ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Τμήμα Πληροφορικής

Εικόνα που περιέχει ορθογώνιο παραλληλόγραμμο, σχεδίαση

Το περιεχόμενο που δημιουργείται από τεχνολογία AI ενδέχεται να είναι εσφαλμένο.

Εργασία Μαθήματος ***Σχεδίαση Αρχιτεκτονικών Ασφαλείας***

|  |  |
| --- | --- |
| Αρ. Άσκησης - Τίτλος Άσκησης | ***Application Threat Modelling*** |
| Όνομα φοιτητή - Αρ. Μητρώου | Ραυτόπουλος Μάριος – ΜΠΚΕΔ24034 |
| Ημερομηνία παράδοσης | 05/03/25 |

Εκφώνηση Εργασίας:

Θα πρέπει να επιλέξετε μια web εφαρμογή την οποία έχετε αναπτύξει σε κάποιο άλλο μάθημα (ή το eclass) ώστε να διενεργήσετε Application Threat Modelling λαμβάνοντας υπόψη σας:

* τις περαιτέρω οδηγίες και παραδείγματα από το site του OWASP (<https://www.owasp.org/index.php/Application_Threat_Modeling>)
* Για τα μέτρα προστασίας θα επιλέξετε τις βασικές κατηγορίες του OWASP ASVS (<https://www.owasp.org/index.php/Category:OWASP_Application_Security_Verification_Standard_Project>, <https://owasp.org/www-chapter-frankfurt/assets/slides/43_OWASP_Frankfurt_Stammtisch_1.pdf>
* Αξιολογήστε τους κινδύνους με βάση τη μεθοδολογία DREAD Threat Modeling (https://threat-modeling.com/dread-threat-modeling/)

Το παραδοτέο θα είναι σε μορφή doc ή docx και θα περιλαμβάνει τουλάχιστον τα υπόλοιπα:

* τα βήματα που προτείνει η μεθοδολογία του OWASP (<https://www.owasp.org/index.php/Application_Threat_Modeling>)
* τα αποτελέσματα της αξιολόγησης.

Περιεχόμενα

[1. Εισαγωγή: 3](#_Toc191480884)

[1.1 Σκοπός Εγγράφου 3](#_Toc191480885)

[1.2 Πεδίο Εφαρμογής 4](#_Toc191480886)

[1.3 Μεθοδολογία 4](#_Toc191480887)

[2. Βήματα OWASP Application Threat Modeling 4](#_Toc191480888)

[2.1 Step1: Scope your Work 4](#_Toc191480889)

[2.2 Step 2: Determine Threats 9](#_Toc191480890)

[2.3 Step 3: Determine Countermeasures and Mitigation 18](#_Toc191480891)

[2.4 Step 4: Assess your work 21](#_Toc191480892)

[3. Συμπεράσματα: 22](#_Toc191480893)

[4. Πηγές: 23](#_Toc191480894)

# 1. Εισαγωγή:

Σε αυτό το έγγραφο πραγματοποιείται Application Threat Modeling για μια e-learning Web εφαρμογή (π.χ. “Open eClass”). Ακολουθούνται κυρίως τις κατευθύνσεις του OWASP Application Threat Modeling σε συνδυασμό με το μοντέλο STRIDE για την κατηγοριοποίηση των απειλών και το μοντέλο DREAD για την αξιολόγηση και την ποσοτικοποίησή τους. Τέλος, υιοθετούνται βέλτιστες πρακτικές από το OWASP ASVS για τα απαραίτητα μέτρα προστασίας. Για την οπτική αναπαράσταση των στοιχείων και των αλληλεπιδράσεων της εφαρμογής χρησιμοποιείται το Microsoft Threat Modeling Tool (MTMT) 2016.

## 1.1 Σκοπός Εγγράφου

* Να παρουσιάσει τα βήματα του Threat Modeling (OWASP-based).
* Να αναδείξει τις κύριες απειλές με STRIDE.
* Να βαθμολογήσει (DREAD) τους κινδύνους.
* Να προτείνει αντίμετρα ασφάλειας βάσει OWASP ASVS.

## 1.2 Πεδίο Εφαρμογής

Εξετάζεται η e-learning πλατφόρμα (Open eClass) με δύο είδη χρηστών:

* Student (πρόσβαση σε υλικό, βαθμολογίες, ανέβασμα εργασιών).
* Teacher (ανέβασμα εκπαιδευτικού υλικού, βαθμολογίες, διαχείριση περιεχομένου).

## 1.3 Μεθοδολογία

Η μεθοδολογία που ακολουθείται βασίζεται στο OWASP Threat Modeling Process και περιλαμβάνει τα εξής βήματα:

* Βήμα 1: Scope your work – Αναγνώριση περιουσιακών στοιχείων, ροών δεδομένων, ορίων εμπιστοσύνης και επιφανειών επίθεσης.
* Βήμα 2: Determine Threats – Χρήση μιας δομημένης προσέγγισης για την ανακάλυψη ευπαθειών και την κατηγοριοποίησή τους με το μοντέλο STRIDE.
* Βήμα 3: Determine Countermeasures and Mitigation

Ορισμός μέτρων ασφαλείας και βέλτιστων πρακτικών για τη μείωση των κινδύνων.

