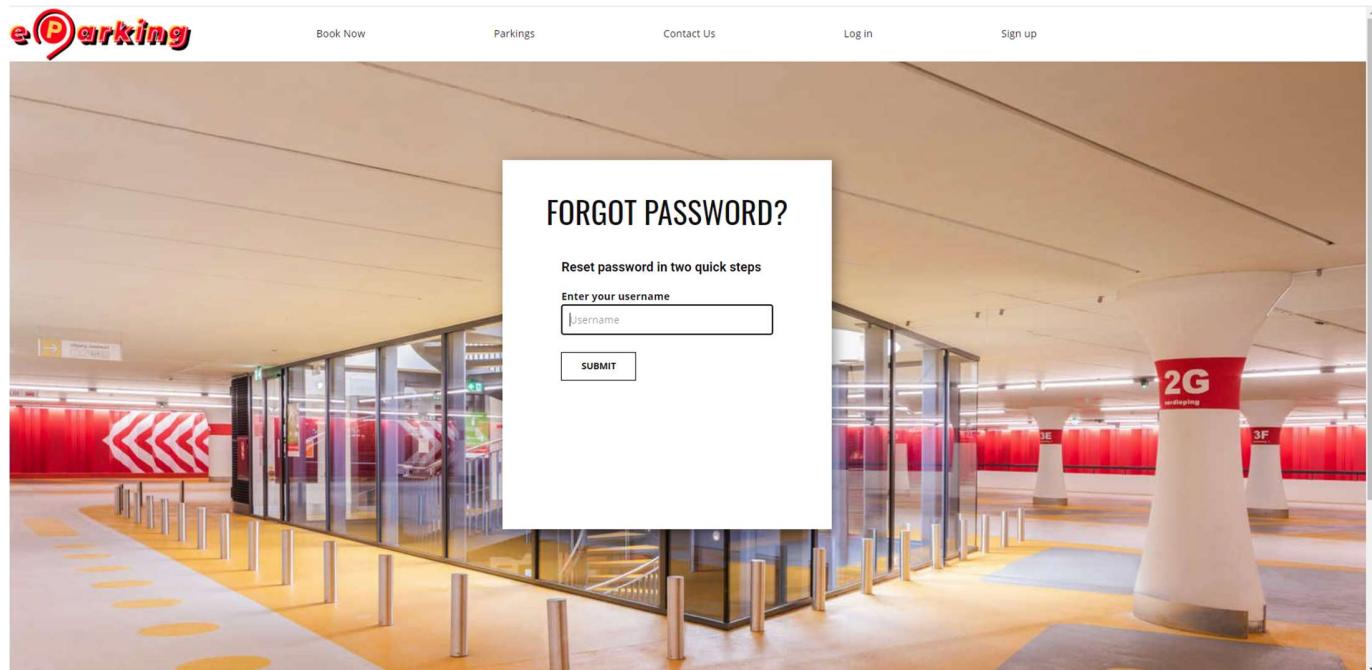
Parking Name	Slot	Plate Number	Start Date	Start Time	Finish Date	Finish Time	Active	Delete



Για την διαδικασία της σύνδεσης του χρήστη στην εφαρμογή, ο χρήστης χρησιμοποιεί την σελίδα Log In. Εκεί του ζητείται να εισάγει τα στοιχεία του και εφόσον είναι σωστά, μεταφέρεται πάλι στην σελίδα My Profile.



Αν ο χρήστης έχει ξεχάσει τον κωδικό του, τότε πατώντας το «I forgot my password» μπορεί να τον επαναφέρει. Αρχικά, θα του ζητηθεί να εισάγει το username του και μετά, θα σταλεί στο email που έχει καταχωρήσει στην εφαρμογή κατά την εγγραφή του, ένα mail που θα περιέχει το verification code που θα του ζητηθεί να εισάγει στην εφαρμογή, με σκοπό να αποδείξει την ταυτότητά του.





eparking password recovery email ➤ Εισερχόμενα x

✉ eparking.papei2021@gmail.com
προς εγώ ✉
Hi stacy.
Here is your 5-digit code to recover your password: 53197

4:32 μ.μ. (πριν από 0 λεπτά) ☆ ↗ ⋮

◀ Απάντηση ▶ Προώθηση

eParking

Book Now Parkings Contact Us Log in Sign up

SET UP NEW PASSWORD

Enter the recovery code you received in your email
53197

VERIFY CODE

Enter new password for your account

Enter your password

Confirm password

Confirm password

CHANGE PASSWORD

Τέλος, εισάγει τον καινούργιο κωδικό και πλέον μπορεί να συνδεθεί στην εφαρμογή μόνο με αυτόν.



