**ΤΜΗΜΑ ΑΡΧΕΙΟΝΟΜΙΑΣ, ΒΙΒΛΙΟΘΗΚΟΝΟΜΙΑΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΗΣΗΣ**
**ΣΧΟΛΗ ΔΙΟΙΚΗΤΙΚΩΝ, ΟΙΚΟΝΟΜΙΚΩΝ ΚΑΙ ΚΟΙΝΩΝΙΚΩΝ ΕΠΙΣΤΗΜΩΝ**

---

**DEPARTMENT OF ARCHIVAL, LIBRARY AND INFORMATION STUDIES**
**SCHOOL OF MANAGEMENT, ECONOMICS AND SOCIAL SCIENCES**

**Πτυχιακή Εργασία**

# Ασφάλεια διαδικτυακών εφαρμογών: Σχεδιαστικές λύσεις και αξιολόγηση της ασφάλειας

**Συγγραφέας**

Κωνσταντίνα Μπεκιάρη (ΑΜ: 59913118)

**Επιβλέπων:** Ιωάννης Τριανταφύλλου

*Αθήνα, Μάρτιος 2021*

**Επιτροπή Εξέτασης**

**1.** Ονοματεπώνυμο: Ιωάννης Τριανταφύλλου

Ioannis Triantafyllou

Digitally signed by Ioannis Triantafyllou
Date: 2021.03.26 11:48:07 +02'00'

**2.** Ονοματεπώνυμο: Δημήτριος Κουής

DIMITRIOS KOUIS

Digitally signed by DIMITRIOS KOUIS
Date: 2021.03.26 14:39:46 +02'00'

**3.** Ονοματεπώνυμο: Μάρκος Δενδρινός

Markos Dendrinos

Digitally signed by Markos Dendrinos
Date: 2021.03.29 22:22:48 +03'00'

**Δήλωση συγγραφέα πτυχιακής/ διπλωματικής εργασίας**

Η κάτωθι υπογεγραμμένη Μπεκιάρη Κωνσταντίνα του Ιωάννη, με αριθμό μητρώου 59913118 φοιτήτρια του Πανεπιστημίου Δυτικής Αττικής της Σχολής Διοικητικών, Οικονομικών και Κοινωνικών Επιστημών του Τμήματος Αρχειονομίας, Βιβλιοθηκονομίας και Συστημάτων Πληροφόρησης, δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της πτυχιακής/διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Η Δηλούσα

# Ευχαριστίες – Αφιερώσεις

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή της πτυχιακής εργασίας μου κ. Ιωάννη Τριανταφύλλου αφενός για την εμπιστοσύνη του και αφετέρου για την υπομονή του κατά τη διάρκεια υλοποίησης της πτυχιακής εργασίας. Θα ήθελα επίσης να τον ευχαριστήσω για την πολύτιμη βοήθεια και την καθοδήγηση του.

Τέλος, θα ήθελα να ευχαριστήσω τον αδερφό μου, Μπεκιάρη Σπύρο, για την πολύτιμη βοήθεια και τη στήριξη του.

# Περίληψη

Με την εξέλιξη των τεχνολογιών πληροφοριών στη σύγχρονη εποχή έχει αναδειχθεί η αναγκαιότητα ύπαρξης εφαρμογών και υπηρεσιών, οι οποίες έχουν ως βάση τους το διαδίκτυο. Θέμα της συγκεκριμένης πτυχιακής εργασίας αντικατοπτρίζεται η ασφάλεια διαδικτυακών εφαρμογών. Με στόχο να μελετηθεί το παραπάνω φαινόμενο, θα μελετηθούν αρχικά τα χαρακτηριστικά των διαδικτυακών εφαρμογών και η ασφάλεια πληροφοριών στο διαδίκτυο. Βασικός στόχος της πτυχιακής εργασίας είναι να μελετηθεί η ασφάλεια των web-applications καθώς και το hacking (penetration testing) ως ένα σημαντικό εργαλείο αξιολόγησης της ασφάλειας τους. Εν συνεχεία, θα μελετηθούν οι προϋποθέσεις, τις οποίες θα πρέπει να πληροί μία διαδικτυακή εφαρμογή, με στόχο την προστασία των πληροφοριών της (π.χ. κρυπτογράφηση πληροφοριών).

Αρχικά, θα πραγματοποιηθεί βιβλιογραφική επισκόπηση επιστημονικών άρθρων σχετικών με την ασφάλεια διαδικτυακών εφαρμογών και την αξιολόγηση της μέσω hacking (penetration testing). Εν συνεχεία, θα πραγματοποιηθεί ανάλυση περιεχομένου αναφορικά με τις προϋποθέσεις σχεδίασης και ανάπτυξης διαδικτυακών εφαρμογών με στόχο την προστασία των πληροφοριών τους. Τέλος, βάσει των παραπάνω θα πραγματοποιηθεί σχεδίαση και ανάπτυξη μίας διαδικτυακής εφαρμογής καθώς και αξιολόγηση της ασφάλειας των πληροφοριών της μέσω hacking (penetration testing) κατά τη σχεδίαση και την ανάπτυξη της.

**Λέξεις-κλειδιά:** ασφάλεια πληροφοριών, ασφάλεια διαδικτυακών εφαρμογών, σχεδιασμός διαδικτυακών εφαρμογών, ανάπτυξη διαδικτυακών εφαρμογών, δικτυοπαραβίαση

# Abstract

Nowadays, in the modern world, due to the evolution of information technology, the necessity of web-applications' development has arisen. This thesis aims to study the security of web-applications and their information flow. The basic objective of my thesis is to study the security surrounding the web regarding information of web-applications. Another phenomenon, known as hacking (penetration testing), needs to be studied and regarded as a useful tool relating to the evaluation of their security. Besides, the security conditions surrounding the design and the development of a web-application, for instance encryption of information, need to be encompassed.

Firstly, a literature review of scientific articles, with reference to the information security of web-applications as well as the evaluation of it via hacking (penetration testing) will be conducted. Apart from this, a content analysis referring to the conditions of designing and developing web-applications, so as to secure their information, will be entailed. Finally, according to the information found, a web-application will be designed and developed at first and following that, the information security of it will be evaluated via hacking (penetration testing).



**Keywords:** information security, web-applications' security, web-applications' design, web-applications' development, penetration testing

# Πίνακας Περιεχομένων

# Πίνακας Εικόνων

# Πίνακας Πινάκων

# Πίνακας Ακρωνυμίων

| | |
|---|---|
| API | **A**pplication **P**rogramming **I**nterface |
| CSS | **C**ascading **S**tyle **S**heets |
| DKIM | **D**omain **K**eys **I**dentified **M**ail |
| DMARC | **D**omain based **M**essage **A**uthentication **R**eporting & **C**onformance |
| GUI | **G**raphical **U**ser **I**nterface |
| HDD | **H**ard **D**isk **D**rive |
| HTML | **H**yper**t**ext **M**arkup **L**anguage |
| HTTP | **H**yper**t**ext **T**ransfer **P**rotocol |
| HTTPS | **H**ypertext **T**ransfer **P**rotocol **S**ecure |
| IDS | **I**ntrusion **D**etection **S**ystem |
| IP | **I**nternet **P**rotocol |
| IPS | **I**ntrusion **P**revention **S**ystem |
| MIME | **M**ultipurpose **I**nternet **M**ail **E**xtensions |
| LDAP | **L**ightweight **D**irectory **A**ccess **P**rotocol |
| OWASP | **O**pen **W**eb **A**pplication **S**ecurity **P**roject |
| PHP | **H**ypertext **P**re**p**rocessor |
| PSW | **P**assword **S**tealing **W**are |
| RCE | **R**emote **C**ode **E**xecution |
| SPF | **S**ender **P**olicy **F**ramework |
| SSD | **S**olid **S**tate **D**rive |
| SSL | **S**ecure **S**ockets **L**ayer |
| SQL | **S**tructured **Q**uery **L**anguage |
| TLS | **T**ransport **L**ayer **S**ecurity |
| WAF | **W**eb **A**pplication **F**irewall |
| URL | **U**niform **R**esource **L**ocator |
| XML | E**x**tensible **M**arkup **L**anguage |
| XSS | Cross-**S**ite **S**cripting |
| XXE | **X**ML E**x**ternal **E**ntities |

# Εισαγωγή

Η ασφάλεια των διαδικτυακών εφαρμογών ήταν από τα πρώτα χρόνια της ανάπτυξης των υπολογιστών ο πιο κρίσιμος τομέας ανάπτυξης, καθώς η συγκέντρωση και η επεξεργασία τεράστιου όγκου πληροφοριών και στοιχείων σε ένα σύστημα και ο κίνδυνος απώλειας ή αλλοίωσης τους δημιούργησε επικίνδυνες προοπτικές. Μέχρι και την έλευση του διαδικτύου, οι κίνδυνοι ήταν πιο περιορισμένοι, καθώς ένας μικρός κύκλος ανθρώπων (επιστημόνων και τεχνικών) είχαν πρόσβαση και δυνατότητα επέμβασης στους υπολογιστές, τα προγράμματα εφαρμογών και τα δίκτυα, ενώ ακόμα μικρότερος ήταν ο αριθμός αυτών, οι οποίοι είχαν πρόσβαση στα δεδομένα **(Ransome & Stewart, 2015)**.

Από την αρχή της "ζωής" των ηλεκτρονικών υπολογιστών, υπήρξε μεγάλο ενδιαφέρον για την εύρεση τρόπων και τεχνικών που θα επέτρεπαν σε μία εφαρμογή να λειτουργήσει με διαφορετικό ή μη αναμενόμενο τρόπο απ' ότι είναι προκαθορισμένη. Τα κίνητρα ήταν διαφορετικά για τους ενδιαφερόμενους· οι συνέπειες, όμως, υπαρκτές.