* Βήμα 4: Assess your work– Εκτίμηση της αποτελεσματικότητας του Threat Model και βελτιστοποίηση των μετριασμών βάσει της ανάλυσης κινδύνου.

# 2. Βήματα OWASP Application Threat Modeling

## 2.1 Step1: Scope your Work

### 2.1.1 Threat Model Information:

Όνομα Εφαρμογής: Open eClass

Έκδοση: 1.0

Σκοπός: Παροχή διαδικτυακών εκπαιδευτικών υπηρεσιών σε φοιτητές και εκπαιδευτικούς. Οι φοιτητές μπορούν να εγγραφούν σε μαθήματα, να κατεβάζουν εκπαιδευτικό υλικό και να ανεβάζουν εργασίες. Οι εκπαιδευτικοί μπορούν να δημιουργούν/επεξεργάζονται μαθήματα, να ανεβάζουν σημειώσεις/διαφάνειες, να διαχειρίζονται βαθμολογίες και να βλέπουν αναφορές προόδου φοιτητών.

Κεντρικές Λειτουργίες:

Διαχείριση Μαθημάτων: Ο εκπαιδευτικός δημιουργεί/διαχειρίζεται το μάθημα, ανεβάζει εκπαιδευτικό υλικό.

Υποβολή Εργασιών: Ο φοιτητής ανεβάζει την εργασία του, και ο εκπαιδευτικός τη βαθμολογεί.

Ενημέρωση & Συνεργασία: Ανακοινώσεις, δημοσίευση βαθμών.

Ιδιοκτήτης εγγράφου: -

Συμμετέχοντες: -

Κριτής: -

### 2.1.2 External Dependencies:

HTTPS / TLS: Απαιτείται κρυπτογραφημένη επικοινωνία μεταξύ browser και server.

Web Server: Apache

Firewall: Προστατεύει το εσωτερικό δίκτυο, φιλτράρει απροσδόκητα αιτήματα.

Database (MySQL): Αποθήκευση δεδομένων χρηστών, μαθημάτων, βαθμολογιών.

File Storage: Χώρος αποθήκευσης των αρχείων (σημειώσεις, διαφάνειες, εργασίες).

### 2.1.3 Entry Points:

Τα Entry Points είναι τα «κανάλια» μέσω των οποίων εισέρχονται αιτήματα / δεδομένα στην εφαρμογή.

1. Login/Authentication Page

Δέχεται credentials (username/password) από τον χρήστη (Student/Teacher) για είσοδο στο σύστημα.

2. File Upload Endpoint

Ο φοιτητής ανεβάζει εργασία ή ο εκπαιδευτικός ανεβάζει εκπαιδευτικό υλικό. Αποτελεί κρίσιμο σημείο, γιατί εισέρχονται αρχεία στο σύστημα.

3. Course Management / Content Creation

Ο εκπαιδευτικός εισάγει δεδομένα (τίτλος μαθήματος, περιγραφή) στο σύστημα.

4. Password Reset (Forgot Password) Endpoint

Δέχεται email ή username για την εκκίνηση της διαδικασίας επαναφοράς κωδικού.

### 2.1.4 Assets:

1. User Credentials (Όνομα χρήστη / Κωδικός, διακριτά για φοιτητή ή εκπαιδευτικό).

2. Βαθμολογίες / Αξιολογήσεις (επικίνδυνο να τροποποιηθούν παράνομα).

3. Εκπαιδευτικό Υλικό (σημειώσεις, διαφάνειες, διαλέξεις) με πνευματικά δικαιώματα

4. Υποβληθείσες Εργασίες (Assignments) των φοιτητών, με πνευματικά δικαιώματα.

5. Φήμη / Αξιοπιστία Οργανισμού (αν η εφαρμογή παραβιαστεί ή σταματήσει να λειτουργεί, πλήττεται το κύρος του ιδρύματος).