3.2. Κώδικας μηχανισμού αυθεντικοποίησης

Παρακάτω ακολουθεί ο κώδικας που καλείται όταν ο χρήστης πατήσει το κουμπί του Sign up και εκτελεί τις λειτουργίες κρυπτογράφησης του password και δημιουργίας του τυχαίου πενταψήφιου one time password και αποστολής του αντίστοιχου email.

```
public void Submit_Click(object sender, EventArgs e)
{
    //Check if user's input is valid
    if (!Password.Text.Trim().Equals(Repeat_Password.Text.Trim()))
        ClientScript.RegisterStartupScript(this.GetType(), "alert", "alert(' Passwords must match ')", true);
    else
    {
        //Check database for duplicates (email, username)
        string notification = Auxiliary.CheckForDuplicates(Email.Text, Username.Text);
        if (notification != null)
            ClientScript.RegisterStartupScript(this.GetType(), "alert", "alert('" + notification + "')", true);
        else
        {
            //Create salt
            string salt = Convert.ToBase64String(Auxiliary.GenerateSalt());
            //Hash password
            string hashed_password = Auxiliary.HashPassword(Password.Text, Convert.FromBase64String(salt));
            //Create user object
            User user = new User(FirstName.Text, LastName.Text, Email.Text, Username.Text, hashed_password, salt);
            //Generate 5-digit code
            StringBuilder code = new StringBuilder("", 5);
            for (int i = 1; i <= 5; i++)
                code.Append(new Random(i * DateTime.Now.Millisecond).Next(10).ToString());
            //Create session variable
            Session["Verification_code"] = code.ToString();
            //Save user
            userToSignUp = user;
            //Send verification email
            notification = userToSignUp.SendVerificationEmail();
            if (notification != null)
                ClientScript.RegisterStartupScript(this.GetType(), "alert", "alert('" + notification + "')", true);
            else
                Response.Redirect("VerificationCode.aspx");
        }
    }
}
```

3.3. Ελέγχου πρόσβασης (authorization)

Έχουμε υλοποιήσει role-based authorization μέσω session management. Συγκεκριμένα, αποθηκεύουμε μέσα στο session μια boolean τιμή true ή false, για το αν ο χρήστης που συνδέθηκε είναι διαχειριστής ή όχι, καθώς και το username του.

Με αυτόν τον τρόπο, αν ο χρήστης προσπαθήσει να αποκτήσει πρόσβαση σε μια σελίδα που δεν είναι εξουσιοδοτημένος, τότε η εφαρμογή τον μεταφέρει αυτόματα σε σελίδα γενικής πρόσβασης, διαγράφοντας το περιεχόμενο του τρέχοντος session για λόγους ασφάλειας.



3.4. Κώδικας ελέγχου πρόσβασης

Ο κώδικας που είναι υπεύθυνος για τον έλεγχο, αν ο χρήστης που συνδέθηκε είναι διαχειριστής, φαίνεται παρακάτω.

```
//Check if user is admin.
public string IsAdmin()
{
    string isAdmin = null;
    string query = "select admin from users where username = @username";
    try
    {
        NpgsqlConnection connection = new NpgsqlConnection(Auxiliary.CONNECTION_STRING);
        connection.Open();
        NpgsqlCommand command = new NpgsqlCommand(query, connection);
        command.Parameters.AddWithValue("username", username);
        NpgsqlDataReader dataReader = command.ExecuteReader(); //run query
        while (dataReader.Read())
            isAdmin = dataReader[0].ToString(); //get result
        connection.Close();
        return isAdmin;
    }
    catch
    {
        return null;
    }
}
```

Η μέθοδος αυτή επιστρέφει *true* αν ο χρήστης είναι όντως διαχειριστής και *false* σε αντίθετη περίπτωση.

Όταν ο χρήστης πατήσει το κουμπί login για να συνδεθεί στον λογαριασμό του, τότε καλείται η παραπάνω μέθοδος *IsAdmin()*, και το αποτέλεσμά της μαζί με το username του χρήστη, αποθηκεύονται στο session.

```
//Check if user is admin
string isAdmin = user.IsAdmin();
if (isAdmin != null)
{
    Session["Username"] = user.Username;
    Session["UserType"] = isAdmin;
    if (isAdmin.Equals("True"))
        Response.Redirect("Admin.aspx");
    else if (isAdmin.Equals("False"))
        Response.Redirect("MyProfile.aspx");
}
else
    ClientScript.RegisterStartupScript(this.GetType(), "alert", "alert(' An exception was caught.Please report it. ')", true);
```



