**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**  **e-learning platform** |
| **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** 4](#_Toc191996255)

[1.1 Σκοπός του Εγγράφου 4](#_Toc191996256)

[1.2 Πεδίο Εφαρμογής 4](#_Toc191996257)

[1.3 Μεθοδολογία 4](#_Toc191996258)

[**2. Βήματα OWASP Application Threat Modeling** 5](#_Toc191996259)

[*2.1 Step1: Decompose the Application* 5](#_Toc191996260)

[2.1.1 Threat Model Information: 5](#_Toc191996261)

[2.1.2 External Dependencies: 5](#_Toc191996262)

[2.1.3 Entry Points: 5](#_Toc191996263)

[2.1.4 Assets: 6](#_Toc191996264)

[2.1.5 Trust Levels: 6](#_Toc191996265)

[2.1.6 Exit Points: 6](#_Toc191996266)

[2.1.7 Entities: 7](#_Toc191996267)

[2.1.8 Data Flow Diagrams: 8](#_Toc191996268)

[2.1.9 Data Flows Analysis 9](#_Toc191996269)

[*2.2 Step 2: Determine and Rank Threats* 10](#_Toc191996270)

[2.2.1 Threat Categorization 10](#_Toc191996271)

[2.2.2 Threat Identification 10](#_Toc191996272)

[2.2.3 Threat Ranking 11](#_Toc191996273)

[2.2.4 Ενδεικτικά Παραδείγματα Βαθμολόγησης DREAD 12](#_Toc191996274)

[2.2.5 Threat Ranking with DREAD 13](#_Toc191996275)

[*2.3 Step 3: Determine Countermeasures and Mitigation* 14](#_Toc191996276)

[2.3.1 Countermeasures by OWASP ASVS 14](#_Toc191996277)

[2.3.2 Mitigation Strategy 15](#_Toc191996278)

[**3. Συμπεράσματα** 17](#_Toc191996279)

[**4. Πηγές** 18](#_Toc191996280)

# **1. Εισαγωγή**

Σε αυτό το έγγραφο πραγματοποιείται Application Threat Modeling για μια e-learning Web εφαρμογή (π.χ. “Open eClass”). Ακολουθούνται κυρίως οι 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 του 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:

1. Student (πρόσβαση σε υλικό, βαθμολογίες, ανέβασμα εργασιών).
2. 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: Decompose the Application**

* Αναγνώριση περιουσιακών στοιχείων, ροών δεδομένων, ορίων εμπιστοσύνης και επιφανειών επίθεσης.

**Βήμα 2: Determine and Rank Threats**

* Χρήση μιας δομημένης προσέγγισης για την ανακάλυψη ευπαθειών και την κατηγοριοποίησή τους με το μοντέλο STRIDE.
* Αξιολόγηση των κινδύνων με το μοντέλο DREAD.

**Βήμα 3: Determine Countermeasures and Mitigations**

* Ορισμός μέτρων ασφαλείας και βέλτιστων πρακτικών για τη μείωση των κινδύνων, ακολουθώντας το πρότυπο OWASP ASVS.
* Ορισμός πλάνου και χρονοδιαγράμματος αντιμετώπισης των κινδύνων.

# **2. Βήματα OWASP Application Threat Modeling**

## *2.1 Step1: Decompose the Application*

### 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.

### 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, βαθμολογίες. | |
| 6 | File Storage / File System | Χώρος αποθήκευσης των αρχείων (pdfs, docs, zip) που ανεβάζουν εκπαιδευτικοί ή φοιτητές (σημειώσεις, εργασίες κ.λπ.). | |
| 7 | 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. |

### A diagram of a diagram AI-generated content may be incorrect.A diagram of a computer system AI-generated content may be incorrect.2.1.8 Data Flow Diagrams:

Εικόνα 2: Ενδεικτικό Data flow Diagram της λειτουργίας file upload, εφόσον έχουμε επιτυχημένη σύνδεση