Τη σύγχρονη εποχή η ασφάλεια στην πληροφορική αποτελεί ένα από τα μεγαλύτερα κλαδιά, η οποία με τη σειρά της χωρίζεται σε πολλές κατηγορίες. Μία από αυτές είναι και η ασφάλεια διαδικτυακών εφαρμογών (web-applications' security). Κάθε φορά οι δημιουργοί βελτιώνουν την εφαρμογή από άποψη ασφάλειας και αυτό οδηγεί τους μεν επιτιθέμενους, με στόχο την εκμετάλλευση ευπαθειών των εφαρμογών, στην αναζήτηση νέων μεθόδων και τους δε ερευνητές ασφάλειας, με στόχο την ενίσχυση της ασφάλειας των εφαρμογών, στην αναζήτηση νέων τεχνικών. Ουσιαστικά, οι δύο ομάδες βρίσκονται σε έναν διαρκή πόλεμο γνώσεων.

Η μία ομάδα είναι οι προγραμματιστές, οι οποίοι είναι υπεύθυνοι για τη σχεδίαση και κατ' επέκταση την ασφάλεια της εφαρμογής. Στην άλλη ομάδα βρίσκονται οι penetration testers, των οποίων στόχος είναι να εξάγουν πληροφορίες από τη βάση, να πάρουν δικαιώματα χρήστη ή και administrator,

χωρίς να είναι εξουσιοδοτημένοι, και γενικά να εντοπίσουν και να εκμεταλλευτούν αδυναμίες στην εφαρμογή.

Αυτή η ανάγκη συνεχούς βελτίωσης δημιούργησε ομάδες, φορείς και οργανισμούς, οι οποίοι ασχολούνται αποκλειστικά με την ασφάλεια διαδικτυακών εφαρμογών. Ερευνητικές ομάδες προσπαθούν να ανακαλύψουν νέες μεθόδους, εκπαιδεύουν υποψηφίους και προάγουν security awareness.

Σε έναν ψηφιακό κόσμο, όπου τα προσωπικά δεδομένα και η πληροφορία είναι στο επίκεντρο, η ασφάλεια έχει γίνει περισσότερο σημαντική από ποτέ.

# Κεφάλαιο 1. Διαδικτυακές εφαρμογές

Μία διαδικτυακή εφαρμογή επιτρέπει την εκκίνηση των λειτουργιών επεξεργασίας πληροφοριών από ένα πρόγραμμα περιήγησης και εκτελείται εν μέρει σε διακομιστή ιστού, διακομιστή εφαρμογών ή/ και διακομιστή βάσης δεδομένων. Μία διαδικτυακή εφαρμογή είναι μία εφαρμογή, η οποία έχει σχεδιαστεί με στόχο να εκτελείται σε περιβάλλον, το οποίο βασίζεται στο διαδίκτυο, και είναι κάτι παραπάνω από ένα σύνολο ρυθμίσεων ιστοσελίδων με συνδέσμους πλοήγησης **(Finkelstein et al., 2002)**.

## 1.1. Ασφάλεια διαδικτυακών εφαρμογών

Το Διαδίκτυο, όπως έχει διαμορφωθεί, επεκτείνεται ταχύτατα, καθώς σχετίζεται άμεσα με την καθημερινή ζωή. Τη σύγχρονη εποχή οι άνθρωποι αλληλεπιδρούν ολοένα και περισσότερο με το διαδίκτυο και έρχονται αντιμέτωποι με θέματα ασφάλειας προσωπικών και επιχειρηματικών δεδομένων. Αν και μέχρι πρόσφατα οι κυριότερες και πιο επικίνδυνες επιθέσεις αφορούσαν συνήθως κυβερνητικούς οργανισμούς και τραπεζικά ιδρύματα, τα τελευταία χρόνια όλος ο επιχειρηματικός κόσμος είναι ευάλωτος σε τέτοιες επιθέσεις **(Griffor, 2017)**.

Έτσι, προκύπτει ότι μία διαδικτυακή εφαρμογή είναι περισσότερο ευάλωτη από μία εφαρμογή, η οποία είναι τοπικά εγκατεστημένη, εξαιτίας της πολυπλοκότητας του σχεδιασμού της. Το πλήθος των τεχνολογιών, οι οποίες χρησιμοποιούνται για την ενίσχυση της λειτουργικότητας της εφαρμογής μεγιστοποιεί την έκθεση της σε κίνδυνο **(Μαυρίδης, 2015)**. Παραδείγματος χάρη, εφαρμογές, οι οποίες προσφέρουν υπηρεσίες online αγορών προσπαθούν να χρησιμοποιούν state-of-the-art τεχνολογίες και μεθόδους, με στόχο να επιτυγχάνουν τη δυνατότερη υψηλή ασφάλεια, συγκριτικά με άλλες, οι οποίες αποτελούνται από ένα απλό blog. Η διαδικασία ασφαλούς ανάπτυξης διαδικτυακών εφαρμογών αποτελεί πρόκληση για τους σχεδιαστές και τους

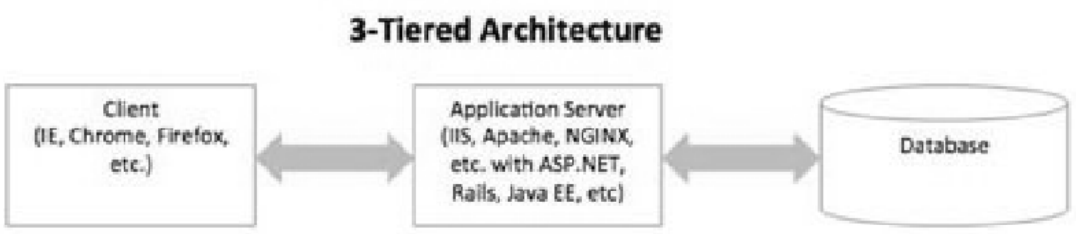
προγραμματιστές της εποχής μας. Έχει αποδειχθεί ότι κανένα πληροφοριακό σύστημα δεν είναι απολύτως ασφαλές και πάντοτε υπάρχουν περιθώρια βελτίωσης.

Η ασφάλεια των διαδικτυακών εφαρμογών διέπεται από μία σειρά βασικών αρχών, η οποία περιλαμβάνει την προστασία της εμπιστευτικότητας (Confidentiality), της ακεραιότητας (Integrity) και της διαθεσιμότητας (Availability) των δεδομένων. Η εμπιστευτικότητα αναφέρεται στη δυνατότητα διασφάλισης ότι οι πληροφορίες είναι ιδιωτικές και προστατεύονται από μη εξουσιοδοτημένη πρόσβαση. Η ακεραιότητα αντικατοπτρίζει την ακρίβεια των πληροφοριών και απαιτεί τεχνολογία και διαδικασίες που εμποδίζουν τη μη εξουσιοδοτημένη τροποποίηση πληροφοριών. Η διαθεσιμότητα αναφέρεται στη δυνατότητα να διασφαλιστεί ότι οι πληροφορίες είναι προσβάσιμες από τους τελικούς χρήστες σε εύθετο χρόνο. Εντούτοις, προκειμένου να διασφαλίζεται η ακεραιότητα των δεδομένων και η προστασία του χρήστη πολλές φορές “θυσιάζεται” η απόδοση και αυξάνεται η πολυπλοκότητα του συστήματος ή της εφαρμογής. Αυτό έχει ως αποτέλεσμα την ύπαρξη σύνθετων αρχιτεκτονικών και σχεδιάσεων μοντέλων εφαρμογών, με στόχο να επιτευχθεί ένα ιδανικό αποτέλεσμα.

Με στόχο την παρεμπόδιση των επιθέσεων, τις οποίες αντιμετωπίζουν οι εφαρμογές, πολλοί οργανισμοί αναζητούν παραδοσιακές λύσεις ασφάλειας δικτύου. Η ανάπτυξη ενός τείχους προστασίας δικτύου, ενός συστήματος ανίχνευσης εισβολών ή ενός συστήματος πρόληψης εισβολής είναι μερικοί τρόποι προστασίας. Η παραδοσιακή προσέγγιση στην ασφάλεια δικτύου στοχεύει στην προστασία πόρων όπως διακομιστές, σταθμούς εργασίας, εκτυπωτές, εσωτερικές βάσεις δεδομένων και άλλους πόρους δικτύου. Τα εργαλεία, τα οποία χρησιμοποιούνται για τη διασφάλιση αυτών των πόρων, εμποδίζουν την πρόσβαση σε συγκεκριμένες θύρες ή υπηρεσίες δημιουργώντας κανόνες αδειοδότησης ή άρνησης στο δίκτυο **(Ransome & Stewart, 2015)**.

## 1.2. Αρχιτεκτονική διαδικτυακών εφαρμογών

Η αρχιτεκτονική μίας εφαρμογής εξαρτάται από πολλούς παράγοντες, οι οποίοι σχετίζονται με τους στόχους του δημιουργού – εταιρείας, με το κεφάλαιο το οποίο θα επενδυθεί σε αυτήν καθώς επίσης και με τα ήδη υπάρχοντα υλικά και λογισμικά, τα οποία πιθανόν θα διαθέτουν. Αν και πολλές εφαρμογές βασίζονται καθολικά σε cloud based services, ένα πολύ συνηθισμένο μοντέλο, το οποίο συναντάται, είναι η αρχιτεκτονική 3-tier. Τα τρία επίπεδα του μοντέλου αυτού είναι τα: **Presentation tier**, **Application tier** και **Data tier**.