Κατά την φόρτωση των σελίδων, ελέγχονται οι τιμές του session και αν ο χρήστης δεν έχει πρόσβαση σε κάποια σελίδα, τότε ανακατευθύνεται σε σελίδα γενικής πρόσβασης, διαγράφοντας το περιεχόμενο του τρέχοντος session για λόγους ασφάλειας.

```
if (Session.Contents.Count == 0) //user is not supposed to use this webform, redirect to index
    Response.Redirect("HomePage.aspx");
else
{
    if ((string)HttpContext.Current.Session["UserType"] != null)
    {
        if (((string)HttpContext.Current.Session["UserType"]).Equals("True"))
            Response.Redirect("Admin.aspx");
    }
    usernamel.InnerText = "Hi " + (string)HttpContext.Current.Session["Username"];
    UsernameLabel.Text = (string)HttpContext.Current.Session["Username"];
}

user = new User((string)HttpContext.Current.Session["Username"]);
```

4. Input filtering και validation

Έχουμε υλοποιήσει έλεγχο επικύρωσης της εισόδου του χρήστη (input validation), τόσο σε συντακτικό επίπεδο (syntactic level), όσο και σε σημασιολογικό επίπεδο (semantic level).

4.1. Regular expressions

Έχουμε πραγματοποιήσει client side ελέγχους με την χρήση regular expressions, για την συμπλήρωση πεδίων όπως για το ονοματεπώνυμο, username, password, email, αριθμός πινακίδων κλπ.

```
<div class="u-form-group u-form-group-4 u-form-spacing-30 u-form-vertical"><br />
<label for="text-4f9f" class="u-label u-label-3">First Name:</label><br />
<asp:TextBox placeholder="Enter your First Name" ID="FirstName" name="name1" class="u-border-1 u-border-grey-75 u-input u-input-rectangle u-white" required runat="server"></asp:TextBox>
<asp:RegularExpressionValidator ID="RegularExpressionValidator2" runat="server" ControlToValidate="FirstName" ErrorMessage="Please enter valid name" ForeColor="Red" ValidationExpression="(?![\s.]+$)[a-zA-Z]{1,30}"> </asp:RegularExpressionValidator>
<!--<input type="text" placeholder="Enter your First Name" id="text-4f9f" name="text-1" class="u-border-1 u-border-grey-75 u-input u-input-rectangle u-white" required="required">-->

<div class="u-form-group u-form-group-5"><br />
<label for="text-4f00" class="u-label u-label-4">Last Name:</label><br />
<asp:TextBox placeholder="Enter your Last Name" ID="LastName" name="name2" class="u-border-1 u-border-grey-75 u-input u-input-rectangle u-white" required runat="server"></asp:TextBox>
<asp:RegularExpressionValidator ID="RegularExpressionValidator1" runat="server" ControlToValidate="LastName" ErrorMessage="Please enter valid name" ForeColor="Red" ValidationExpression="(?![\s.]+$)[a-zA-Z]{1,20}"> </asp:RegularExpressionValidator>
<!--<input type="text" placeholder="Enter your Last Name" id="text-4f00" name="text-2" class="u-border-1 u-border-grey-75 u-input u-input-rectangle u-white" required="required">-->

<div class="u-form-email u-form-group u-form-group-6"><br />
<label for="email-d70e" class="u-label u-label-5">Email:</label><br />
<asp:TextBox placeholder="Enter your Email" ID="Email" name="name3" class="u-border-1 u-border-grey-75 u-input u-input-rectangle u-white" required runat="server"></asp:TextBox>
<asp:RegularExpressionValidator ID="RegularExpressionValidator3" runat="server" ControlToValidate="Email" ErrorMessage="Please enter valid email address" ForeColor="Red" ValidationExpression="\w+([-.\w+]*)@\w+([-.\w+]*)\.\w+([-.\w+]*")> </asp:RegularExpressionValidator>
<!--<input type="email" placeholder="Email" id="email-d70e" name="email" class="u-border-1 u-border-grey-75 u-input u-input-rectangle u-white" required="required">-->

<div class="u-form-group u-form-name u-form-group-7"><br />
<label for="name-d70e" class="u-label u-label-6">Username:</label><br />
<asp:TextBox placeholder="Enter your Username" ID="Username" name="name4" class="u-border-1 u-border-grey-75 u-input u-input-rectangle u-white" required runat="server"></asp:TextBox>
<asp:RegularExpressionValidator ID="RegularExpressionValidator4" runat="server" ControlToValidate="Username" ErrorMessage="Please enter valid username" ForeColor="Red" ValidationExpression="(?![\s.]+$)[a-zA-Z0-9]{1,30}"> </asp:RegularExpressionValidator>
<!--<input type="text" placeholder="Enter your Username" id="name-d70e" name="name" class="u-border-1 u-border-grey-75 u-input u-input-rectangle u-white" required="required">-->

<div class="u-form-group u-form-group-8"><br />
<label for="text-28cb" class="u-label u-label-7">Password:</label><br />
<asp:TextBox type="password" placeholder="Enter your Password" ID="Password" name="name5" class="u-border-1 u-border-grey-75 u-input u-input-rectangle u-white" required runat="server"></asp:TextBox>
<asp:RegularExpressionValidator ID="RegularExpressionValidator5" runat="server" ControlToValidate="Password" ErrorMessage="Please enter valid password" ForeColor="Red" ValidationExpression="(?![\s.]+$)[a-zA-Z0-9]{6,}> </asp:RegularExpressionValidator>
<!--<input type="text" placeholder="Enter your Password" id="text-28cb" name="text" class="u-border-1 u-border-grey-75 u-input u-input-rectangle u-white" required="required">-->

<div class="u-form-group u-form-group-8"><br />
<label for="text-28cb" class="u-label u-label-7">Repeat Password:</label><br />
<asp:TextBox type="password" placeholder="Repeat Password" ID="Repeat_Password" name="name5" class="u-border-1 u-border-grey-75 u-input u-input-rectangle u-white" required runat="server"></asp:TextBox>
<asp:RegularExpressionValidator ID="RegularExpressionValidator6" runat="server" ControlToValidate="Repeat_Password" ErrorMessage="Please enter valid password" ForeColor="Red" ValidationExpression="(?![\s.]+$)[a-zA-Z0-9]{6,}> </asp:RegularExpressionValidator>
<!--<input type="text" placeholder="Enter your Password" id="text-28cb" name="text" class="u-border-1 u-border-grey-75 u-input u-input-rectangle u-white" required="required">-->
```