Εικόνα 1: High-level Data flow diagram

### 2.1.9 Data Flows Analysis

|  |  |  |
| --- | --- | --- |
| Login / Authentication | User (Student/Teacher) ανοίγει το login page. | Σημείο Ελέγχου: |
| Πληκτρολογεί username/password → γίνεται HTTPS POST αίτημα στον Web Server. | Επιβεβαίωση user role  (Student/Teacher), ρύθμιση session cookies (httpOnly, secure flag), rate limiting για brute force. |
| 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. |
| Ανέβασμα Αρχείου (Upload) | Ο Teacher (ή Student, π.χ. για εργασία) επιλέγει αρχείο και στέλνει POST αίτημα (multipart form data) στο endpoint POST /upload. | Input validation (π.χ. περιορισμός μεγέθους αρχείου), έλεγχος τύπου αρχείου, antivirus scan (πιθανώς), proper permissions στο filesystem. |
| 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». |
| Κατέβασμα Αρχείου (Download) | Ο Student/Teacher κάνει κλικ σε σύνδεσμο (π.χ. /files/12345) για να κατεβάσει ένα συγκεκριμένο αρχείο. | Authorization checks (μήπως ένας Student προσπαθεί να κατεβάσει αρχείο άλλου μαθήματος/φοιτητή;), σωστό content-type κ.λπ. |
| 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. |
| Password Reset | Χρήστης που ξέχασε κωδικό μπαίνει στο /forgot\_password. Στέλνει email/username. | Προστασία του token (cryptographically secure), λήξη token μετά από Χ ώρα, hashing νέου password, μη διαρροή email σε logs κ.ο.κ. |
| 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). |

## *2.2 Step 2: Determine and Rank Threats*

### 2.2.1 Threat Categorization

Για τον εντοπισμό και την κατηγοριοποίηση των απειλών χρησιμοποιείται το μοντέλο STRIDE ( Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privileges)

**STRIDE Threat List:**

**Spoofing:** Δράση απειλής με στόχο την πρόσβαση και τη χρήση των διαπιστευτηρίων ενός άλλου χρήστη, όπως το όνομα χρήστη και ο κωδικός πρόσβασης. (Authentication Threats)

**Tampering:** Δράση απειλής που αποσκοπεί στην κακόβουλη αλλαγή ή τροποποίηση μόνιμων δεδομένων, όπως εγγραφές σε μια βάση δεδομένων, και η αλλοίωση δεδομένων κατά τη διαμετακόμιση μεταξύ δύο υπολογιστών μέσω ανοικτού δικτύου, όπως το Διαδίκτυο. (Integrity Threats)

**Repudiation:** Δράση απειλής που αποσκοπεί στην εκτέλεση απαγορευμένων λειτουργιών σε ένα σύστημα που δεν έχει τη δυνατότητα να τις εντοπίσει. (Non-Repudiation Threats)

**Information Disclosure:** ενέργεια απειλής που αποσκοπεί στην ανάγνωση αρχείου στο οποίο δεν έχει χορηγηθεί πρόσβαση ή στην ανάγνωση δεδομένων κατά τη μεταφορά. (Confidentiality Threats)

**Denial of service:** Δράση απειλής που επιχειρεί να αρνηθεί την πρόσβαση σε έγκυρους χρήστες, π.χ. καθιστώντας προσωρινά μη διαθέσιμο ή άχρηστο έναν διακομιστή ιστού. (Availability Threats)

**Elevation of privilege:** Δράση απειλής που αποσκοπεί στην απόκτηση προνομιακής πρόσβασης σε πόρους για την απόκτηση μη εξουσιοδοτημένης πρόσβασης σε πληροφορίες ή για την παραβίαση ενός συστήματος. (Authorization Threats)

### 2.2.2 Threat Identification

A screenshot of a computer

AI-generated content may be incorrect.Μέσω του analysis view του Microsoft Threat Modelling Tool παρατηρούνται τα παρακάτω

Εικόνα 3: Data Analysis view

Πιο συγκεκριμένα παρατηρούνται 43 απειλές και πιο συγκεκριμένα ανά κατηγορία:

Spoofing: 8

Tampering: 9

Repudiation: 4

Information Disclosure: 2

Denial Of Service: 8

Elevation Of Privilege: 12

STRIDE: Αναλυτική Παρουσίαση Απειλών

|  |  |  |
| --- | --- | --- |
| Απειλή | Περιγραφή | Συνέπειες |
| Spoofing (Πλαστοπροσωπία):  Spoofing the Human User External Entity π.χ. Teacher User may be spoofed by an attacker to lead to unauthorized access. | Ένας κακόβουλος χρήστης (π.χ. φοιτητής) προσπαθεί να αποκτήσει τα διαπιστευτήρια (username/password) ή το session token κάποιου καθηγητή. Με αυτόν τον τρόπο αποκτά δικαιώματα “Teacher” και μπορεί να τροποποιήσει βαθμολογίες, να ανεβάσει/διαγράψει υλικό κ.λπ. | Παραβίαση εμπιστευτικότητας (δείτε δεδομένα μαθητών), ακεραιότητας (αλλοίωση βαθμολογιών/υλικού). |
|  |
|  |
| Tampering (Αλλοίωση/Παραποίηση Δεδομένων):  Potential SQL Injection Vulnerability for SQL Database. | Μέσω φόρμας (π.χ. αναζήτηση μαθήματος ή login), ο επιτιθέμενος στέλνει ειδικά crafted SQL queries. Αν δεν γίνεται proper parameterization, μπορεί να τροποποιήσει/διαγράψει εγγραφές (π.χ. βαθμολογίες). | Καταστροφή ή αλλοίωση βαθμολογιών, παραβίαση εμπιστευτικότητας (π.χ. άντληση λιστών χρηστών). |  |
| Repudiation (Άρνηση Πράξεων):  Potential Data Repudiation by Web Server / Database/ File System π.χ. Web Server claims that it did not receive data from a source outside the trust boundary. | Το σύστημα δεν κρατά επαρκή logs για αλλαγές βαθμολογιών, ανεβάσματα αρχείων κ.λπ. Κάποιος μπορεί να αλλάξει δεδομένα ή να αφαιρέσει υλικό και μετά να ισχυριστεί ότι “δεν το έκανε”, χωρίς δυνατότητα εντοπισμού. | Αδυναμία απόδοσης ευθύνης, δεν μπορούμε να αποδείξουμε πότε και από ποιον έγιναν οι κρίσιμες ενέργειες. |  |
| Information Disclosure (Διαρροή Πληροφοριών):  Weak access control of a resource π.χ. Improper data protection of SQL Database or File System can allow an attacker to read information not intended for disclosure. | Ένας φοιτητής αποκτά πρόσβαση σε αρχεία που ανήκουν σε άλλο φοιτητή ή σε υλικό που δεν θα έπρεπε να δει (π.χ. λύσεις ασκήσεων, βαθμοί). | Παραβίαση εμπιστευτικότητας, πιθανή διαρροή προσωπικών δεδομένων (GDPR issues). |  |
| Denial of Service (Άρνηση Εξυπηρέτησης):  Potential Excessive Resource Consumption for Web Server or File System or Database. Π.χ. Large File Upload Flood | Ο επιτιθέμενος ανεβάζει πολλαπλά τεράστια αρχεία (π.χ. αρχεία-«σκουπίδια» gigabytes) γονατίζοντας το χώρο αποθήκευσης ή τον ίδιο τον server. | Το σύστημα γίνεται μη διαθέσιμο (γεμάτος δίσκος, κρεμάει ο server). |  |
| Elevation of Privilege (Αναβάθμιση Προνομίων):  Elevation Using Impersonation π.χ Browser Client may be able to impersonate the context of a Teacher User to gain additional privileges Student or Anonymous Web User → Teacher Escalation | Μέσω κάποιας ευπάθειας (Insecure Direct Object References, ελλιπής έλεγχος ρόλου), ένας απλός φοιτητής ή κάποιος ανώνυμος χρήστης αποκτά πρόσβαση σε λειτουργίες του καθηγητή (π.χ. μεταβολή βαθμολογιών). | Παραβίαση ακεραιότητας δεδομένων, διάλυση αξιοπιστίας του συστήματος. |  |

### 2.2.3 Threat Ranking

Η αξιολόγηση των απειλών πραγματοποιείται με το μοντέλο DREAD (Damage Potential, Reproducibility, Exploitability, Affected Users, Discoverability)

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.