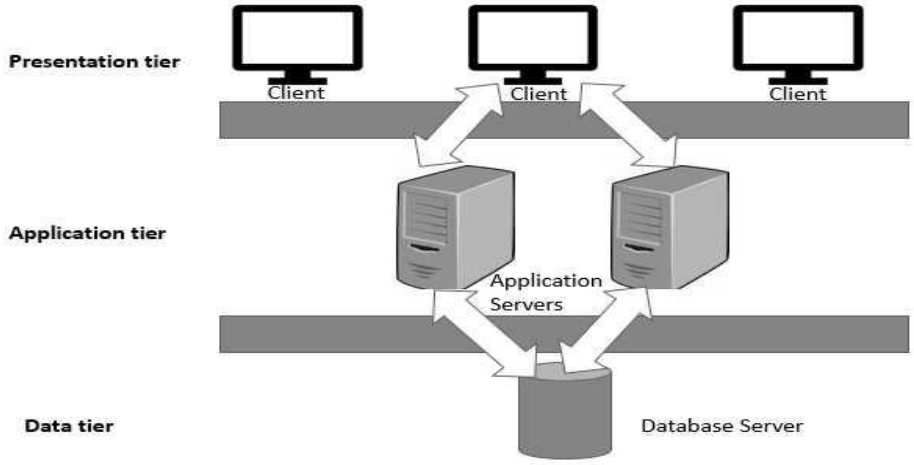

**Εικόνα 1. Αρχιτεκτονική 3 tier (πηγή: Gacheru Evans, 2019)**

Το Presentation tier (επίπεδο παρουσίασης) είναι αυτό με το οποίο αλληλεπιδρά ο χρήστης. Τα γραφικά μίας ιστοσελίδας δηλαδή, τα οποία δημιουργούνται με γλώσσα περιγραφής HTML (**H**yper**T**ext **M**arkup **L**anguage) και υποστηρίζονται από CSS (**C**ascading **S**tyle **S**heets) και Javascript. Το επίπεδο αυτό επικοινωνεί με το Application tier, καθώς οι ενέργειες, τις οποίες θα επιλέξει ο χρήστης, θα επεξεργαστούν από την εφαρμογή για την εξυπηρέτηση του αιτήματος του.

Το Application tier (επίπεδο εφαρμογής) είναι υπεύθυνο για τη λειτουργικότητα και τη λογική της εφαρμογής. Ελέγχει τις επιλογές του χρήστη και έπειτα από την επεξεργασία τους κατευθύνει τον έλεγχο σε άλλες συναρτήσεις για την εξυπηρέτηση του αιτήματος του.

Το τελευταίο επίπεδο είναι το Data tier (επίπεδο δεδομένων). Τα δεδομένα τα οποία πιθανόν περιέχει μία εφαρμογή, όπως ονόματα χρηστών και κωδικούς, διάφορες πληροφορίες για τον κάθε χρήστη, μηνύματα, κείμενα σε blog, αποθηκεύονται σε βάσεις δεδομένων. Παρέχονται μηχανισμοί, με στόχο το επίπεδο εφαρμογής να διαβάσει, να διαγράψει και να επεξεργαστεί τα δεδομένα. Συνήθως, η βάση και το API (**A**pplication **P**rogramming **I**nterface) της βρίσκονται σε ξεχωριστούς servers (database servers), οι οποίοι έχουν περιορισμένη πρόσβαση ακόμα και στο εσωτερικό δίκτυο.

Η διάδοση της πληροφορίας στην αρχιτεκτονική 3-tier, η οποία έχει περιγραφεί παραπάνω, συνήθως ξεκινάει από τις ενέργειες του χρήστη στο επίπεδο παρουσίασης. Όταν ο client θα γράψει το domain στο URL (**U**niform **R**esource **L**ocator) bar στον περιηγητή του και θα πατήσει enter, το αίτημα στέλνεται στον server και αυτός πρέπει να επιστρέψει την κατάλληλη σελίδα HTML. Εάν για παράδειγμα στη συνέχεια, ο χρήστης επιλέξει να συνδεθεί στη σελίδα, με την προϋπόθεση ότι διαθέτει κάποιον ενεργό λογαριασμό, το αίτημα του θα σταλεί μαζί με τα δεδομένα εισόδου, τα οποία θα καταχωρήσει, θα μεταβούν στο επίπεδο εφαρμογής και από εκεί στο επίπεδο δεδομένων. Αφού πραγματοποιηθεί ο έλεγχος επαλήθευσης της ταυτότητας του χρήστη (authentication), η εφαρμογή θα του επιτρέψει να συνδεθεί σε περίπτωση επιτυχίας ή θα του ζητήσει να προσπαθήσει ξανά να συνδεθεί μαζί με κάποιο μήνυμα σφάλματος, σε περίπτωση αποτυχίας. Στο παραπάνω παράδειγμα αποδεικνύεται ότι η διάδοση της πληροφορίας είναι αμφίδρομη, καθώς το κάθε επίπεδο επικοινωνεί με το “γειτονικό” του και η σωστή λειτουργία όλων αποσκοπεί στην εξυπηρέτηση του χρήστη.




**Εικόνα 2. Διάδοση πληροφορίας (πηγή: Pethuru Raj, Anupama Raman, Harihara Subramanian, 2017)**

## 1.3. Χαρακτηριστικά διαδικτυακών εφαρμογών

Όπως έχει ήδη αναφερθεί κάθε εφαρμογή έχει συγκεκριμένο σκοπό, οπότε απαιτούνται διαφορετικές τεχνολογίες σχεδίασης και ανάπτυξης. Ωστόσο, όλες οι εφαρμογές έχουν κοινά χαρακτηριστικά, τα οποία είναι απαραίτητα για τη λειτουργικότητα τους. Σύμφωνα με το μοντέλο αρχιτεκτονικής 3-tier, το οποίο έχει περιγραφεί προηγουμένως, θα αναλυθούν και τα βασικά στοιχεία, τα οποία τις χαρακτηρίζουν.

Όλες οι εφαρμογές, οι οποίες απευθύνονται σε απλό χρήστη, είτε είναι διαδικτυακές είτε όχι, απαιτείται να έχουν γραφικό περιβάλλον ή στην αγγλική ορολογία GUI (**G**raphical **U**ser **I**nterface). Ακόμα και τα λειτουργικά συστήματα Windows, Linux, Android κ.λπ. έχουν GUI. Αυτό διευκολύνει πολύ τη χρήση της εφαρμογής, καθώς ο χρήστης δεν απαιτείται να απομνημονεύει εντολές για τη λειτουργία της και κατ' επέκταση εξοικονομεί χρόνο. Επίσης, η σωστή σχεδίαση γραφικών και η παροχή έξυπνης βοήθειας μέσα από αυτά, όταν απαιτείται, βοηθάει τον χρήστη να καταλάβει τον λόγο ύπαρξης της και να εκμεταλλευτεί

όλες τις διαθέσιμες δυνατότητες της. Ένας από τους λόγους για τους οποίους τα Windows κατέλαβαν την αγορά στους προσωπικούς υπολογιστές έναντι των Unix-Like συστημάτων είναι επειδή στην αρχή της ζωής και των δύο τα πρώτα παρείχαν πιο απλό user experience, κάτι το οποίο ένας απλός χρήστης επιζητάει διαρκώς.

Η ομάδα, η οποία είναι υπεύθυνη για τη δημιουργία των σελίδων, είναι οι Front-end developers. Ανήκουν και αυτοί στη γενικότερη ομάδα των προγραμματιστών· θα χρειαστεί όμως να συνεργαστούν με τους σχεδιαστές (web designers) και τους προγραμματιστές της εφαρμογής. Ένας από τους πιο σημαντικούς παράγοντες επιτυχίας μίας εφαρμογής είναι η εμφάνιση της. Το έργο τους και το τελικό προϊόν, το οποίο θα είναι διαθέσιμο, είναι ιδιαίτερης σημασίας και πρέπει να δίνεται έμφαση σε αυτό.

Οι τεχνολογίες, οι οποίες χρησιμοποιούνται, είναι η γλώσσα περιγραφής HTML, η CSS και η Javascript. Η HTML είναι η βάση κάθε σελίδας. Εκεί περιγράφονται οντότητες και ομαδοποιούνται σε κλάσεις, ανάλογα με το τι θέλει να επιτύχει ο developer. Η CSS χρησιμοποιείται σε συνδυασμό με την HTML. Μέσα στο αρχείο .css περιγράφεται ο τρόπος εμφάνισης των κλάσεων και των οντοτήτων. Προστίθενται χρώματα/ γραμματοσειρές, αλλάζει το background σειρών ανάλογα με τη θέση τους και γενικά η σελίδα γίνεται πιο ευδιάκριτη και κατανοητή. Αφού πραγματοποιηθεί ο συνδυασμός τους και έχει επιτευχθεί το επιθυμητό αποτελέσμα, προστίθεται η Javascript. Η Javascript επίσης παρέχει τη δυνατότητα να βελτιώσει τα γραφικά σε μία σελίδα, συνήθως όμως χρησιμοποιείται με στόχο να διευκολύνει την περιήγηση σε αυτήν και την κατανόηση της. Παραδείγματος χάρη, ενδεχομένως να εμφανίζει text ή να αλλάζει τον κέρσορα πάνω από links ή εικόνες - τα οποία θα κάνουν redirect σε κάποιο άλλο σημείο της εφαρμογής - να χρησιμοποιήσει το ρολόι του συστήματος, να μεταφέρει δεδομένα κ.α. Αν και οι τρεις τεχνολογίες, οι οποίες περιγράφηκαν, συναντώνται στις περισσότερες σελίδες, μόνο η HTML είναι βασική για την ύπαρξη της.