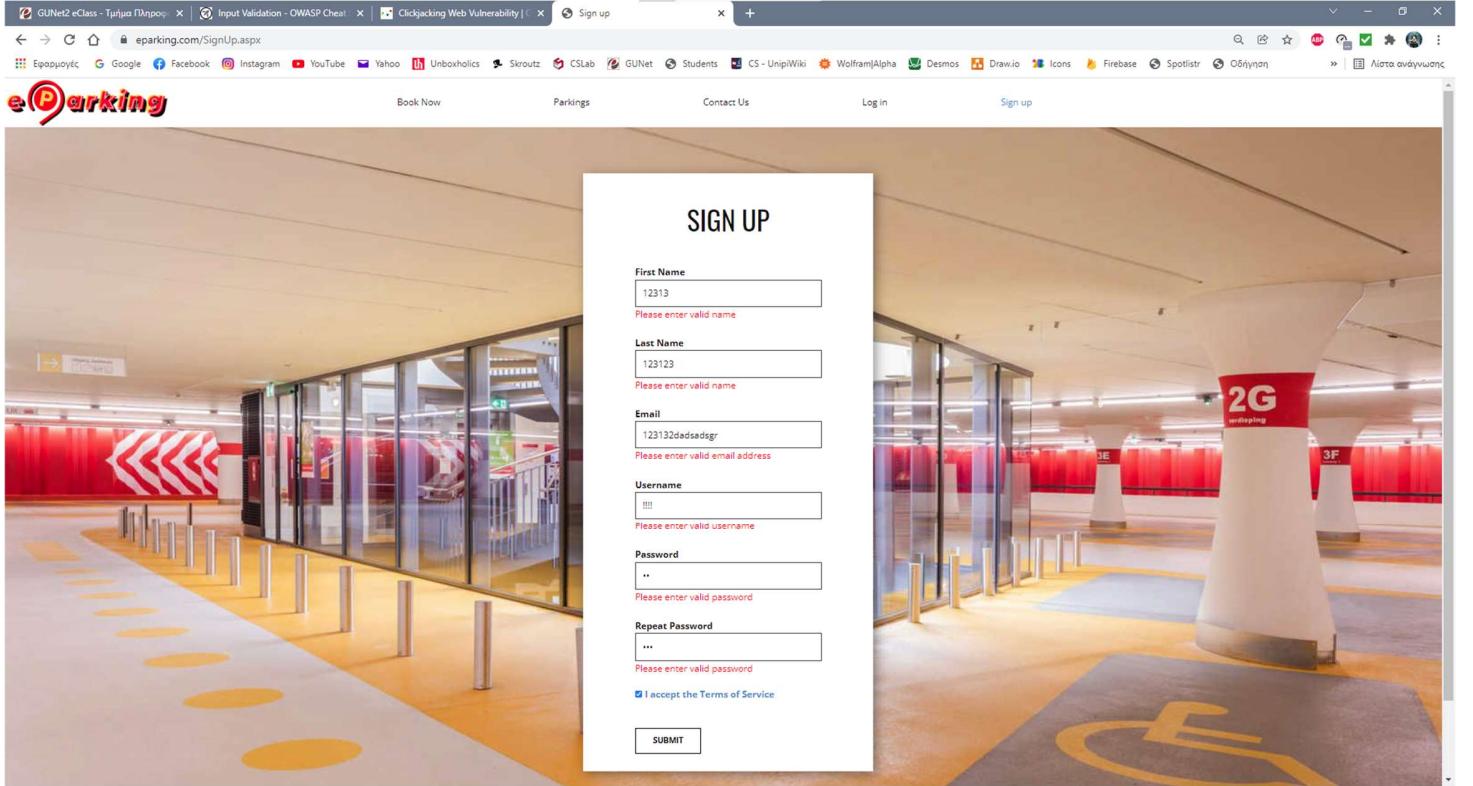
Κώδικας σελίδας Sign up



```
div class="u-form-group u-form-group-4 u-form-spacing-30 u-form-vertical">  
<br />  
<label for="text-47f9" class="u-label u-label-3">Enter your plate number</label><br />  
<asp:TextBox placeholder="plate number" ID="PlateNumberTextBox" name="name1" class="u-border-1 u-border-grey-75 u-input u-input-rectangle u-white" runat="server"></asp:TextBox>  
<asp:RegularExpressionValidator ID="RegularExpressionValidator1" runat="server" ControlToValidate="PlateNumberTextBox" ErrorMessage="Please enter valid plate number" ForeColor="Red" ValidationExpression="([A-Z]{3}[0-9]{3,4})" > </asp:RegularExpressionValidator>  
<!--<input type="text" placeholder="Enter your First Name" id="text-47f9" name="text-1" class="u-border-1 u-border-grey-75 u-input u-input-rectangle u-white" required="required">-->  
</div>
```