### 2.2.4 Ενδεικτικά Παραδείγματα Βαθμολόγησης DREAD

|  |  |
| --- | --- |
| Damage Potential (D) | • 0: Καμία ζημιά (π.χ. ανούσια ή εικονική επίθεση). • 5: Απώλεια/confidential data με μέτριο αντίκτυπο (π.χ. μη ευαίσθητα δεδομένα). • 8: Παραβίαση προσωπικών δεδομένων (non-critical personal data). • 9: Παραβίαση ευαίσθητων ή διοικητικών δεδομένων. • 10: Ολοσχερής καταστροφή του συστήματος ή απώλεια πολύ ευαίσθητων δεδομένων (π.χ. κρυφές πληροφορίες, οικονομικά, GDPR κ.λπ.) |
| Reproducibility (R) | • 0: Σχεδόν αδύνατο να αναπαραχθεί (χρειάζεται πολύ εξεζητημένο περιβάλλον). • 5: Απαιτείται κάποια πολυπλοκότητα (συνδυασμός εργαλείων, συνθήκες). • 7.5: Σχετικά εύκολο, υπάρχουν διαθέσιμα εργαλεία (script, scanners). • 10: Πολύ εύκολο/απλό, οποιοσδήποτε μπορεί να το κάνει με ελάχιστες γνώσεις (π.χ. μια απλή φόρμα επιτρέπει injection). |
| Exploitability (E) | • 2.5: Χρειάζεται εξειδικευμένες γνώσεις και τεχνικές ικανότητες (π.χ. reverse engineering). • 5: Υπάρχουν public εργαλεία/exploits, αλλά απαιτούνται κάποιοι βασικοί χειρισμοί. • 9: Υπάρχουν ευρέως διαθέσιμα scripts ή tutorials, εύκολα στη χρήση. • 10: Μπορεί να γίνει ακόμα και από browser χωρίς επιπλέον εργαλεία (π.χ. copy-paste σε URL). |
| Affected Users (A) | • 0: Κανένας χρήστης δεν επηρεάζεται (θεωρητικό exploit). • 2.5: Μόνο ένας χρήστης (στοχευμένα). • 6: Μια μικρή ομάδα χρηστών (subset). • 8: Αφορά πιο κρίσιμους χρήστες (admin, teacher), ή αρκετούς «υψηλής αξίας» λογαριασμούς. • 10: Όλοι οι χρήστες ή όλη η υποδομή μπορούν να επηρεαστούν. |
| Discoverability (D’) | • 0: Εξαιρετικά δύσκολη ή άγνωστη ευπάθεια (πρέπει να είσαι “insider”). • 5: Χρειάζεται ψάξιμο, αλλά όχι αδύνατον· με δοκιμές ή ειδικά εργαλεία. • 8: Η ευπάθεια είναι γνωστή (public) ή υπάρχει POC/αναφορά ήδη στο internet. • 10: Προφανής ευπάθεια, πολύ εμφανής (π.χ. ένα πεδίο που δεν έχει καν input validation). |

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

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

### 2.2.5 Threat Ranking with DREAD

|  |  |  |
| --- | --- | --- |
| Spoofing: Teacher Spoofing | • Damage (D): 9 (μπορεί να τροποποιήσει βαθμολογίες/υλικό) • Reproducibility (R): 7.5 (αρκετά εφικτό με brute force, κλοπή session κ.λπ.) • Exploitability (E): 5 (χρειάζεται κάποια εργαλεία/γνώσεις) • Affected Users (A): 10 (επηρεάζει όλα τα μαθήματα/φοιτητές, γιατί Teacher έχει ευρύτερη πρόσβαση) • Discoverability (D’): 10 (καθόλου κρυφό, εύκολο να δοκιμάσει κάποιος brute force / session hijack) | Σύνολο: 9 + 7.5 + 5 + 10 + 10 = 41.5  Σοβαρότητα: Critical |