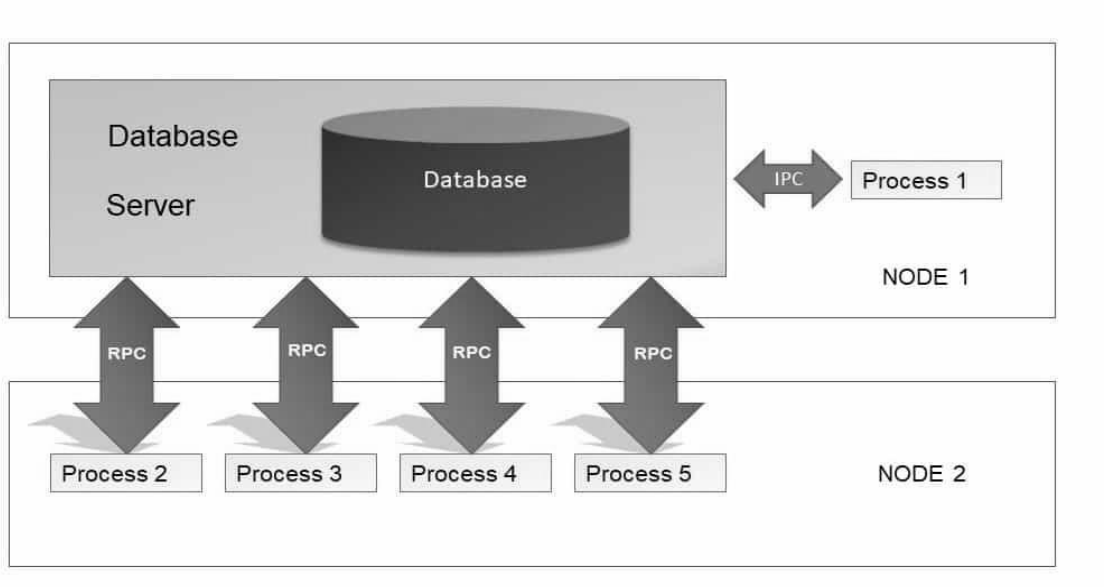

**Εικόνα 3. Front end development (πηγή: Petoskey C., 2010)**

Προχωρώντας στο επίπεδο Application tier, εκεί θα βρίσκεται κάποια γλώσσα προγραμματισμού. Οι πιο συνηθισμένες είναι η Java, η PHP (**H**ypertext **P**re**p**rocessor) και η Python. Η PHP αν και χρησιμοποιείται, είναι πιθανό να εμφανίσει security holes καθιστώντας την εφαρμογή ευάλωτη σε ποικιλία επιθέσεων. Η Java και η Python χρησιμοποιούν εξειδικευμένες βιβλιοθήκες και συναρτήσεις για αυτόν τον λόγο. Οι προγραμματιστές, οι οποίοι θα αναπτύξουν την εφαρμογή, ονομάζονται Back-end developers.

Η επιλογή γλώσσας εξαρτάται από παράγοντες, οι οποίοι έχουν σχέση με την εταιρεία ή τον δημιουργό της εφαρμογής. Επίσης, εξαρτάται από το διαθέσιμο υλικό (hardware) και το λογισμικό (software), το οποίο ήδη υπάρχει. Κάθε γλώσσα έχει τα μειονεκτήματα και τα πλεονεκτήματα της. Αυτά είναι η απόδοση, οι απαιτούμενες ανθρώπινες ώρες, με στόχο να αναπτυχθεί μία εφαρμογή, η διαθέσιμη υποστήριξη καθώς και οι διαθέσιμοι πόροι. Συγκρίνοντας την Python με την Java, η πρώτη είναι πολύ εύκολη στο συντακτικό, μειώνοντας τα λάθη και καθιστώντας τον developer ικανό να ολοκληρώσει σε λιγότερο χρόνο. Όμως, η Java είναι πιο αποδοτική πολλές φορές και γι’ αυτό προτιμάται. Ανάλογα με τη γλώσσα προγραμματισμού, θα χρησιμοποιείται και αντίστοιχο λογισμικό, με το οποίο θα υποστηριχτεί συνολικά το project. Η PHP για να “τρέξει” χρειάζεται

server software (εξυπηρετητή), από τους οποίους πολύ γνωστός είναι ο Apache. Η Python χρησιμοποιεί frameworks, όπως το Django και το Flask, το οποίο μοιάζει με το πρώτο, με τη διαφορά ότι είναι πιο εύκολο στη χρήση και την εκμάθηση και προτείνεται σε σχέση με μικρότερες εφαρμογές.

Το τελευταίο επίπεδο είναι το Data tier, το οποίο, όπως περιγράφηκε παραπάνω, αποθηκεύει δεδομένα απαραίτητα για τη σωστή λειτουργία της εφαρμογής. Τον τύπο των δεδομένων και τη διαχείριση τους την αναλαμβάνει επίσης ο Back-end developer. Είναι πολύ πιθανό αυτός να επιλέξει και την τεχνολογία, η οποία θα χρησιμοποιηθεί. Παραδείγματος χάρη, για μικρό όγκο δεδομένων προτείνεται η SQLite3. Δεν είναι απαραίτητο όλες οι εφαρμογές να χρειάζονται βάση δεδομένων για τη λειτουργία τους. Ένας ιστότοπος, ο οποίος έχει μόνο στατικές HTML σελίδες, χωρίς να δίνει δυνατότητα σύνδεσης, δεν χρειάζεται βάση δεδομένων.




**Εικόνα 4. Εξυπηρετητής βάσης δεδομένων (πηγή: https://medium.com/@gacheruevans0/2-tier-vs-3-tier-architecture-26db56fe7e9c)**

Εκτός από τις τεχνολογίες και το λογισμικό (software), τα οποία θα χρησιμοποιηθούν, ένα ακόμα πολύ σημαντικό χαρακτηριστικό είναι το υλικό (hardware). Υπεύθυνοι για την εγκατάσταση υλικού και λογισμικού είναι οι System Administrators (διαχειριστές συστήματος). Ανάλογα με το διαθέσιμο κεφάλαιο για κάποιο έργο, θα εγκαταστήσουν τα απαραίτητα λογισμικά και ίσως προβούν σε αγορές υλικού. Πολύ συχνά είναι υπεύθυνοι και για τη διαχείριση δικτύου· υπάρχει όμως ξεχωριστός κλάδος γι' αυτό, οι Network Administrators.

Αφού ολοκληρωθεί η εγκατάσταση των απαραίτητων τεχνολογιών, οι System Administrators πρέπει να κάνουν τις απαραίτητες ρυθμίσεις, με στόχο να λειτουργούν σωστά. Συγκεκριμένα, πρέπει να ασχοληθούν με την επικοινωνία μεταξύ των λογισμικών, τα οποία είναι απαραίτητα για τη συνολική λειτουργία της εφαρμογής, να δώσουν στις ρουτίνες και τα προγράμματα τα δικαιώματα, τα οποία χρειάζονται, καθώς και να ορίσουν υπό–διαχειριστές. Είναι πιθανό, δηλαδή, ένας προγραμματιστής να έχει δικαιώματα διαχειριστή στη βάση δεδομένων ή ακόμα και στον data server, έτσι ώστε να φτιάξει ό,τι χρειάζεται για την εφαρμογή.

# Κεφάλαιο 2. Σχεδίαση και ανάπτυξη διαδικτυακών εφαρμογών

## 2.1. Αρχές ασφαλούς προγραμματισμού

Για την ανάπτυξη αξιόπιστων διαδικτυακών εφαρμογών έχουν προταθεί κάποιες αρχές ασφαλούς προγραμματισμού από ανεξάρτητους οργανισμούς (όπως ο οργανισμός OWASP - **O**pen **W**eb **A**pplication **S**ecurity **P**roject) ή ομάδες ασφάλειας (όπως η ομάδα CERT του πανεπιστημίου Carnegie Mellon), οι οποίες συνοπτικά περιγράφονται παρακάτω:

i. Η **Αρχή της Ελάχιστης Επιφάνειας Επίθεσης** (Minimum Attack Surface Principle) κατά την οποία πρέπει να επιδιώκεται η μείωση της ονομαζόμενης "επιφάνειας επίθεσης" (attack surface), προσθέτοντας εκείνα τα ελάχιστα χαρακτηριστικά, τα οποία είναι απαραίτητα για τη λειτουργικότητα της.

ii. Η **Αρχή του Ελάχιστου Προνομίου** (Principle of Least Privilege), η οποία ορίζει στον κάθε χρήστη τα ελάχιστα προνόμια, τα οποία απαιτούνται για την εκτέλεση μίας εργασίας.

iii. Η χρήση περισσότερων του ενός μηχανισμών ασφάλειας.

iv. Η πιστή εφαρμογή της **Αρχής του Διαχωρισμού Καθηκόντων** (Separation of Duties) είναι εξίσου σημαντική, καθώς είναι πιο εύκολο να οριστούν τα δικαιώματα της κάθε διεργασίας ή του κάθε ρόλου, ο οποίος εμπλέκεται στην επιτέλεση μίας εργασίας, χωρίς να δημιουργηθούν γκρίζες περιοχές αμφισβήτησης δικαιωμάτων.

v. Θα πρέπει να αποφεύγεται επίσης η απόκρυψη του τρόπου λειτουργίας των μηχανισμών ασφάλειας, η οποία συναντάται ως "**security by obscurity**".

vi. Σε συνδυασμό με την **Αρχή της Ελάχιστης Επιφάνειας Επίθεσης**, προτείνεται η εφαρμογή της **Αρχής της Απλότητας**, η οποία ορίζει ότι μεταξύ δύο λύσεων ο μηχανικός λογισμικού (Software Engineer) θα πρέπει να επιλέξει την πιο απλή λύση. Η αρχή αυτή αναφέρεται στη βιβλιογραφία και ως το "Ξυράφι του Όκαμ" (Occam's Razor) **(Μαυρίδης, 2015)**.

## 2.2. Σημαντικά ζητήματα ασφάλειας

Κατά τη σχεδίαση μίας διαδικτυακής εφαρμογής πρέπει να ληφθούν υπόψιν τα πιο βασικά σημεία, στα οποία μία εφαρμογή είναι πιθανό να παρουσιάζει ευπάθειες. Παραδείγματος χάρη, ζητήματα επικύρωσης δεδομένων εισόδου, αυθεντικοποίησης και εξουσιοδότησης, χρήσης κρυπτογραφίας και προστασίας ευαίσθητων δεδομένων κ.α. **(Μαυρίδης, 2015)**. Στη συνέχεια, θα αναφερθούν συνοπτικά τα πιο σημαντικά σημεία, τα οποία πρέπει να εξετάζονται, προκειμένου να αντιμετωπιστούν ή και να αποφευχθούν ζητήματα παραβίασης της ασφάλειας της εκάστοτε εφαρμογής.