Κώδικας σελίδας Insert car

Προφανώς, αντίστοιχοι έλεγχοι υπάρχουν και σε άλλες σελίδες όπως Contact us, Login κλπ.



4.2. Email Address Validation

Για την επικύρωση του email, πέρα από regular expression έλεγχο σε client side, γίνεται έλεγχος και από την πλευρά του application server με την χρήση της κλάσης `EmailAddressAttribute` και της μεθόδου της `IsValid()`.

```
if (!new EmailAddressAttribute().IsValid(Email.Text.Trim())) //returns true if email is valid  
{  
    ClientScript.RegisterStartupScript(this.GetType(), "alert", "alert(' Invalid email ')", true);  
    return;  
}
```

4.3. Date Validation

Για τον έλεγχο έγκυρων σημασιολογικά ημερομηνιών πρέπει να ισχύουν τα εξής:

- Η ημερομηνία του check out να είναι μεταγενέστερη του check in



- Ο χρήστης να έχει επιλέξει μια ώρα κατά την οποία το επιλεγμένο parking να είναι ανοιχτό (σύμφωνα με τις ώρες λειτουργίας).
- Οι ημερομηνίες check in και check out να μην είναι παρελθοντικές.

```
bool flag = true; //valid check in and check out dates and times
bool flag2 = true; //check in and check out dates are not past dates and times
bool flag3 = true; //valid working hours

List<string> workingHours = Reservation.GetWorkingHours(ParkingList.Text);
TimeSpan parkingStartTime = TimeSpan.Parse(workingHours[0]);
TimeSpan parkingFinishTime = TimeSpan.Parse(workingHours[1]);
if ((parkingStartTime <= TimeSpan.Parse(time1 + ":00") && TimeSpan.Parse(time1 + ":00") <= parkingFinishTime)
    && (parkingStartTime <= TimeSpan.Parse(time2 + ":00") && TimeSpan.Parse(time2 + ":00") <= parkingFinishTime))...
else...

reservation = new Reservation(ParkingList.Text, DateTextBox1.Text, TimeTextBox1.Text, DateTextBox2.Text, TimeTextBox2.Text, true);

string notification;
if (!flag)
{
    notification = "Check out date and time cannot be sooner than Check in.";
    ClientScript.RegisterStartupScript(this.GetType(), "alert", "alert('" + notification + "')", true);
    return;
}
if (!flag2)
{
    notification = "Check in date cannot be a past date.";
    ClientScript.RegisterStartupScript(this.GetType(), "alert", "alert('" + notification + "')", true);
    return;
}
else if (!flag3)
{
    string notification1 = "Parking is closed in this timespan. Please choose another time.";
    ClientScript.RegisterStartupScript(this.GetType(), "alert", "alert('" + notification1 + "')", true);
    return;
}
```

Ο ιστότοπος www.eparking.com λέει

Check in date cannot be a past date.

OK

Ο ιστότοπος www.eparking.com λέει

Check out date and time cannot be sooner than Check in.

OK

Ο ιστότοπος www.eparking.com λέει

Parking is closed in this timespan. Please choose another time.