| Tampering: SQL Injection | • Damage (D): 10 (μπορεί να καταστρέψει/διαρρεύσει ολόκληρη τη βάση) • Reproducibility (R): 7.5 (ευρέως γνωστές τεχνικές, αρκετά “standard” exploit) • Exploitability (E): 5 (χρειάζεται κάποια γνώση, αλλά υπάρχουν πολλά tools) • Affected Users (A): 10 (πρακτικά όλοι οι χρήστες επηρεάζονται, καθώς η DB περιέχει τα πάντα) • Discoverability (D’): 8 (συνήθως υπάρχουν scanners και δημόσιες λίστες injection payloads) | Σύνολο: 10 + 7.5 + 5 + 10 + 8 = 40.5  **Σοβαρότητα: Critical** |
| Repudiation: Insufficient Logging | • Damage (D): 10 (η επίθεση μπορεί να καταστρέψει/αλλοιώσει δεδομένα χωρίς ίχνος) • Reproducibility (R): 5 (χρειάζεται κάποια “αλλοίωση” ή πράξη που περνάει κάτω από τα ραντάρ) • Exploitability (E): 5 (όποιος έχει πρόσβαση στο σύστημα με κάποια μορφή εξουσίας, π.χ. Teacher με κακόβουλη πρόθεση) • Affected Users (A): 6 (μπορεί να επηρεάσει έναν αριθμό χρηστών — όχι πάντα όλους, αλλά και αρκετούς) • Discoverability (D’): 8 (δεν είναι “κρυφό” το ότι δεν υπάρχουν logs· ο επιτιθέμενος το ανακαλύπτει εύκολα) | Σύνολο: 10 + 5 + 5 + 6 + 8 = 34  **Σοβαρότητα: High** |
| Information Disclosure: Unauthorized File Access | • Damage (D): 5 (συνήθως διαρροή υλικού/βαθμών· όχι ολική καταστροφή) • Reproducibility (R): 7.5 (εφικτό με δοκιμές path, IDOR, λανθασμένα permission checks) • Exploitability (E): 5 (υπάρχουν πιθανές μέθοδοι, αλλά δεν είναι “κάνε κλικ και έγινε”) • Affected Users (A): 10 (στη χειρότερη, μπορούν να διαρρεύσουν δεδομένα όλων των φοιτητών) • Discoverability (D’): 8 (εάν δεν υπάρχει server-side authorization, εύκολα εντοπίζεται με βασικές δοκιμές) | Σύνολο: 5 + 7.5 + 5 + 10 + 8 = 35.5  **Σοβαρότητα: High** |
| Denial of Service:  Large File Upload Flood | • Damage (D): 10 (το σύστημα “κρεμάει” ή γεμίζει ο δίσκος· το e-class γίνεται μη διαθέσιμο) • Reproducibility (R): 7.5 (αρκετά απλό με scripts/automation) • Exploitability (E): 5 (χρειάζεται κάποια βασική ικανότητα, αλλά όχι κάτι τρομερό) • Affected Users (A): 10 (κανείς δεν έχει πρόσβαση στο σύστημα όταν πέφτει) • Discoverability (D’): 8 (η δυνατότητα ανεβάσματος μεγάλων αρχείων είναι εμφανής, μάλλον εύκολο να το δοκιμάσει κάποιος) | Σύνολο: 10 + 7.5 + 5 + 10 + 8 = 40.5  **Σοβαρότητα: Critical** |
| Elevation of Privilege:  Student/ Anonymous web user→ Teacher Escalation | • Damage (D): 9 (ο φοιτητής γίνεται δυνάμει “καθηγητής”, τροποποιεί μαθήματα/βαθμολογίες) • Reproducibility (R): 7.5 (συνήθως αν υπάρχει κάποιο λογικό κενό π.χ. IDOR ή λανθασμένος role check) • Exploitability (E): 5 (απαιτείται γνώση ή κάποιο exploit λογικής) • Affected Users (A): 10 (όλοι οι φοιτητές μπορεί να επηρεαστούν· π.χ. αλλοιώσεις βαθμών σε πολλά μαθήματα) • Discoverability (D’): 8 (σχεδόν όσο εμφανής όσο π.χ. μία κλασική SQLi· αλλά με προσπάθεια εντοπίζεται) | Σύνολο: 9 + 7.5 + 5 + 10 + 7 = 39.5  **Σοβαρότητα: High** |