Εξίσου σημαντικό είναι κατά τη φάση της σχεδίασης μίας διαδικτυακής εφαρμογής να εξεταστεί η **πολιτική ασφάλειας** [1] , η οποία θα χρησιμοποιηθεί. Οι απαιτήσεις ασφάλειας ενός υπολογιστικού συστήματος προσδιορίζονται διαμέσου μίας πολιτικής ασφάλειας.

Ακόμα, θα πρέπει να έχει μελετηθεί σωστά η δομή του δικτύου, το οποίο παρέχεται από το περιβάλλον, το οποίο θα φιλοξενήσει την εφαρμογή, καθώς και οι βασικές απαιτήσεις ασφάλειας σε ό,τι αφορά στους κανόνες φιλτραρίσματος, τους περιορισμούς χρήσης θυρών, τα υποστηριζόμενα πρωτόκολλα επικοινωνίας, κ.α. Σε αυτό το σημείο θα πρέπει να σημειωθεί ότι το πρωτόκολλο HTTP (**H**yper**T**ext **T**ransfer **P**rotocol) είναι ένα καταστατικό πρωτόκολλο, το οποίο σημαίνει ότι ο εντοπισμός της κατάστασης κάθε χρήστη

**[1] Η πολιτική ασφάλειας (security policy) ενός υπολογιστικού συστήματος είναι ένα σύνολο από αρχές (principles) και οδηγίες υψηλού επιπέδου (high level guidelines), οι οποίες αφορούν στη σχεδίαση και διαχείριση συστημάτων ελέγχου προσπέλασης.**

ανά σύνοδο είναι ευθύνη της εφαρμογής. Έτσι, η εφαρμογή θα πρέπει να είναι σε θέση να προσδιορίσει τον χρήστη, χρησιμοποιώντας κάποια μορφή ελέγχου ταυτότητας **(Μαυρίδης, 2015)**.

Εν συνεχεία, η σωστή **επικύρωση των δεδομένων εισόδου** αποτελεί μία σημαντική τεχνική άμυνας κατά των επιθέσεων σε μία διαδικτυακή εφαρμογή. Επομένως, θα πρέπει:

- να ληφθεί υπόψιν ότι όλες οι είσοδοι δεν προέρχονται από αξιόπιστη πηγή.
- να χρησιμοποιείται ένας εξυπηρετητής αυθεντικοποίησης.
- να φιλτραριστούν, περιοριστούν και να απορριφθούν όλες οι είσοδοι σύμφωνα με τους κανόνες φιλτραρίσματος.

Η **αυθεντικοποίηση** ορίζεται ως η επιβεβαίωση της ταυτότητας του χρήστη. Η αναγνώριση και η αυθεντικοποίηση των χρηστών πραγματοποιείται είτε με την εισαγωγή κωδικού (π.χ. password/ PIN) είτε με τη χρήση προσωποποιημένης κάρτας ταυτοποίησης (smart card) ή ακόμα, με τον έλεγχο βιομετρικών χαρακτηριστικών (π.χ. δακτυλικό αποτύπωμα).

Κάποια σημεία της αυθεντικοποίησης, τα οποία πρέπει να ληφθούν υπόψιν, είναι:

- ο εντοπισμός των σημείων της εφαρμογής, όπου απαιτείται έλεγχος ταυτότητας
- η επαλήθευση της ταυτότητας του χρήστη, με τη χρήση username και password
- ο προσδιορισμός της ταυτότητας του χρήστη σε επόμενες κλήσεις με κάποια μορφή token για παρουσίαση της αρχικά επαληθευμένης ταυτότητας (προβλήματα απώλειας, αντιγραφής και παραποίησης ;)
- ότι η πιο ανθεκτική σε παραβιάσεις αυθεντικοποίηση πραγματοποιείται μέσω των βιομετρικών χαρακτηριστικών του χρήστη.

Για την αντιμετώπιση των ζητημάτων αυθεντικοποίησης θα πρέπει:

- να γίνεται διαχωρισμός της εφαρμογής σε δημόσιες και ιδιωτικές ζώνες χρήσης.
- να υποστηρίζεται η δυνατότητα άμεσης απενεργοποίησης λογαριασμού.
- να μην αποθηκεύονται passwords σε αποθηκευτικό χώρο του τελικού χρήστη. (κίνδυνος ο χρήστης να ξεχάσει τον κωδικό password ;)
- να απαιτείται η χρήση ισχυρού password.
- να απαιτείται ανανέωση του κάθε password (ανά 3 ή 6 μήνες).
- να μην μεταδίδονται απροστάτευτα τα passwords μέσω του δικτύου. (κίνδυνος υποκλοπής ;)
- να προστατεύονται κατάλληλα τα cookies.

Η **εξουσιοδότηση** καθορίζει όλες εκείνες τις ενέργειες, τις οποίες έχει τη δυνατότητα να εκτελέσει ένας χρήστης, ο οποίος έχει επαληθεύσει την ταυτότητα του. Μία λανθασμένη εξουσιοδότηση είναι πιθανό να οδηγήσει σε αποκάλυψη και αθέμιτη μεταβολή πληροφοριών από έναν επιτιθέμενο. Κάποιες από τις προτεινόμενες λύσεις είναι:

- η χρήση πολλαπλών ελέγχων εξουσιοδότησης
- ο περιορισμός των δικαιωμάτων του χρήστη
- η χρήση επιπέδων εξουσιοδότησης.

Θα πρέπει επίσης να δοθεί προσοχή στον **τρόπο διαχείρισης** των **ρυθμίσεων** και των **παραμέτρων** της εφαρμογής. Οι συνέπειες από την εκμετάλλευση λανθασμένων ή αντικρουόμενων ρυθμίσεων ενδεχομένως να είναι ιδιαίτερα σημαντικές. Οι πιο συνήθεις πρακτικές, οι οποίες ακολουθούνται για ζητήματα διαχείρισης των ρυθμίσεων, είναι:

- η διασφάλιση της ελεγχόμενης πρόσβασης στις διεπαφές διαχείρισης
- η προστασία του χώρου, όπου αποθηκεύονται οι ρυθμίσεις
- η ανάπτυξη διαφορετικών επιπέδων διαχείρισης για κάθε χρήστη

- η χρήση λογαριασμών με ελάχιστα δικαιώματα για κάθε ξεχωριστή υπηρεσία.

Οι εφαρμογές, οι οποίες επεξεργάζονται και αποθηκεύουν **προσωπικά δεδομένα** χρηστών, θα πρέπει να λαμβάνουν ειδικά μέτρα με στόχο τη διασφάλιση της εμπιστευτικότητας και της ακεραιότητας των δεδομένων αυτών, όπως αναφέρθηκε παραπάνω. Επίσης, πρέπει να προστατεύονται επαρκώς τα ευαίσθητα δεδομένα, τα οποία χρησιμοποιούνται από την εφαρμογή, όπως τα passwords και οι ρυθμίσεις σύνδεσης με το σύστημα διαχείρισης βάσεων δεδομένων ***(Εικόνα 5)***.

Κάποιες από τις τακτικές, οι οποίες ακολουθούνται για τη διασφάλιση της προστασίας των ευαίσθητων δεδομένων, είναι:

- η αποθήκευση μόνο των "απαραίτητων" ευαίσθητων δεδομένων για την εκάστοτε λειτουργία της εφαρμογής
- η αποφυγή αποθήκευσης ευαίσθητων δεδομένων μέσα στον κώδικα της εφαρμογής
- η κρυπτογραφημένη αποθήκευση των ρυθμίσεων σύνδεσης σε συστήματα διαχείρισης βάσεων δεδομένων, συνθηματικών, κλειδιών κρυπτογράφησης κ.λπ.
- η εκτεταμένη χρήση κρυπτογραφικών τεχνικών.

Οι διαδικτυακές εφαρμογές αναπτύσσονται με βάση το πρωτόκολλο HTTP, το οποίο, όπως αναφέρθηκε, λειτουργεί χωρίς επίβλεψη της κατάστασης σύνδεσης του χρήστη. Έτσι, η ίδια η εφαρμογή είναι υπεύθυνη για τη **διαχείριση** μίας **συνόδου**. Η ασφάλεια συνόδου είναι πολύ σημαντική σε σχέση με τη συνολική ασφάλεια μίας διαδικτυακής εφαρμογής.

Οι ακόλουθες πρακτικές ενισχύουν την ασφάλεια της διαχείρισης συνόδου μίας διαδικτυακής εφαρμογής:

- η εφαρμογή πρωτοκόλλου SSL (**S**ecure **S**ockets **L**ayer) για την προστασία των μεταδιδόμενων δεδομένων
- η κρυπτογράφηση των cookies
- ο περιορισμός του χρόνου ζωής μίας ενεργής συνόδου
- η προστασία από το ενδεχόμενο υποκλοπής κατάστασης μίας συνόδου από μη εξουσιοδοτημένους χρήστες.

Οι διαδικτυακές εφαρμογές συχνά χρησιμοποιούν **κρυπτογραφικές μεθόδους** με στόχο να προστατεύσουν τα δεδομένα κατά την αποθήκευση ή τη μετάδοση τους.

Με τις ακόλουθες πρακτικές ενισχύεται η ασφάλεια των διαδικτυακών εφαρμογών, όταν χρησιμοποιείται κρυπτογραφία:

- με τη χρήση κατάλληλου αλγορίθμου, με το κατάλληλο μήκος κλειδιού, με την προϋπόθεση ότι υποστηρίζεται μεταβλητό μήκος
- και με την προστασία των κρυπτογραφικών κλειδιών.

Με τις **επιθέσεις χειραγώγησης παραμέτρων** ο επιτιθέμενος αποσκοπεί στη μεταβολή των δεδομένων, τα οποία ανταλλάσσονται μεταξύ του χρήστη και της διαδικτυακής εφαρμογής. Αυτά τα δεδομένα είναι δυνατόν να είναι αλφαριθμητικά ερωτήματος, πεδία φόρμας, cookies, επικεφαλίδες HTTP κ.α. Οι ακόλουθες πρακτικές αποσκοπούν στην προστασία μίας διαδικτυακής εφαρμογής από τη χειραγώγηση των παραμέτρων της:

- η κρυπτογράφηση των cookies
- η επικύρωση των δεδομένων εισόδου
- ο προσεκτικός έλεγχος των επικεφαλίδων HTTP.