OK



4.4. SQL και XSS injection

SQL injection είναι μια τεχνική έγχυσης κώδικα, που χρησιμοποιείται για την επίθεση σε εφαρμογές λογισμικού που χρησιμοποιούν βάση δεδομένων, στην οποία εισάγονται κακόβουλα SQL statements προς εκτέλεση σε ένα πεδίο εισαγωγής δεδομένων της εφαρμογής. Ο εισβολέας θα στείλει μια ειδικά κατασκευασμένη δήλωση SQL που έχει σχεδιαστεί για να προκαλέσει κάποια κακόβουλη ενέργεια.

Για να αποφύγουμε τέτοιου είδους επιθέσεις, χρησιμοποιούμε parameterized queries με σκοπό την προμεταγλώττιση του κώδικα SQL, διαχωρίζοντάς τον από τα δεδομένα.

```
//Stores user's info(firstname, lastname, email, username,hashedpassword,salt) in DB when he signs up.
public string StoreUserInfoToDB()
{
    string query = "insert into users values (@firstname, @lastname, @email, @username, @hashedPassword, @salt, false)";
    int rowsAffected = -1; //false value
    try
    {
        NpgsqlConnection connection = new NpgsqlConnection(Auxiliary.CONNECTION_STRING);
        connection.Open();
        //define query's parameters
        NpgsqlCommand command = new NpgsqlCommand(query, connection);
        command.Parameters.AddWithValue("firstname", First_name);
        command.Parameters.AddWithValue("lastname", Last_name);
        command.Parameters.AddWithValue("email", Email);
        command.Parameters.AddWithValue("username", Username);
        command.Parameters.AddWithValue("hashedPassword", Hashed_Password);
        command.Parameters.AddWithValue("salt", Salt);
        rowsAffected = command.ExecuteNonQuery(); //run query
        connection.Close();
    }
    catch
    {
        return "Sign up failed. Please report this error through the \"Contact us\" page.";
    }
    if (rowsAffected == -1)
        return "Sign up failed. There is a problem with the database. Please report this error through the \"Contact us\" page.";
    else
        return null;
}
```

Με τον όρο Cross-site Scripting ή XSS αναφερόμαστε σε μία ευπάθεια ασφάλειας που επιτρέπει σε έναν κακόβουλο χρήστη να εγχύσει κώδικα JavaScript σε μία ιστοσελίδα.

Για να αποτρέψουμε τέτοιου είδους επιθέσεις, προσθέσαμε την παρακάτω γραμμή κώδικα (έντονα τονισμένη με κίτρινο χρώμα) στο Web.config αρχείο της εφαρμογής μας.

```
1  <?xml version="1.0" encoding="utf-8"?>
2
3  <!--
4      For more information on how to configure your ASP.NET application, please visit
5      https://go.microsoft.com/fwlink/?LinkId=169433
6      -->
7  <configuration>
8      <system.web>
9          <compilation debug="true" targetFramework="4.6.1"/>
10         <httpRuntime targetFramework="4.6.1"/>
11         <pages buffer="true" validateRequest="true" />
12     </system.web>
```



5. Εύρεση ευπαθειών ασφάλειας

Πραγματοποιήσαμε αυτοματοποιημένο έλεγχο ασφαλείας με την χρήση του προτεινόμενου εργαλείου *Zap*. Εισαγάγαμε το url της ιστοσελίδας μας ως url στόχο και πραγματοποιήσαμε την επίθεση.

The screenshot shows the OWASP ZAP 2.11.1 interface. The main window title is "Automated Scan". It contains a form to enter the URL to attack (https://www.eparking.com), options for spiders (Use traditional spider: ; Use ajax spider: with Firefox Headless), and a progress bar indicating the attack is complete. Below the main window, the "Alerts" tab is open, displaying a list of findings under the "Eidotopoiisis" category. Some findings are expanded, such as "Absence of Anti-CSRF Tokens" (6), "Cookies Without Secure Flag" (1), and "Incomplete or No Cache-control Header Set" (9). The bottom status bar shows "Primary Proxy: localhost:8081".

Στο κάτω αριστερό μέρος της παραπάνω εικόνας, απεικονίζονται οι ευπάθειες που εντόπισε το εργαλείο.

5.1. Absence of Anti-CSRF Tokens

Το Cross-Site Request Forgery (CSRF) είναι ένας τύπος επίθεσης που συμβαίνει όταν ένας κακόβουλος ιστότοπος, ηλεκτρονικό ταχυδρομείο, ιστολόγιο, άμεσο μήνυμα ή πρόγραμμα αναγκάζει το πρόγραμμα περιήγησης ιστού ενός χρήστη να εκτελέσει μια ανεπιθύμητη ενέργεια σε έναν αξιόπιστο ιστότοπο κατά τον έλεγχο ταυτοποίησης του χρήστη.

Δεν έχουμε εισάγει Anti-CSRF tokens για να προστατευτούμε από τέτοιου είδους επιθέσεις.