Συνολική Εικόνα:

* Critical (≥ 40): Teacher Spoofing, SQL Injection, Large File Upload DoS
* High (25–39): Insufficient Logging, Unauthorized File Access, Teacher Escalation

## *2.3 Step 3: Determine Countermeasures and Mitigation*

### 2.3.1 Countermeasures by OWASP ASVS

Για την επιλογή των μέτρων προστασίας ακολουθείται το πρότυπο OWASP ASVS

|  |  |  |
| --- | --- | --- |
|  | OWASP ASVS Κατηγορίες | Προτεινόμενα Αντίμετρα |
| Teacher Spoofing (Spoofing) | • V2: Authentication o V2.2 General Authenticator Security o V2.4 Credential Storage • V3: Session Management o V3.2 Session Binding | **1. Strong Password Policies & Storage** o ASVS V2.4.1: Χρήση αλγορίθμων hashing (bcrypt, Argon2). o Επιβολή ελάχιστου μήκους, πολυπλοκότητας, επανάληψης check για κοινά passwords (password blacklist). **2. Brute Force Protection** o ASVS V2.2.1: Rate limiting, CAPTCHA μετά από συγκεκριμένο αριθμό αποτυχημένων προσπαθειών. o Lockout / progressive delay για επαναλαμβανόμενες αποτυχημένες συνδέσεις. **3. Secure Session Tokens** o ASVS V3.2.1: μοναδικά και τυχαία session IDs o Προαιρετικά, υλοποίηση session rotation μετά το login (ανανεώνεται το session ID). |
| SQL Injection (Tampering) | • V5: Validation, Sanitization and Encoding o V5.1 Input Validation o V5.3 (Output Encoding and Injection Prevention) | **1. Parametrized Queries / Prepared Statements** ASVS V5.3.4: Verify parameterized queries: prepared statements για DB calls. o Για κάθε εντολή SQL, να χρησιμοποιούνται prepared statements (π.χ. SELECT \* FROM users WHERE email=?) με binding παραμέτρων. o Αποφεύγουμε concatenation (π.χ. "...WHERE email='" + userInput + "'"). **2. Input Validation & Escaping** ASVS V5.1.3: Verify input validation is enforced on a trusted service layer: Όλα τα κρίσιμα checks γίνονται σε server-side. o Επιβολή ορίων (length checks), whitelist χαρακτήρων όπου εφικτό. o Ειδικά σε πεδία επικίνδυνα (π.χ. login, search), κάνουμε αυστηρό filtering πριν περάσει στην DB. |
| Insufficient Logging (Repudiation) | • V7: Error Handling and Logging  o V7.1 (Log Content) o V7.3 (Log Protection) | **1. Audit Log** ASVS: V7.1.4: Verify that each log event includes necessary information that would allow for a detailed investigation of the timeline when an event happens. o Καταγραφή κρίσιμων γεγονότων (login, logout, upload, αλλαγή βαθμολογιών). o Include userID, timestamp, IP, τι δράση έγινε. **2. Log Integrity** o ASVS V7.3.3: Προστασία logs από τροποποίηση (access μόνο από admin, hashing logs, κ.λπ.). |
| Unauthorized File Access (Information Disclosure) | • V4: Access Control Verification o V4.1 General Access Control Design o V4.2 Operation Level Access Control | **1. Server-Side Authorization Checks** o ASVS: V4.2.1 (Verify that sensitive data and APIs are protected against Insecure Direct Object Reference (IDOR) attacks targeting creation, reading, updating and deleting records, such as creating or updating someone else's record, viewing everyone's records, or deleting all records. o Σε κάθε αίτημα (π.χ. /files/123.pdf), ο server ελέγχει αν ο τρέχων χρήστης δικαιούται πρόσβαση (role, owner κ.λπ.). o Αποφυγή “direct link” που παρακάμπτει δικαιώματα. **2. Least Privilege File System Permissions** o V4.1.3 (Verify that the principle of least privilege exists) o Ο web server να μην έχει read/write δικαιώματα εκτός του απαραίτητου φακέλου. o Αν οι φοιτητές δεν δικαιούνται πρόσβαση σε κάποια αρχεία, αυτά παραμένουν εκτός “public” περιοχής. |
| Large File Upload DoS (Denial of Service) | • V12: File and Resources Verification  o V12.1 (File Upload) o V.12.2 File Integrity • V11 Business Logic  o V11.1 Business Logic Security (Έλεγχος για υπερβολικά / ύποπτα αιτήματα) | **1. Μέγιστο Επιτρεπόμενο Μέγεθος Αρχείου** o ASVS V12.1.1: Δεν δεχόμαστε μεγάλα μεγέθη αρχείων. Θέτουμε όριο μεγέθους (π.χ. 10MB). Απορρίπτουμε ή τερματίζουμε σύνδεση αν ξεπεραστεί. o Εφαρμογή quota ανά χρήστη ή ανά μάθημα. **2. Rate Limiting / Throttling** ASVS: V11.1.4: 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. o Περιορισμός ταχύτητας ανεβάσματος (π.χ. 1 request το δευτερόλεπτο) ή αριθμού uploads ανά λεπτό για τον ίδιο χρήστη/IP. o Αν παρατηρείται burst traffic, μπορεί να χρειαστεί CAPTCHA ή προσωρινό block. **3. Validation Integrity (Τύπος Αρχείου)** ASVS V12.2.1: Verify that files obtained from untrusted sources are validated to be of expected type based on the file's content. o Αν η εφαρμογή επιτρέπει μόνο PDF, DOC κ.λπ., επιβάλλουμε MIME/extension checks (μειώνει έμμεσα κάποιους τύπους κακόβουλων επιθέσεων). |
| Teacher Escalation (Elevation of Privilege) | • V4: Access Control o V4.1 General Access Control Design o V4.2 Operation Level Access Control | **1. Strict Role-Based Access Control (RBAC)** ASVS V4.1.2: Verify that all user and data attributes and policy information used by access controls cannot be manipulated by end users unless specifically authorized. o Ο ρόλος “Student” έχει μόνο readonly πρόσβαση σε βαθμούς του ίδιου. o Ο ρόλος “Teacher” διαχειρίζεται μόνο συγκεκριμένα μαθήματα. o Να μην υπάρχουν endpoints που επιτρέπουν αλλαγή ρόλου (εκτός αν το κάνει ο Admin). **2. Server-Side Role Validation** ASVS V4.2.1: Verify that sensitive data and APIs are protected against Insecure Direct Object Reference (IDOR) attacks targeting creation, reading, updating and deletion of records, such as creating or updating someone else's record, viewing everyone's records, or deleting all records. o Σε κάθε αίτημα που καλεί μια ευαίσθητη λειτουργία (update grade, upload υλικού), ελέγχουμε if (role == TEACHER) στο server. Δεν βασιζόμαστε σε client-side “role=teacher”. **3. Least Privilege File System Permissions** o V4.1.3 (Verify that the principle of least privilege exists) o Ο web server να μην έχει read/write δικαιώματα εκτός του απαραίτητου φακέλου. o Αν οι φοιτητές δεν δικαιούνται πρόσβαση σε κάποια αρχεία, αυτά παραμένουν εκτός “public” περιοχής. |