Ο **ασφαλής χειρισμός εξαιρέσεων** είναι ικανός να βοηθήσει στην πρόληψη κάποιων επιθέσεων σε επίπεδο εφαρμογής, όπως παραδείγματος χάρη άρνησης

εξυπηρέτησης, η οποία προσβάλλει την αρχή της διαθεσιμότητας. Ακόμα, χρησιμοποιείται με στόχο να αποτρέψει την αποκάλυψη σημαντικών πληροφοριών στον τελικό χρήστη. Οι ακόλουθες πρακτικές βοηθούν στη διασφάλιση του σωστού χειρισμού εξαιρέσεων από μία διαδικτυακή εφαρμογή, σύμφωνα με τις οποίες:

- απαγορεύεται η επιστροφή ευαίσθητων πληροφοριών στον χρήστη.
- προτείνεται η αναλυτική καταγραφή των μηνυμάτων λαθών.
- είναι απαραίτητη η ύπαρξη χειρισμού εξαιρέσεων.

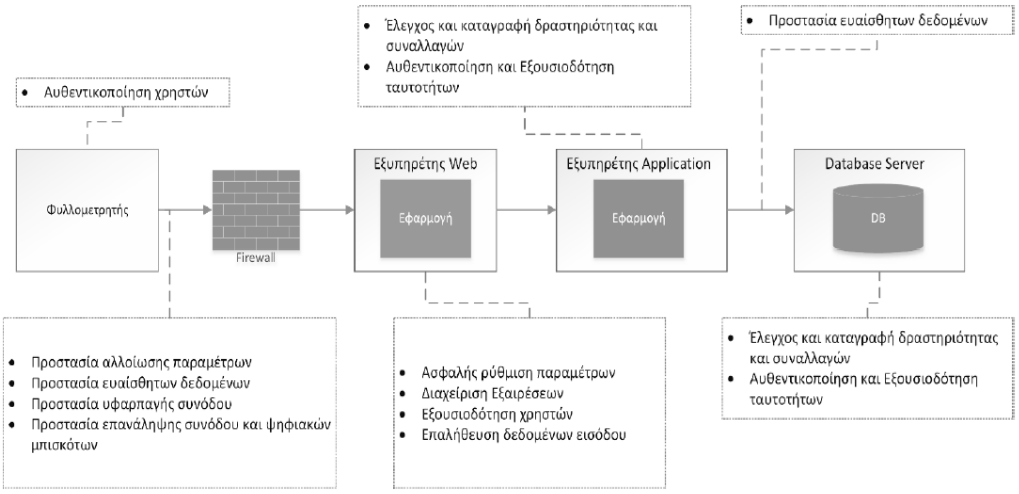
Τέλος, θα πρέπει να ελέγχονται και να καταγράφονται όλες οι δραστηριότητες σε κάθε επίπεδο λειτουργίας της εφαρμογής. Αξιοποιώντας και μελετώντας τα αρχεία καταγραφής (logs), είναι δυνατόν να εντοπιστούν ύποπτες δραστηριότητες στο σύστημα. Μάλιστα, σε πολλές περιπτώσεις είναι δυνατόν να προληφθεί μία επικείμενη επίθεση. Σε γενικές γραμμές, ο έλεγχος των αρχείων καταγραφής ενδεχομένως να αποδειχθεί πολύ σημαντικός, εάν πραγματοποιηθεί σε πραγματικό χρόνο λειτουργίας της εφαρμογής.

Παρακάτω ακολουθούν πρακτικές, οι οποίες ενισχύουν τη διαδικασία ελέγχου και καταγραφής των logs:

- ο έλεγχος και η καταγραφή σε όλα τα επίπεδα λειτουργίας της εφαρμογής
- η αναλυτική καταγραφή των κύριων συμβάντων
- η διασφάλιση και προστασία των αρχείων καταγραφής
- η περιοδική δημιουργία αντιγράφων και ανάλυση των αρχείων καταγραφής **(Μαυρίδης, 2015)**.

Σε αυτό το σημείο είναι εμφανές ότι η αποτελεσματικότητα των μέτρων, τα οποία εξετάστηκαν παραπάνω, εξαρτάται σε σημαντικό βαθμό από τη σωστή χρήση τους από τους προγραμματιστές – διαχειριστές των εφαρμογών. Οι πιο σημαντικοί παράγοντες, οι οποίοι επηρεάζουν την αποτελεσματικότητα των αντίμετρων, είναι:

- η επίγνωση του μεγέθους του προβλήματος, καθώς τα άτομα τα οποία είναι υπεύθυνα για την εφαρμογή τους θα πρέπει να έχουν πειστεί για τη χρησιμότητα τους.
- περιοδικές αναθεωρήσεις, καθώς με τις σύγχρονες συνθήκες οι ανάγκες και οι απειλές εξελίσσονται συνεχώς.
- αλληλοεπικάλυψη μέτρων - φιλοσοφία του "ασθενέστερου σημείου" (weakest point philosophy). Η ασφάλεια ενός συστήματος είναι δυνατόν να προσομοιαστεί με μία αλυσίδα, όπου η ισχύς της είναι ισοδύναμη με την ισχύ του ασθενέστερου κρίκου της. Αυτό σημαίνει ότι δεν πρέπει να μένουν απροστάτευτα σημεία, καθώς ενδεχομένως να γίνουν στόχοι επιθέσεων. οι πιθανότητες χρησιμοποίησης. Σύμφωνα με την "Αρχή της Αποτελεσματικότητας" τα μέτρα ασφάλειας θα πρέπει να είναι επαρκή, κατάλληλα και εύκολα στη χρήση τους με στόχο να είναι αποτελεσματικά **(Πάγκαλος & Μαυρίδης, 2002)**.




**Εικόνα 5. Μία τυπική αρχιτεκτονική μίας διαδικτυακής εφαρμογής συσχετιζόμενη με τα επιμέρους ζητήματα ασφάλειας διαδικτυακής εφαρμογής (πηγή: Μαυρίδης, 2015)**

# Κεφάλαιο 3. Ασφάλεια προσωπικών δεδομένων σε δημοφιλείς διαδικτυακές εφαρμογές

## 3.1. Εισαγωγή στα κοινωνικά δίκτυα

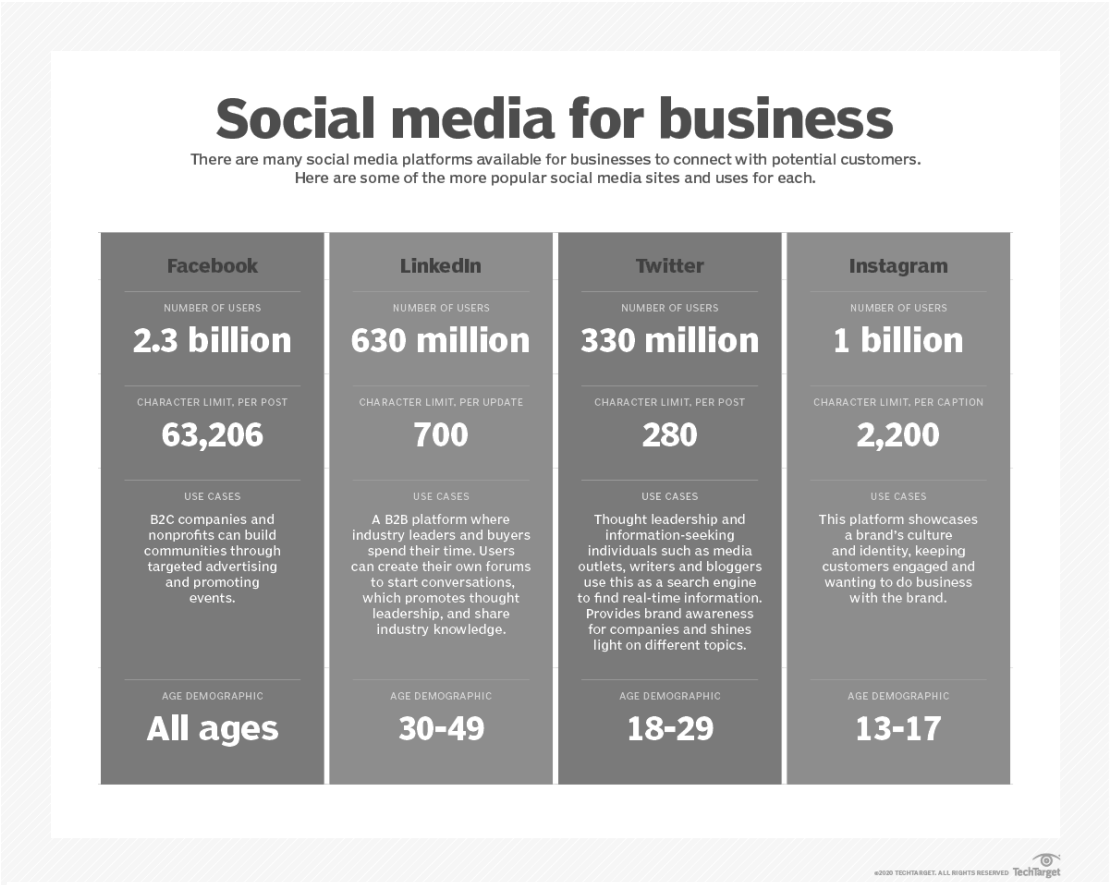
Τη σύγχρονη εποχή οι υπηρεσίες κοινωνικής δικτύωσης (social networks) έχουν εγκατασταθεί στη ζωή εκατομμυρίων χρηστών. Η πανευρωπαϊκή έρευνα, η οποία πραγματοποιήθηκε από την Internet & Mobile MC DC, επιβεβαιώνει την παραπάνω δήλωση, όπου σύμφωνα με τα αποτελέσματα της το 63% του πληθυσμού έχει δημιουργήσει προφίλ σε κάποια υπηρεσία κοινωνικής δικτύωσης, ενώ ο ευρωπαϊκός μέσος όρος βρίσκεται στο 45%, ποσοστό το οποίο κατατάσσει την Ελλάδα στην 3η θέση **(IAB HELLAS, 2009)**.