5.2. Cookie Without Secure Flag

Χρησιμοποιώντας αυτήν την ευπάθεια, ένας εισβολέας μπορεί να ανακατευθύνει τον χρήστη σε κακόβουλο ιστότοπο για κλοπή πληροφοριών/δεδομένων και να εμφανίσει ψευδή δεδομένα στον χρήστη, τα οποία με τη σειρά τους θα επηρεάσουν την αξιοπιστία του ιστοτόπου.

Όταν χρησιμοποιείται ένα πρωτόκολλο HTTP για επικοινωνία μεταξύ πελάτη και εξυπηρετητή, η κίνηση δεδομένων αποστέλλεται σε απλό κείμενο. Ένα HTTP request επιτρέπει στον εισβολέα να δει/τροποποιήσει την κυκλοφορία χρησιμοποιώντας μια επίθεση Man-In-The-Middle (MITM). Το HTTPS είναι μια ασφαλής έκδοση του HTTP. Αυτό το πρωτόκολλο χρησιμοποιεί SSL/TLS για την προστασία των δεδομένων στο επίπεδο εφαρμογής. Το HTTPS χρησιμοποιείται για καλύτερο έλεγχο ταυτότητας και ακεραιότητα δεδομένων. Ένα secure flag ορίζεται από τον εξυπηρετητή της εφαρμογής κατά την αποστολή ενός νέου cookie στον χρήστη χρησιμοποιώντας ένα HTTP response. Το secure flag χρησιμοποιείται για την αποτροπή παρατήρησης και χειραγώγησης cookie από μη εξουσιοδοτημένα μέρη. Αυτό συμβαίνει επειδή το cookie αποστέλλεται ως κανονικό κείμενο. Ένα πρόγραμμα περιήγησης δεν θα στείλει ένα cookie με το secure flag που αποστέλλεται μέσω μη κρυπτογραφημένου αιτήματος HTTP. Δηλαδή, ορίζοντας το secure flag το πρόγραμμα περιήγησης θα αποτρέψει/διακόψει τη μετάδοση ενός cookie σε ένα μη κρυπτογραφημένο κανάλι.

Παρόλα αυτά, ακόμα κι αν η ίδια η ιστοσελίδα αποσταλεί μέσω HTTPS, ένας εισβολέας θα μπορούσε να κλέψει το session που χρησιμοποιείται αναγκάζοντας τον χρήστη να κάνει ένα αίτημα HTTP και στη συνέχεια, κλέβοντας εκεί το cookie του session.

5.3. Incomplete or No Cache-control Header Set

Επιτρέπουμε στον browser να αποθηκεύσει προσωρινά δεδομένα, κάτι που δεν είναι απαραίτητα αρνητικό.

5.4. Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)

Ο web/application server διαρρέει πληροφορίες μέσω ενός ή περισσότερων "X-Powered-By" HTTP response headers. Η πρόσβαση σε τέτοιες πληροφορίες μπορεί να διευκολύνει τους εισβολείς να προσδιορίσουν άλλα frameworks/components στα οποία βασίζεται η εφαρμογή και τις ευπάθειες στις οποίες ενδέχεται να υπόκεινται αυτά.

5.5. X-AspNet-Version Response Header

Ο server διαρρέει πληροφορίες μέσω των "X-AspNet-Version"/"X-AspNetMvc-Version" HTTP response headers. Ένας εισβολέας μπορεί να χρησιμοποιήσει αυτές τις πληροφορίες για να εκμεταλλευτεί γνωστά τρωτά σημεία.



5.6. X-Content-Type-Options Header Missing

Το MIME sniffing ήταν, και εξακολουθεί να είναι, μια τεχνική που χρησιμοποιείται από ορισμένα προγράμματα περιήγησης ιστού (κυρίως τον Internet Explorer) για την εξέταση του περιεχομένου ενός συγκεκριμένου στοιχείου. Αυτό γίνεται για τον προσδιορισμό της μορφής αρχείου ενός στοιχείου. Αυτή η τεχνική είναι χρήσιμη σε περίπτωση που δεν υπάρχουν αρκετές πληροφορίες metadata για ένα συγκεκριμένο στοιχείο, αφήνοντας έτσι την πιθανότητα το πρόγραμμα περιήγησης να ερμηνεύει εσφαλμένα το στοιχείο.

Αν και το MIME sniffing μπορεί να είναι χρήσιμο για τον προσδιορισμό της σωστής μορφής αρχείου ενός στοιχείου, μπορεί επίσης να προκαλέσει μια ευπάθεια ασφαλείας. Αυτή η ευπάθεια μπορεί να είναι αρκετά επικίνδυνη τόσο για τους ιδιοκτήτες του ιστοτόπου όσο και για τους επισκέπτες του ιστοτόπου. Αυτό συμβαίνει επειδή ένας εισβολέας μπορεί να αξιοποιήσει το MIME sniffing για να στείλει μια επίθεση XSS (Cross Site Scripting).