### 2.3.2 Mitigation Strategy

Προχωρώντας σε ένα ολοκληρωμένο σχέδιο προστασίας, εξετάζονται πρώτα τα ευρήματα της αξιολόγησής DREAD, ώστε να αναγνωριστούν ποιες απειλές επηρεάζουν περισσότερο την πλατφόρμα.

Με γνώμονα τη σοβαρότητα που προκύπτει από την ανάλυση δίνεται προτεραιότητα στις πιο επικίνδυνες ευπάθειες, όπως η πιθανότητα αλλοίωσης δεδομένων από SQL Injection ή η πλαστοπροσωπία χρηστών με αυξημένα δικαιώματα.

Αρχικά, υλοποιούνται άμεσα διορθωτικές δράσεις για να περιοριστεί ο κίνδυνος στις κρισιμότερες περιοχές ,ενώ παράλληλα προβλέπεται επόμενο στάδιο όπου θα ενσωματωθούν βελτιωμένες πρακτικές θωράκισης και επιπρόσθετα εργαλεία παρακολούθησης. Τέλος, θα υλοποιηθεί διαρκής παρατήρηση και επαναξιολόγηση του συστήματος εφαρμόζοντας περιοδικούς ελέγχους και αναβαθμίσεις, ώστε να διασφαλίζεται ότι η πλατφόρμα ανταποκρίνεται στις νέες απαιτήσεις και τις εξελισσόμενες τεχνικές επιθέσεις.

Με αυτήν τη στρατηγική, εξασφαλίζεται ότι η αντιμετώπιση των απειλών δεν εξαντλείται σε μια μονοδιάστατη υλοποίηση, αλλά αποτελεί μια δυναμική διαδικασία που προσαρμόζεται στη συνεχή εξέλιξη του περιβάλλοντος ασφαλείας.

1. Phase 1 (Immediate Risk Mitigation)
   * Σκοπός είναι η άμεση αντιμετώπιση των ευπαθειών που ενέχουν τον υψηλότερο κίνδυνο, όπως SQL Injection (που δύναται να καταστρέψει/διαρρεύσει ολόκληρη τη βάση), Spoofing (κλοπή διαπιστευτηρίων εκπαιδευτικών) ή DoS μέσω μαζικών uploads.
   * Εφαρμόζονται γρήγορα διορθωτικά μέτρα (π.χ. prepared statements για SQLi, brute force protection για login, όριο μεγέθους στα αρχεία), ώστε να μειωθεί δραστικά η πιθανότητα εκμετάλλευσης.
   * Επανεξέταση ή αναβάθμιση κρίσιμων υποσυστημάτων (π.χ. διαχωρισμός ρόλων με Role-Based Access Control, least privilege, βελτίωση authorization checks).
2. Phase 2 (Hardening & Enhancements)
   * Αφού οι κρίσιμες ευπάθειες περιοριστούν, υλοποιείται ένας ενισχυμένος σχεδιασμός που αντιμετωπίζει τις απειλές.
     + Προσθήκη βελτιωμένων πρακτικών (2FA για λογαριασμούς Teacher/Admin, WAF κανόνες για περισσότερη θωράκιση, central logging).
   * Σε αυτό το στάδιο γίνεται πιο οργανωμένη διαχείριση δικαιωμάτων και γενικότερη θωράκιση της αρχιτεκτονικής, ώστε να κλείσουν τυχόν λογικά “παράθυρα” που μπορεί να επιτρέψουν μελλοντικές επιθέσεις.
3. Phase 3 (Ongoing Monitoring & Continuous Improvement)
   * Εδώ επικεντρωνόμαστε στη συνεχή παρακολούθηση και διαρκή προσαρμογή του συστήματος στις νέες απειλές:
     + Υλοποιούνται διεργασίες για καταγραφή και ανάλυση συμβάντων (π.χ. SIEM, συστήματα alerting), έτσι ώστε να εντοπίζονται έγκαιρα ύποπτες ενέργειες (μαζικές αλλαγές βαθμών, ασυνήθιστα uploads κ.λπ.).
     + Εκτελούνται περιοδικά penetration tests, code reviews ή ακόμα και αυτοματοποιημένοι έλεγχοι (SAST/DAST) για να επιβεβαιώνεται ότι οι διορθώσεις παραμένουν αποτελεσματικές.
     + Το Threat Model ανανεώνεται όποτε αλλάζει η αρχιτεκτονική (π.χ. προσθήκη νέου ρόλου, νέου module), ώστε οι νέες ροές δεδομένων να λαμβάνονται υπόψη στην ανάλυση των κινδύνων.
   * Με αυτόν τον τρόπο, η ασφάλεια δεν αντιμετωπίζεται ως μία στατική λίστα διορθώσεων, αλλά ως δυναμική διαδικασία, ικανή να προσαρμόζεται σε διαρκώς μεταβαλλόμενες απαιτήσεις και εξελισσόμενες τεχνικές επιθέσεων.