### 2.1.5 Trust Levels:

1. Anonymous (Untrusted)
   * Ο χρήστης που δεν έχει καν λογαριασμό ή δεν έχει κάνει login. Έχει πρόσβαση μόνο σε ελάχιστες/δημόσιες σελίδες (π.χ. homepage, public info).
2. User with Invalid Credentials (Unsuccessful Login)
   * Κάποιος προσπαθεί να συνδεθεί (login) αλλά απέτυχε ή δεν έχει δικαιώματα. Θεωρείται αναξιόπιστος (επίπεδο “low trust”).
3. Student (Valid Credentials)
   * Ρόλος μεσαίου επιπέδου εμπιστοσύνης. Έχει πρόσβαση μόνο στις δικές του εργασίες, μπορεί να δει υλικό/βαθμολογία του, αλλά όχι να διαχειριστεί τα δεδομένα άλλων φοιτητών.
4. Teacher (Higher Trust)
   * Εκπαιδευτικός με δικαιώματα δημιουργίας/επεξεργασίας περιεχομένου (upload/download αρχείων, αλλαγή βαθμολογιών, προσθήκη ανακοινώσεων). Πιθανόν ευαίσθητος στόχος, καθώς χειρίζεται πολλά δεδομένα.
5. Admin / System Administrator (Highest Trust)
   * (Προαιρετικός ρόλος) Μπορεί να προσθέσει/αφαιρέσει χρήστες, να ρυθμίσει την πλατφόρμα, να δει logs. Έχει τα μέγιστα δικαιώματα, άρα αποτελεί “high value target”.

### 2.1.6 Exit Points:

Τα Exit Points είναι τα «κανάλια» μέσω των οποίων φεύγουν δεδομένα από το σύστημα προς τον χρήστη ή προς εξωτερικά συστήματα.

1. File Download Endpoint

Ο φοιτητής ή ο εκπαιδευτικός κατεβάζει αρχεία (PDF, διαφάνειες, κ.λπ.) ή βαθμολογημένες εργασίες.

Αποτελεί σημείο «εξόδου» δεδομένων από το σύστημα προς τον πελάτη (browser).

2. Logout Endpoint

Παρότι δεν «επιστρέφει» μεγάλο όγκο δεδομένων, θεωρείται τεχνικά σημείο «εξόδου» από την authenticated session. Μπορεί να τερματίζει ή να καθαρίζει session tokens.

### 2.1.7 Entities:

1. Student (Χρήστης Φοιτητής)
   * Μπορεί να κάνει login, να εγγραφεί σε μαθήματα, να βλέπει ανακοινώσεις, να ανεβάζει εργασίες, να βλέπει μόνο τα δικά του στοιχεία.
2. Teacher (Χρήστης Εκπαιδευτικός)
   * Μπορεί να δημιουργήσει/επεξεργαστεί μαθήματα, να ανεβάζει εκπαιδευτικό υλικό, να βαθμολογεί εργασίες, να προβάλει στατιστικά επίδοσης.
3. Web Client / Browser
   * Το λογισμικό (frontend) που τρέχει στον υπολογιστή/κινητό του χρήστη. Στέλνει αιτήματα στο server (HTTPS), λαμβάνει HTML/JSON/CSS.
4. Web Server
   * Η βασική εφαρμογή (business logic, authorization) που διαχειρίζεται τα αιτήματα χρηστών, εφαρμόζει κανόνες ασφάλειας (session management κ.ο.κ.) και συνδέεται στη βάση δεδομένων.
5. Database
   * Αποθηκεύει δεδομένα χρηστών, στοιχεία μαθημάτων, βαθμολογίες.
6. File Storage / File System
   * Χώρος αποθήκευσης των αρχείων (pdfs, docs, zip) που ανεβάζουν εκπαιδευτικοί ή φοιτητές (σημειώσεις, εργασίες κ.λπ.).
7. Optional: Admin (Διαχειριστής Συστήματος)
   * Πρόσωπο ή ομάδα με δικαιώματα ρύθμισης υποδομών (εγκατάσταση ενημερώσεων, ρυθμίσεις server, σύνδεση με SMTP). Άμεσα υπεύθυνος για ζητήματα ασφάλειας, εφεδρικών αντιγράφων (backups), logs κ.λπ.
8. Όρια ασφαλείας (Security Boundaries):

Boundary #1: Ανάμεσα στο Public Internet και τον Web Client / Web Server

* + Ο χρήστης βρίσκεται σε ένα απρόβλεπτο/μη ελεγχόμενο περιβάλλον (διαδίκτυο).
  + Τα αιτήματα φτάνουν στον Web Server μέσω HTTPS.
  + Εφόσον βρισκόμαστε στο Διαδίκτυο, τα δεδομένα πρέπει να κρυπτογραφούνται (TLS) για να αποφευχθεί υποκλοπή (network sniffing).
* Κίνδυνοι:
  + Κακόβουλοι χρήστες μπορούν να στέλνουν «πειραγμένα» αιτήματα (SQLi, XSS payloads, τεράστια αρχεία για DoS, κ.λπ.).
  + Επιθέσεις credential stuffing, brute force attack.