Ωστόσο, είναι ιδιαίτερα σημαντική η συμβολή των υπηρεσιών κοινωνικής δικτύωσης στα άτομα με αναπηρία (ΑμεΑ), στα οποία παρέχονται ίσες ευκαιρίες κοινωνικοποίησης, εκπαίδευσης και ψυχαγωγίας.

Τα social networks και τα social media είναι δύο έννοιες, οι οποίες πολλές φορές συγχέονται· παρουσιάζουν όμως σημαντικές διαφορές. Στα social networks συμπεριλαμβάνονται τα social media, καθώς οι επιχειρήσεις κάνουν χρήση των υπαρχόντων υπηρεσιών κοινωνικής δικτύωσης, δημιουργώντας το προφίλ της επιχείρησης τους, ως ένα εργαλείο προβολής ψηφιακού περιεχομένου στους δισεκατομμύρια χρήστες των κοινωνικών δικτύων (π.χ. Facebook, Instagram – ***Εικόνα 6***).

Επιπρόσθετα, η αμεσότητα με την οποία πραγματοποιείται η επικοινωνία μέσα των κοινωνικών δικτύων, όπως είναι το Facebook, Instagram, Twitter, LinkedIn κ.α., δίνει την αίσθηση ικανοποίησης στους χρήστες και έχει αποκτήσει διάφορες μορφές. Εκτός από την παραδοσιακή αποστολή μηνυμάτων, φωτογραφιών και βίντεο δίνεται πλέον η δυνατότητα πλήθους δραστηριοτήτων,

όπως είναι η παρακολούθηση live μεταδόσεων, η διαφήμιση προϊόντων, ηλεκτρονικές πωλήσεις κ.α.



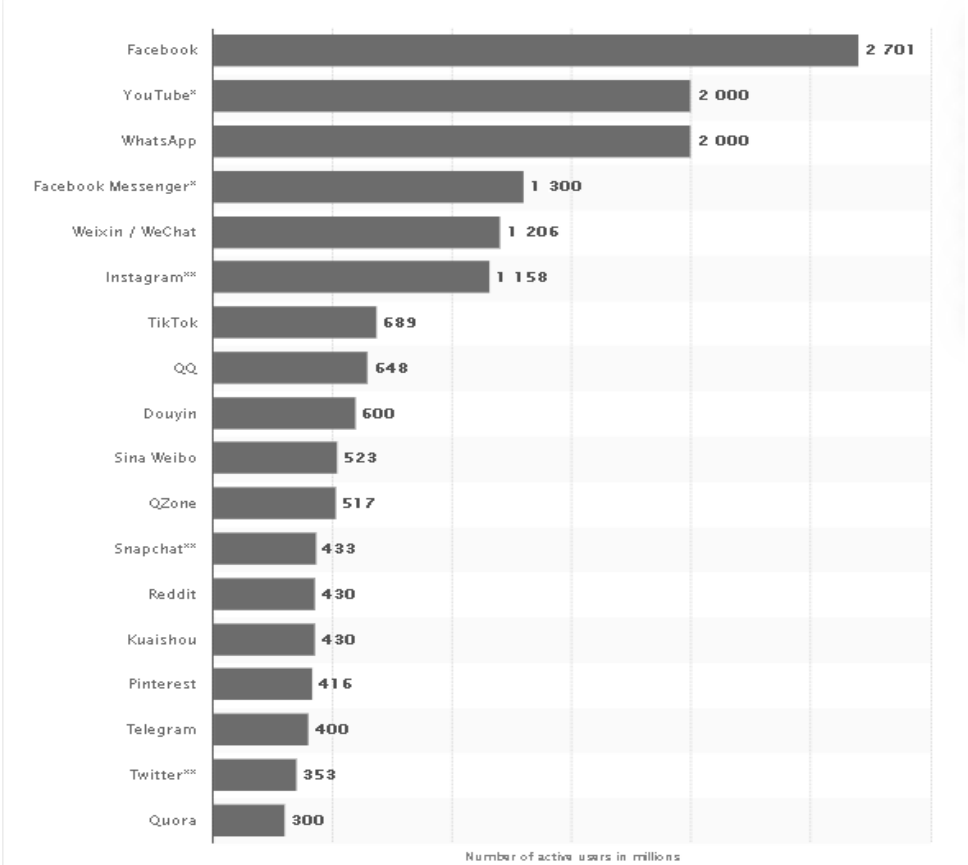

**Εικόνα 6. Businesses can connect to customers through social media networks. (πηγή: SearchUnifiedCommunications)**

### 3.1.1. Θέματα ασφάλειας κοινωνικών δικτύων

Εκτός από τα οφέλη της κοινωνικοποίησης του ατόμου και της εποικοδομητικής ανταλλαγής απόψεων με άτομα από κάθε σημείο της γης, η απρόσκοπτη χρήση των κοινωνικών δικτύων ελλοχεύει μία σειρά από κινδύνους. Ένα κοινό παράδειγμα αποτελεί η πλαστοπροσωπία, η οποία σε συνδυασμό με την εξαπάτηση των χρηστών (π.χ. Ηλεκτρονικό Ψάρεμα - Phising) πλήττει την εμπιστοσύνη των χρηστών και την ιδιωτικότητα τους και έχει ως σκοπό την αθέμιτη απόκτηση ευαίσθητων προσωπικών δεδομένων ή passwords. Αυτή η

απειλή ανήκει στην κατηγορία κινδύνου απάτης και κλοπής ταυτότητας (Risk of Fraudand Identity Theft).

Οι περισσότεροι χρήστες δείχνουν “τυφλή” εμπιστοσύνη στην κοινότητα του Facebook κάνοντας χρήση του πραγματικού τους ονόματος, διεύθυνσης, ημερομηνίας γέννησης, φωτογραφίες, βίντεο κ.α., χωρίς προηγουμένως να έχουν μελετήσει τους Όρους Χρήσης, όταν προχώρησαν στην εγγραφή τους. Είναι σημαντικό να σημειωθεί, όμως, ότι οι Όροι είναι δυνατόν να αλλάξουν ανά πάσα στιγμή· ότι οι εφαρμογές κοινωνικής δικτύωσης δεν εγγυώνται για την ασφάλεια τους και ότι οι χρήστες αποποιούνται των πνευματικών τους δικαιωμάτων κατά την αποδοχή των Όρων Χρήσης της εφαρμογής. Το τελευταίο συνεπάγεται τη χορήγηση άδειας στο Facebook να χρησιμοποιήσει, αντιγράψει, αποδώσει δημόσια, επιδείξει δημόσια, επαναφορμάρει και μεταφράσει απόσπασμα (γενικά ή εν μέρει) και να διανείμει το περιεχόμενο χρηστών για οποιοδήποτε σκοπό σχετικά με τον ιστοχώρο ή την προώθηση του.




**Εικόνα 7. Η προτίμηση των χρηστών σε social networks εκφρασμένη σε εκατ.**
**(πηγή:** https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/**)**

Εν συνεχεία, είναι δύσκολη η διαγραφή του ηλεκτρονικού ίχνους, το οποίο μένει ακόμη και μετά τη διαγραφή ενός λογαριασμού στο Facebook, καθώς θα υπάρχει κάποια φωτογραφία, ετικέτα ή κάποια αναφορά του ονόματος του σε λογαριασμό κάποιου άλλου χρήστη του Facebook, με τον οποίο ήταν προηγουμένως συνδεδεμένος. Υπάρχει επίσης και η απενεργοποίηση λογαριασμού, όπου ο χρήστης κάνοντας σύνδεση με τα στοιχεία του είναι σε θέση να τον ενεργοποιήσει ξανά όποτε επιθυμεί.

**Διαδικτυακές απάτες**

Εκτός από την περίπτωση του ηλεκτρονικού ψαρέματος, η οποία αναφέρθηκε παραπάνω, έχει παρατηρηθεί η αύξηση επιθέσεων με τη χρήση κακόβουλου λογισμικού σχεδιασμένου να αποσπά τα ψηφιακά δεδομένα χρηστών, το οποίο συναντάται ως κλέφτης κωδικών πρόσβασης PSW (**P**assword **S**tealing **W**are). Το 2019 σημειώθηκαν επιθέσεις σε πάνω από 940.000 χρήστες, με τους ειδικούς να εφιστούν την προσοχή μας. Για την αντιμετώπιση τους είναι απαραίτητη η χρήση ενημερωμένου λογισμικού antivirus.

Μία ακόμα περίπτωση επίθεσης εναντίον ανυποψίαστων χρηστών είναι η αποστολή spam στο Twitter ή στο Facebook καθώς και η δημιουργία σχολίων, τα οποία ανακατευθύνουν τον χρήστη σε κακόβουλα sites. Επιλέγονται τα πιο δημοφιλή θέματα και προστίθενται σύνδεσμοι, οι οποίοι παραπέμπουν σε κακόβουλο λογισμικό. Συνεπώς, οι χρήστες θα πρέπει να είναι ιδιαίτερα υποψιασμένοι όταν συναντούν κακογραμμένα από ξένη μετάφραση σχόλια ή μηνύματα και μη ασφαλείς συνδέσμους άγνωστης προέλευσης καθώς και να τα αποφεύγουν.

Ακόμα, μία μορφή απάτης είναι η πρόταση ευκαιριών απασχόλησης σε μη υπαρκτές θέσεις εργασίας. Ο χρήστης θα πρέπει να επαληθεύσει την ύπαρξη της αντίστοιχης θέσης εργασίας προτού προβεί σε επικοινωνία. Παρόμοιες απάτες εμφανίζονται με τη μορφή εκπαιδευτικής ευκαιρίας. Εμφανίζονται επίσης και με την υπόσχεση παροχής υπηρεσιών: αναφέρονται σε προπληρωμένες

υπηρεσίες, οι οποίες μετά την πληρωμή δεν παρέχονται στον χρήστη, όπως παραδείγματος χάρη πωλήσεις αγαθών.