# **3. Συμπεράσματα**

Το παρόν έργο είχε ως στόχο την εφαρμογή του OWASP Threat Modeling Process σε μια Web εφαρμογή e-learning (Open eClass) και την παρουσίαση των αποτελεσμάτων. Αρχικά, αναλύθηκε η εφαρμογή σε επιμέρους οντότητες και ροές δεδομένων όπως αποτυπώθηκαν στα Data Flow Diagrams.

Στη συνέχεια, με βάση το μοντέλο STRIDE, εντοπίστηκαν και κατηγοριοποιήθηκα οι απειλές. Για την ιεράρχηση των απειλών , επιλέχθηκε το μοντέλο DREAD, που απέδωσε συγκεκριμένες αριθμητικές βαθμολογίες (0–50). Οι απειλές που βρέθηκαν στην περιοχή “Critical” (≥40), όπως το SQL Injection ή η πλαστοπροσωπία εκπαιδευτικών, τέθηκαν σε απόλυτη προτεραιότητα για άμεση αντιμετώπιση. Ομοίως, απειλές “High” (≥30 και <40), όπως η ελλιπής καταγραφή ενεργειών (Insufficient Logging) ή η κλιμάκωση προνομίων (Teacher User Escalation), επισημάνθηκαν για γρήγορο μετριασμό.

Για την προστασία της πλατφόρμας, υιοθετήθηκαν βέλτιστες πρακτικές από το OWASP ASVS (Authentication, Access Control, Logging, Injection Prevention, File Upload κ.λπ.), εξασφαλίζοντας ότι οι λύσεις είναι συστημικά καλυμμένες και δε βασίζονται μόνο σε αποσπασματικές διορθώσεις. Ακολούθησε η διαμόρφωση μιας στρατηγικής τριών φάσεων (Phase 1: Immediate Fixes, Phase 2: Hardening & Enhancements, Phase 3: Ongoing Monitoring), ώστε οι διορθώσεις να υλοποιούνται με ρεαλιστικό τρόπο και να ελέγχονται συνεχώς για την αποτελεσματικότητά τους.

Παράλληλα, έγινε χρήση του Microsoft Threat Modelling Tool 2016, μέσω του οποίου καταγράφηκαν αυτόματα πολλές πιθανές απειλές (STRIDE) και παραγόταν report που επιβεβαίωνε (ή συμπλήρωνε) την ανάλυσή μας. Έτσι συνδυάστηκε η θεωρητική ανάλυση (OWASP-based) με ένα πρακτικό εργαλείο που επιτρέπει οπτική αναπαράσταση των στοιχείων και παραγωγή αναφορών.

Συνολικά, διαπιστώθηκε ότι η διαδικασία Threat Modeling επιτρέπει σε αρχικό στάδιο την προληπτική ανίχνευση των σημείων όπου η εφαρμογή είναι πιο ευάλωτη. Με τις προτεινόμενες διορθώσεις και τη συνεχή επιτήρηση, η e-learning πλατφόρμα μπορεί να διατηρεί υψηλό επίπεδο ασφάλειας, ακόμα και όταν εξελίσσεται τεχνολογικά ή προστίθενται νέες λειτουργίες. Επιπλέον, η προσέγγιση αυτή λειτουργεί ως αποτελεσματική προεργασία για penetration tests ή audit reviews που θα ακολουθήσουν, καθιστώντας το Threat Modeling ένα δυναμικό εργαλείο επικοινωνίας ανάμεσα σε ομάδες ανάπτυξης και ασφάλειας.

# **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>