Boundary #2: Ανάμεσα στον Web Server και τα Εσωτερικά Συστήματα (Database, File System)

* + Ο Web Server επικοινωνεί μέσω ιδιωτικού δικτύου με τη Database και το File System.
  + Ο ίδιος ο τελικός χρήστης (Student/Teacher) *δεν* έχει απευθείας πρόσβαση στη DB ή στο filesystem.
  + Ο Web Server λειτουργεί ως «μεσολαβητής» (mediator): παίρνει τα αιτήματα από τον browser, κάνει έλεγχο πρόσβασης (authorization), επικοινωνεί με τη βάση για να διαβάσει/γράψει δεδομένα ή με το filesystem για ανέβασμα/κατέβασμα αρχείων.
* Κίνδυνοι:
  + Αν ο Web Server παραβιαστεί (π.χ. μέσω RCE ή credentials leak), μπορεί να αποκτήσει κάποιος πρόσβαση στη βάση ή στα αρχεία.
  + Αν δεν υπάρχουν κατάλληλοι έλεγχοι (privileges, firewalls σε επίπεδο εσωτερικού δικτύου), ένας επιτιθέμενος μπορεί να μιμηθεί τον Web Server και να κάνει κακόβουλα queries στη DB.

### 2.1.8 Data Flow Diagrams:

A diagram of a computer system

AI-generated content may be incorrect.

### 2.1.9 Data Flows Analysis

Login / Authentication:

User (Student/Teacher) ανοίγει το login page.

Πληκτρολογεί username/password → γίνεται HTTPS POST αίτημα στον Web Server.

Web Server (Application Logic) ελέγχει τα credentials στη Database (π.χ. SELECT \* FROM users WHERE username=?).

Αν η αυθεντικοποίηση πετύχει, ο Web Server δημιουργεί session/token και το επιστρέφει στον browser. Αυτό το token θα χρησιμοποιείται σε επόμενα αιτήματα.

Σημείο Ελέγχου: Επιβεβαίωση user role (Student/Teacher), ρύθμιση session cookies (httpOnly, secure flag), rate limiting για brute force.

Ανέβασμα Αρχείου (Upload):

Ο Teacher (ή Student, π.χ. για εργασία) επιλέγει αρχείο και στέλνει POST αίτημα (multipart form data) στο endpoint POST /upload.

Ο Web Server ελέγχει αν ο χρήστης έχει δικαιώματα να ανεβάσει (π.χ. Teacher role) και εφόσον είναι έγκυρος, αποθηκεύει το αρχείο στο File System (συνήθως σε φάκελο /var/www/courses/ κ.λπ.).

Καταχωρεί στη Database (π.χ. πίνακας files) μεταδεδομένα (όνομα μαθήματος, ποιος χρήστης το ανέβασε, όνομα αρχείου).

Επιστρέφει επιτυχία στο browser (JSON ή redirect) και εμφανίζει μήνυμα «Το αρχείο ανέβηκε επιτυχώς».

Σημείο Ελέγχου: Input validation (π.χ. περιορισμός μεγέθους αρχείου), έλεγχος τύπου αρχείου, antivirus scan (πιθανώς), proper permissions στο filesystem.

Κατέβασμα Αρχείου (Download):

Ο Student/Teacher κάνει κλικ σε σύνδεσμο (π.χ. /files/12345) για να κατεβάσει ένα συγκεκριμένο αρχείο.

Ο Web Server λαμβάνει το αίτημα και ελέγχει ποιος ζητάει το αρχείο (session token, role).

Αν ο χρήστης έχει δικαίωμα, ο server διαβάζει το αντίστοιχο αρχείο από το File System.

Ο Server επιστρέφει το αρχείο μέσω HTTPS στον browser.

Σημείο Ελέγχου: Authorization checks (μήπως ένας Student προσπαθεί να κατεβάσει αρχείο άλλου μαθήματος/φοιτητή;), σωστό content-type κ.λπ.

Password Reset:

Χρήστης που ξέχασε κωδικό μπαίνει στο /forgot\_password. Στέλνει email/username.

Ο Web Server βρίσκει τον λογαριασμό στη DB, δημιουργεί ένα reset token και το στέλνει με Email (Exit Point) στον χρήστη.

Ο χρήστης κάνει κλικ στον σύνδεσμο (παράμετρο token=), ο server ελέγχει αν είναι έγκυρο + δεν έχει λήξει.

Ο χρήστης επιλέγει νέο κωδικό, ο server κάνει UPDATE στον πίνακα users (πεδίο hashed\_password).

Σημείο Ελέγχου: Προστασία του token (cryptographically secure), λήξη token μετά από Χ ώρα, hashing νέου password, μη διαρροή email σε logs κ.ο.κ.

## 2.2 Step 2: Determine Threats

Εφόσον έχουμε προσδιορίσει τα στοιχεία της εφαρμογής μας, μπορούμε να προχωρήσουμε στον προσδιορισμό των απειλών.

Το μοντέλο που θα χρησιμοποιήσουμε για τον εντοπισμό και την κατηγοριοποίηση των απειλών είναι το STRIDE ( Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privileges), το οποίο μας το προσφέρει και το Microsoft threat modeling tool.