5.7. Timestamp Disclosure – Unix

Η ευπάθεια αυτή στην περίπτωσή μας δεν έχει βάση, γιατί διαβάζει εσφαλμένα ως timestamp μια τιμή ενός χαρακτηριστικού σε ένα CSS αρχείο.

The screenshot shows the OWASP ZAP interface. In the top navigation bar, the tabs are: Αρχικό, Επεξεργασία, Άποψη, Ανάλυση, Αναφορά, Εργολαία, Εισαγωγή, Σε σύνδεση, Βοήθεια. The main pane displays a captured HTTP request and response. The response body shows a CSS file containing a timestamp disclosure vulnerability. The bottom pane shows a tree view of findings, with 'Timestamp Disclosure - Unix' listed under 'Ειδομοίσης' (Findings). The details for this finding include:

- Timestamp Disclosure - Unix
- URL: https://www.sparking.com/page.css
- Πρόσω: Low
- Εμπροσθόντ: Low
- Παραδίσος:
- Έναρξη:
- Τελετή:
- CWE ID: 200
- WASC ID: 13
- Πηγή: Ποθητική (10096 - Timestamp Disclosure)
- Παραγερή:
- A timestamp was disclosed by the application/web server - Unix
- Άλλες Πληροφορίες:
- 11111111, which evaluates to: 1970-05-09 16:25:11



5.8. Clickjacking X Frame Options Header Missing

Clickjacking είναι όταν ένας εισβολέας χρησιμοποιεί πολλαπλά διαφανή ή αδιαφανή επίτεδα για να ξεγελάσει έναν χρήστη ώστε να κάνει κλικ σε ένα κουμπί ή έναν σύνδεσμο σε μια άλλη σελίδα όταν σκόπευε να κάνει κλικ στη σελίδα του ανώτατου επιπέδου. Έτσι, ο εισβολέας "πειρατεύει" τα κλικ που προορίζονται για τη σελίδα του και τα δρομολογεί σε άλλη σελίδα, που πιθανότατα ανήκει σε άλλη εφαρμογή, τομέα ή και τα δύο.

Καταφέραμε να καλύψουμε αυτήν την ευπάθεια προσθέτοντας στον IIS ένα custom HTTP response header και τροποποιώντας κατάλληλα το Web.config αρχείο μας όπως φαίνεται παρακάτω.

The screenshot shows the IIS Manager interface. The left sidebar shows the connection tree with 'ANASTASIA (ANASTASIA\anast)' selected. Under 'Sites', 'Default Web Site' and 'eparking' are listed. The main pane is titled 'HTTP Response Headers' and contains a table with two rows:

Name	Value	Entry Type
X-Frame-Options	SAMEORIGIN	Local
X-Powered-By	ASP.NET	Inherited

The 'Actions' pane on the right has buttons for 'Add...', 'Set Common Headers...', and 'Help'.

```
<system.webServer>
    <httpProtocol>
        <customHeaders>
            <add name="X-Frame-Options" value="SAMEORIGIN" />
        </customHeaders>
    </httpProtocol>
</system.webServer>
```



6. Βιβλιογραφία

1. **Στέφανος Γκρίζαλης και άλλοι.** Ασφάλεια Πληροφοριών και Συστημάτων στον Κυβερνοχώρο. Αθήνα: Εκδόσεις Νέων Τεχνολογιών, 2021.
2. Όλο το υλικό των εργαστηριακών παραδειγμάτων και των διαλέξεων.

Ηλεκτρονική Βιβλιογραφία:

1. <https://www.keycdn.com/support/what-is-mime-sniffing>
2. <https://support.detectify.com/support/solutions/articles/48001048982-cookie-lack-secure-flag>
3. <https://beaglesecurity.com/blog/vulnerability/cookie-session-without-secure-flag.html>
4. https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html
5. <https://stackoverflow.com/questions/26420656/drawbacks-of-using-validateRequest-false-asp-net>
6. <https://www.pico.net/kb/how-do-you-get-chrome-to-accept-a-self-signed-certificate/>
7. <https://stackoverflow.com/questions/8169999/how-can-i-create-a-self-signed-cert-for-localhost>
8. <https://www.valencynetworks.com/kb/clickjacking-x-frame-options-header-missing.html>
9. <https://www.first.org/cvss/>
10. <https://www.tutorialsteacher.com/articles/install-ssl-certificate-in-localhost-website-iis>
11. <https://www.zaproxy.org/>
12. <https://eclass.aueb.gr/modules/document/index.php?course=INF104&download=/5eae9925dZmD/5eaeb43aGK3o.pdf>