STRIDE Threat List:

**Spoofing:** Threat action aimed at accessing and using another user’s credentials, such as username and password. (Authentication Threats)

**Tampering:** Threat action intending to maliciously change or modify persistent data, such as records in a database, and the alteration of data in transit between two computers over an open network, such as the Internet. (Integrity Threats)

**Repudiation:** Threat action aimed at performing prohibited operations in a system that lacks the ability to trace the operations. (Non-Repudiation Threats)

**Information disclosure:** Threat action intending to read a file that one was not granted access to, or to read data in transit. (Confidentiality Threats)

**Denial of service:** Threat action attempting to deny access to valid users, such as by making a web server temporarily unavailable or unusable. (Availability Threats)

**Elevation of privilege**: Threat action intending to gain privileged access to resources to gain unauthorized access to information or to compromise a system. (Authorization Threats)

Πηγαίνοντας στο analysis view παίρνουμε τις παρακάτω πληροφορίες.

A screenshot of a computer

AI-generated content may be incorrect.

Η παραπάνω ανάλυση μας παρουσιάζει όλες τις πιθανές απειλές με βάση τη σχεδίαση που έχουμε υλοποιήσει και μας τις κατηγοριοποιεί με βάση το STRIDE model.

Πιο συγκεκριμένα έχουμε 43 απειλές, όπου:

**S**poofing: 8

**T**ampering: 9

**R**epudiation: 4

**I**nformation Disclosure: 2

**D**enial Of Service: 8

**E**levation Of Privilege: 12

Στη συνέχεια θα αξιολογήσουμε τις απειλές με το DREAD(Damage Potential, Reproducibility, Exploitability, Affected Users, Discoverability) μοντέλο το οποίο χρησιμοποιείται για ranking threats και αποδίδει και μια αριθμητική τιμή.

Για κάθε κατηγορία έχει βαθμολογία 0 έως 10, οπότε μέγιστη βαθμολογία τα 50.

Πιο συγκεκριμένα αναφέρεται στις παρακάτω κατηγορίες:

* Damage: The total damage (or impact) that a threat can cause.
* Reproducibility: The ease at which an attack can occur (or be replicated).
* Exploitability: How likely or easily the weakness or threat can be exploited.
* Affected Users: The number of (end) users that could be affected by a threat being exploited.
* Discoverability: How likely a threat will be discovered by an attacker.

Οι βαθμολογίες που αποδίδονται ανά περίπτωση σε κάθε κατηγορία είναι οι παρακάτω:

**DREAD Category: Damage**

Damage relates to potential damage (or impact) a threat could cause if it is exploited by an attacker.

The ratings consist of:

* 0, which is no damage
* 5, which is information disclosure
* 8, which is non-sensitive data of individuals being compromised
* 9, which is non-sensitive administrative data being compromised
* 10, which is destruction of the system in scope, the data, or loss of system availability

**DREAD Category: Reproducibility**

Reproducibility relates to how easily (or likely) an attack can be replicated.

The ratings consist of:

* 0, which is nearly impossible or difficult
* 5, which is complex
* 7.5, which is easy
* 10, which is very easy

**DREAD Category: Exploitability**

Exploitability relates to how easily or likely a weakness or threat can be exploited.

The ratings consist of:

* 2.5, which requires advanced technical skills
* 5, which requires tools that are available
* 9, which requires application proxies
* 10, which (only) requires browser

**DREAD Category: Affected Users**

Affected Users relates to the number of (end) users that could be affected by a threat being exploited.

The ratings consist of:

* 0, which is no users are affected
* 2.5, which is only individual users are affected
* 6, which is few users are affected
* 8, which is administrative users are affected (these are more important users)
* 10, which is all users are affected

**DREAD Category: Discoverability**

Discoverability relates to how likely a threat will be discovered by an attacker.

The ratings consist of:

* 0, which is hard to discover
* 5, which is open requests can discover the threat
* 8, which is a threat being publicly known or found
* 10, which is the threat is easily discoverable, such as in an easily accessible page or form

**Overall DREAD Rating**

Low for overall ratings between 1-10

Medium for overall ratings between 11-24

High for overall ratings between 25-39

Critical for overall ratings between 40-50

**OVERALL DREAD SCORE == Damage + Reproducibility + Exploitability + Affected Users + Discoverability)**

Προχωράμε σε ανάλυση των κυριότερων απειλών για την εφαρμογή μας.

**Threat Ranking with DREAD**

**Spoofing category: An attacker can impersonate a user. (Authentication Threats)**

**Student User Spoofing:**

Damage= 8, which is non-sensitive data of individuals being compromised

Reproducibility=7.5, which is easy

Exploitability=5, which requires tools that are available

Affected Users=2.5, which is only individual are affected

Discoverability=10, which is the threat is easily discoverable, such as in an easily accessible page or form

**OVERALL DREAD SCORE = 8+7.5+5+2.5+10=33= high**

**Teacher User Spoofing:**

Damage= 9, which is non-sensitive administrative data of individuals being compromised

Reproducibility=7.5, which is easy

Exploitability=5, which requires tools that are available

Affected Users=10, which is all users are affected

Discoverability=10, which is the threat is easily discoverable, such as in an easily accessible page or form

**OVERALL DREAD SCORE = 9+7.5+5+10+10=41.5= critical**

Σημείωση:

Η επίδραση του teacher user είναι μεγαλύτερη λόγω δυνατοτήτων όπως η επεξεργασία βαθμών και εργασιών και γενικότερα αρχείων της σελίδας , ενέργειες που μπορούν να επηρεάσουν πολλαπλούς χρήστες.

**Tampering Category: An attacker can change/modify data. (Integrity Threats)**

**SQL Injection:**

Damage= 10, which is destruction of the system in scope, the data, or loss of system availability

Reproducibility=7.5, which is easy

Exploitability=5, which requires tools that are available

Affected Users=10, which is all users are affected

Discoverability=8, which is a threat being publicly known or found

**OVERALL DREAD SCORE = 10+7.5+5+10+8=40.5= critical**

**Repudiation Category: the ability to trace actions (Non-Repudiation Threats)**

**Web Server claims that it did not receive data from a source outside the trust boundary:**

Damage= 10, which is destruction of the system in scope, the data, or loss of system availability

Reproducibility=5, which is complex

Exploitability=5, which requires tools that are available

Affected Users=6, which is few users are affected

Discoverability=8, which is a threat being publicly known or found

**OVERALL DREAD SCORE = 10+5 +5+6+8=34= high**

**Information Disclosure Category: Reading data without having access (Confidentiality Threats)**

**Authorization Bypass of the database:**

Damage= 10, which is destruction of the system in scope, the data, or loss of system availability

Reproducibility=7.5, which is easy

Exploitability=5, which requires tools that are available

Affected Users=10, which is all users are affected

Discoverability=8, which is a threat being publicly known or found

**OVERALL DREAD SCORE = 10+7.5+5+10+8=40.5= critical**

**Authorization Bypass of the file system:**

Damage= 5, which information disclosure

Reproducibility=7.5, which is easy

Exploitability=5, which requires tools that are available

Affected Users=10, which is all users are affected

Discoverability=8, which is a threat being publicly known or found

**OVERALL DREAD SCORE = 5+7.5+5+10+8=35.5= high**

\*Σημείωση, Η βάση δεδομένων κρατάει πιο ευαίσθητο περιεχόμενο, από ότι το file system, ωστόσο και η παραβίασή του μπορεί να οδηγήσει σε σοβαρά προβλήματα όπως η ανάκτηση πληροφοριών σχετικά με το διδακτικό υλικό και η αντιγραφή εργασιών.

**Denial Of Service Category: making a resource unavailable/unusable (Availability Threats)**

**Potential Process crash on the Web Client / Web Server:**

Damage= 10, which is destruction of the system in scope, the data, or loss of system availability

Reproducibility=7.5, which is easy

Exploitability=5, which requires tools that are available

Affected Users=10, which is all users are affected

Discoverability=8, which is a threat being publicly known or found

**OVERALL DREAD SCORE = 10+7.5+5+10+8=40.5= critical**

**Elevation Of Privilege Category: gaining unauthorized access (Authorization Threats)**

**Elevation by Changing the Execution Flow on the Web Client /Web Server:**

Damage= 10, which is destruction of the system in scope, the data, or loss of system availability

Reproducibility=7.5, which is easy

Exploitability=5, which requires tools that are available

Affected Users=10, which is all users are affected

Discoverability=8, which is a threat being publicly known or found

**OVERALL DREAD SCORE = 10+7.5+5+10+8=40.5= critical**

**DREAD SUMMARY:**

**Spoofing: Student = 33 = high**

**Spoofing: Teacher = 41 = critical**

**Tampering: SQLi = 40.5 = critical**

**Repudiation: Web Server claims that it did not receive data from a source outside the trust boundary =34 high**

**Information Disclosure: Authorization Bypass of the database: 40.5 =critical**

**Information Disclosure: Authorization Bypass of the filesystem: 35.5 =high**

**Denial Of Service: Potential Process crash on the Web Client / Web Server: 40.5=critical**

**Elevation Of Privilege: Elevation by Changing the Execution Flow on the Web Client /Web Server: 40.5 =critical**

## 2.3 Step 3: Determine Countermeasures and Mitigation

Σε αυτό το βήμα θα αποφασίσουμε για τα μέτρα προστασίας που θα εφαρμοστούν για να μπορέσουμε να βελτιώσουμε το security posture της εφαρμογής μας, αντιμετωπίζοντας τις κυριότερες απειλές.

Για την επιλογή των μέτρων προστασίας θα ακολουθήσουμε το πρότυπο OWASP ASVS ( Application Security Verification Standard).

To ASVS έχει 3 επίπεδα εφαρμογών:

Επίπεδο 1(Opportunistic): αποτελεί το ελάχιστο security που απαιτείται για μια εφαρμογή. Αφορά μικρές εφαρμογές/επιχειρήσεις. Mostly automated reviews

Επίπεδο 2(Standard): Καλό για εφαρμογές που περιέχουν ευαίσθητά δεδομένα. Έχει πιο αυστηρό authentication και manual code reviews.

Επίπεδο 3(Advanced): Έχει πολύ αυστηρό security που απαιτεί κρυπτογραφία, εκτενείς μηχανισμούς ασφαλείας και διαρκείς ελέγχους. Αφορά εφαρμογές μεγάλου βεληνεκούς σε τομείς όπως η τράπεζες και η υγεία.

Η δική μας εφαρμογή μπορεί να συνδυάζει τα πρώτα 2 επίπεδα.

Επιπλέον τα μέτρα προστασίας που προτείνονται χωρίζονται σε 14 κατηγορίες.

**Προχωράμε, λοιπόν στην εφαρμογή μέτρων προστασίας με βάση το ASVS του OWASP:**

**Αντιμετώπιση Spoofing Attacks:**

V2: Authentication Verification Requirements

V2.1 Password Security Requirements

Το οποίο προτείνει password strength requirements, μέτρα όπως MFA για να δυσκολέψει τον επιτιθέμενο, device fingerprinting, CAPTCHA για να αποφύγουμε automated/bot attacks.

**Αντιμετώπιση Tampering Attacks:**

V5.3 Output encoding and Injection Prevention Requirements

SQL Injection Prevention:

Προτείνει input validation, prepared statements και γενικά συμβουλές στο development ώστε να γίνεται καλή διαχείριση των εντολών που παίρνουμε από τους χρήστες.

Άλλες λύσεις είναι least privilege, οπότε πρέπον access ο user, και χρήση WAF.

5.3.5 Verify that where parameterized or safer mechanisms are not present, context-specific output encoding is used to protect against injection attacks, such as the use of SQL escaping to protect against SQL injection. (C3, C4)

**Αντιμετώπιση Repudiation Attacks:**

V7: Error Handling and Logging Verification Requirements

V7.1 Log Content Requirements

V7.2 Log Processing Requirements

V7.3 Log Protection Requirements

Το οποίο παρέχει συμβουλές development σχετικά με ένα σωστό logging σύστημα, αναφέροντας ποια δεδομένα πρέπει να περιέχει και ποια όχι π.χ. προσωπικά δεδομένα καθώς και restricted-access σε αυτά και να γράφονται μια φορά μόνο και να χρησιμοποιείται hashing.

**Αντιμετώπιση Information Disclosure Attacks:**

V4: Access Control Verification Requirements

Προτείνει μέτρα για να κάνει harden ένα access control system, όπως τα

Least Privilege, deny default policy if it doesn’t match the rules for a user.

V1.8 Data Protection and Privacy Architecture

1.8.2 Verify that all protection levels have an associated set of protection requirements, such as encryption requirements, integrity requirements, retention, privacy and other confidentiality requirements, and that these are applied in architecture.

**Αντιμετώπιση Denial Of Service Attacks:**

V2.2 General Authenticator Security

2.2.1 Verify that anti-automation controls are effective at mitigating breached credential testing, brute force, and account lockout attacks. Such controls include blocking the most common breached passwords, soft lockouts, rate limiting, CAPTCHA, ever increasing delays between attempts, IP address restrictions, or risk-based restrictions such as location, first login on a device, recent attempts to unlock the account, or similar. Verify that no more than 100 failed attempts per hour is possible on a single account.

V11.1 Business Logic Security

11.1.4 Verify that the application has anti-automation controls to protect against excessive calls such as mass data exfiltration, business logic requests, file uploads or denial of service attacks.

V12.1 File Upload

12.1.1 Verify that the application will not accept large files that could fill up storage or cause a denial of service.

**Αντιμετώπιση Elevation Of Privilege Attacks:**

V1.2 Authentication Architecture

1.2.2 Verify that communications between application components, including APIs, middleware and data layers, are authenticated. Components should have the least necessary privileges needed.

V4.1 General Access Control Design

4.1.3 Verify that the principle of least privilege exists - users should only be able to access functions, data files, URLs, controllers, services, and other resources, for which they possess specific authorization. This implies protection against spoofing and elevation of privilege.

Με τις παραπάνω ενέργειες αντιμετωπίζουμε τις κυριότερες απειλές που αφορούν την εφαρμογή τους.

Προχωράμε στην εφαρμογή τους στο Microsoft Threat Modelling Tool για να δούμε την ανανεωμένη εικόνα της εφαρμογής μας.

Καταλήγουμε σε mitigation 11 από τις 38 απειλές που προτείνει το Microsoft Threat Modelling Tool.

## 2.4 Step 4: Assess your work

Το βήμα στο οποίο αξιολογούμε τις διαδικασίες που ακολουθήσαμε και τα αποτελέσματα που εξαγάγαμε και σε ποια αρχεία καταλήξαμε (DFDs, Reports).

**Σύντομη Ανακεφαλαίωση:**

Το σύστημα υπό ανάλυση ήταν μια web εφαρμογή e-learning πλατφόρμα η οποία έχει 2 χρήστες ( Students, Teachers), Web Client, Web Server, Database και File System.

Υπάρχουν 2 Security Boundaries, 1 μεταξύ users – Web Client και 1 που περιέχει Web Server, Database και File System.

Η επικοινωνία είναι μέσω https , ο web client είναι υπεύθυνος για authentication και να μεταφέρει τα αιτήματα στο web server.

Ο Web Server είναι υπεύθυνος για authorization και όλο το application logic, βρίσκεται στο ίδιο δίκτυο με Database και File System και υπάρχει secure επικοινωνία μεταξύ τους.

Στη συνέχεια κατασκευάσαμε ένα DFD με βάση τα παραπάνω και προχωρήσαμε στην ανάλυση των απειλών με το STRIDE μοντέλο και εξαγάγαμε ένα report που δίνει πιο αναλυτικά τη συνολική εικόνα των απειλών στην εφαρμογή μας.

Στη συνέχεια αξιολογήσαμε τις κυριότερες απειλές με το μοντέλο DREAD που μας έδωσε μια αριθμητική απόδοση των απειλών αυτών.

Τέλος, προχωρήσαμε σε εύρεση αντίμετρων μέσω του OWASP ASVS και κάναμε mitigation στις κυριότερες απειλές και εξαγάγαμε ένα report που δίνει την εικόνα της εφαρμογής μας μετά την αντιμετώπιση των συγκεκριμένων απειλών.

# 3. Συμπεράσματα:

Σκοπός της παραπάνω εργασίας ήταν η υλοποίηση ενός Application Threat Modelling για μια web εφαρμογή ακολουθώντας τα βήματα τις μεθοδολογίας του OWASP Threat Modelling Process. Χρησιμοποιήθηκε το εργαλείο Microsoft Threat Modelling Tool 2016 με το οποίο αναλύσαμε την εφαρμογή σε περαιτέρω οντότητες και δημιουργήσαμε Data Flow Diagrams. Εφόσον είδαμε την εφαρμογή από σχεδιαστική άποψη βάζοντας περιορισμούς/ παραδοχές σε κάθε οντότητα, με βάση τον τρόπο υλοποίησής μας, προχωρήσαμε στην αναλυτική πλευρά της εφαρμογής μας, καθώς το εργαλείο μας αναλύει τις απειλές με βάση το σχεδιασμό που έχουμε πραγματοποιήσει. Σε κάθε φάση του σχεδιασμού είχαμε τη δυνατότητα του να εξάγουμε ένα report από το εργαλείο Microsoft Threat Modelling Tool 2016 για να βλέπουμε πιο αναλυτικά την κατάσταση της εφαρμογής μας.

Οι απειλές κατηγοριοποιούνται από το εργαλείο με βάση το μοντέλο STRIDE, οπότε γνωρίσαμε και ένα τρόπο κατηγοριοποίησής των απειλών.

Στη συνέχεια αξιολογήσαμε τις απειλές με το DREAD μοντέλο το οποίο αποδίδει αριθμητική τιμή σε κάθε απειλή οπότε είναι μια χρήσιμη μεθοδολογία για βαθμολόγηση και σύγκριση των απειλών.

Τέλος προτείναμε αντίμετρα με βάση το OWASP ASVS το οποίο προτείνει security controls και secure development requirements, και κάναμε mitigate τις κυριότερες απειλές.

Όλες αυτές οι διαδικασίες γίνονται χωρίς κώδικα και είναι εξαιρετικά χρήσιμες σε πολλές περιπτώσεις, για παράδειγμα ως προεργασία penetration testing ή και ως μέτρο επικοινωνίας με το developer team σε περίπτωση δημιουργίας εφαρμογής.

# 4. Πηγές:

[1] <https://owasp.org/www-community/Threat_Modeling>

[2] <https://owasp.org/www-community/Threat_Modeling_Process>

[3] <https://en.wikipedia.org/wiki/STRIDE_model>

[4] <https://threat-modeling.com/dread-threat-modeling/>

[5] <https://en.wikipedia.org/wiki/DREAD_(risk_assessment_model)>

[6] <https://owasp.org/www-project-application-security-verification-standard/>

[7] <https://cheatsheetseries.owasp.org/index.html>

[8] https://github.com/OWASP/ASVS/tree/v4.0.3