Εν συνεχεία, μία παρόμοια συνήθης πρακτική είναι η εξαπάτηση με ψευδή επιχειρηματικά σχέδια και χειραγώγηση της αγοράς: η πιο συνήθης μορφή είναι η παραπληροφόρηση σχετικά με τη χρηματιστηριακή αγορά και την αξία μετοχών **(Λαζακίδου et al., 2004)**.

Εκτός των παραπάνω, είναι σημαντικό να αναφερθούν και οι περιπτώσεις Κυβερνο-εκφοβισμού και Εγκλήματος (Cyber - Bulling and Crime), οι οποίες έχουν πολύ σημαντικές επιπτώσεις στον ψυχισμό του θύματος. Οι επιτιθέμενοι κάνουν χρήση των πολλαπλών δυνατοτήτων των κοινωνικών δικτύων με τη δημοσίευση κακοπροαίρετων σχολίων και αναρτήσεων υβριστικών μηνυμάτων καθώς και με την εκμετάλλευση των προσωπικών δεδομένων των χρηστών, όπως φωτογραφίες ή βίντεο.

Τέλος, σε αυτό το σημείο θα πρέπει να σημειωθεί ότι κατά την έρευνα δεν έχουν βρεθεί εις βάθος τεκμηριωμένα στοιχεία σχετικά με τις τεχνολογίες ασφάλειας, τις οποίες χρησιμοποιούν οι δημοφιλείς εφαρμογές κοινωνικών δικτύων, όπως είναι το Facebook, το Instagram κ.α. Εν αντιθέσει, υπάρχουν πληροφορίες μόνο σε σχέση με το επίπεδο ασφάλειας χρήστη του εκάστοτε κοινωνικού δικτύου. Αυτό είναι απόλυτα κατανοητό, εάν αναλογιστεί κάποιος το πόσο συχνά γίνονται στόχος κυβερνο-επιθέσεων. Καθώς αποκαλύπτοντας τις τεχνικές τους, θα καταστεί ευάλωτο το σύστημα τους σε μελλοντικές επιθέσεις.

### 3.1.1.1. Η ασφάλεια των προσωπικών δεδομένων στο Facebook

Στις 17 Μαρτίου του 2018 έγινε λόγος για ένα από τα μεγαλύτερα διαδικτυακά σκάνδαλα της εποχής από την Cambridge Analytica, το οποίο αφορούσε την απόσπαση πληροφοριών 87 εκατομμύριων χρηστών του Facebook, οι οποίοι είχαν επιτρέψει την πρόσβαση σε εφαρμογή 3ου μέρους, ονομαζόμενη ως *"This is your digital life".* Αυτή η παραβίαση δεδομένων αποδείχθηκε ότι αφορούσε μία αδυναμία στο Facebook API. Με αφορμή το παραπάνω, οι χρήστες άρχισαν να αμφισβητούν την ασφάλεια του Facebook και να αποχωρούν διαγράφοντας τον λογαριασμό τους. Εν συνεχεία, εξαιτίας της ανασφάλειας, η οποία επικράτησε στη συνέχεια, επηρεάστηκε σημαντικά η εμπιστοσύνη των χρηστών απέναντι σε όλα τα κοινωνικά δίκτυα. Έτσι, δόθηκε η αφορμή για την περαιτέρω ενίσχυση της ασφάλειας του Facebook καθώς και για την ενημέρωση των χρηστών σχετικά με τους κινδύνους ασφάλειας.

Το Facebook, λαμβάνοντας υπόψιν το πλήθος κινδύνων ασφάλειας, όπως αυτών που αναφέρθηκαν παραπάνω, καθώς και την άγνοια κινδύνου των χρηστών, έχει καθιερώσει μία σειρά ρυθμίσεων ασφάλειας και απορρήτου, με στόχο να διασφαλίσει την ακεραιότητα των πληροφοριών.

Συγκεκριμένα, το Facebook αναφέρει στις βασικές ρυθμίσεις απορρήτου ότι με τα ειδικά εργαλεία παρακολούθησης δικτύου και κρυπτογράφησης, τα οποία έχει στη διάθεση του, εγγυάται για την ασφάλεια των προσωπικών δεδομένων των χρηστών [2]. Ωστόσο, δεν υπάρχουν περισσότερες πληροφορίες σε σχέση με το ποια είναι αυτά τα ειδικά εργαλεία ασφάλειας, όπως αναφέρθηκε παραπάνω.

Επίσης, με τον Έλεγχο Ασφάλειας [3] εκπαιδεύει τον χρήστη σε σχέση με το πως να προστατεύει τις προσωπικές πληροφορίες του. Ο έλεγχος ασφάλειας έχει

[2] **Τα προσωπικά δεδομένα σας (**https://www.facebook.com/help/330229433729799/**)**

[3] **Τι είναι ο έλεγχος ασφάλειας; (**https://www.facebook.com/help/android-app/799880743466869?ref=related**)**

σχεδιαστεί με στόχο να είναι φιλικός προς τον χρήστη και σε γλώσσα απλή και κατανοητή.

Μέσω αυτού, ο χρήστης είναι σε θέση να λάβει άμεσα προειδοποιητικό E-mail, όταν κάποιος προσπαθήσει να συνδεθεί στον λογαριασμό του από νέα IP (**I**nternet **P**rotocol address), να αποσυνδεθεί από εφαρμογές 3$^{ου}$ μέρους και προγράμματα περιήγησης και τέλος, να ενημερωθεί για την προστασία του κωδικού πρόσβασης, τον οποίο διαθέτει. Επιπροσθέτως, προτρέπει τους χρήστες να αναφέρουν κάθε ύποπτη κίνηση εντός της ιστοσελίδας και να καταγγέλλουν την υποψία παραβίασης του λογαριασμού τους.

Μερικές συμβουλές για τους χρήστες του Facebook είναι:

- να αναφέρουν ευαίσθητες ή προσβλητικές φωτογραφίες και βίντεο, ενώ όσον αφορά στους χρήστες - πομπούς του εν λόγω περιεχομένου να μπλοκάρονται και να αναφέρονται στους διαχειριστές του Facebook.
- να αφαιρούν προσβλητικά σχόλια σε φωτογραφίες ή βίντεο, τα οποία συναντώνται στον λογαριασμό τους.
- να αναφέρουν μέσω του *Κέντρου Βοήθειας* φωτογραφίες, βίντεο, σχόλια ή ακόμα και λογαριασμούς που έχουν δημιουργηθεί με σκοπό τον εκφοβισμό ή την παρενόχληση.
- να αναφέρουν και να μπλοκάρουν τους χρήστες – πομπούς προσβλητικών μηνυμάτων.
- να αναφέρουν περιπτώσεις ψευδών ειδήσεων, προσβολής θρησκευτικών πεποιθήσεων, βίας, χρήσης όπλων κ.α.
- να επιλέγουν τις κατάλληλες ρυθμίσεις απορρήτου σε δημοσιευμένες φωτογραφίες ή βίντεο. Δίνεται η δυνατότητα στους χρήστες να επιλέξουν σε ποιον ή σε ποια ομάδα χρηστών θα είναι ορατό το περιεχόμενο τους, παραδείγματος χάρη φίλοι, φίλοι φίλων, δημόσια ή ακόμα και σε προσαρμοσμένες λίστες φίλων κ.α.

Τέλος, το Facebook διαθέτει ορισμένες δικλίδες ασφάλειας, όταν κάποιος χρήστης επιθυμεί να ανακτήσει τον έλεγχο του λογαριασμού του σε περίπτωση απώλειας των στοιχείων σύνδεσης. Η ανάκτηση του ελέγχου πραγματοποιείται είτε μέσω E-mail ανάκτησης, το οποίο έχει δηλώσει ο χρήστης στον λογαριασμό του, είτε μέσω μίας ερώτησης ασφάλειας, η οποία έχει απαντηθεί κατά τη δημιουργία του λογαριασμού. Δίνεται ακόμα η δυνατότητα ανάκτησης του ελέγχου και μέσω των έμπιστων επαφών ενός χρήστη.

### 3.1.1.2. Η ασφάλεια των προσωπικών δεδομένων στο Instagram

Το Instagram, ως μία δημοφιλής εφαρμογή και μέσο κοινωνικής δικτύωσης, δίνει τη δυνατότητα σε εκατομμύρια χρήστες να επικοινωνούν μεταξύ τους, μέσω της κοινοποίησης φωτογραφιών, βίντεο και ιστοριών - δηλαδή φωτογραφιών ή βίντεο, τα οποία "μένουν" online για 24 ώρες. Οι χρήστες έχουν τη δυνατότητα να μοιράζονται φωτογραφίες και βίντεο με τους ακόλουθους τους ή με μία περιορισμένη ομάδα ατόμων της επιλογής τους. Ωστόσο, δεν λείπουν οι φορές, όποτε κακόβουλοι χρήστες χρησιμοποιούν τις υπηρεσίες του Instagram για να κατεβάσουν φωτογραφίες ή βίντεο ανυποψίαστων χρηστών του και να τα χρησιμοποιήσουν για δικό τους όφελος, προσβάλλοντας την ιδιωτικότητα τους. Έτσι, η ανάγκη προστασίας των προσωπικών δεδομένων των χρηστών κρίνεται περισσότερο από ποτέ επιτακτική.

Σε αυτό το σημείο θα πρέπει να επισημανθεί ότι:

- Κάθε προφίλ στο Instagram είναι δημόσιο προς όλους, εκτός από την περίπτωση όπου το προφίλ του χρήστη έχει μετατραπεί σε ιδιωτικό. Αυτή η ρύθμιση αυτόματα συνεπάγεται ότι οποιοσδήποτε επιθυμεί να αποκτήσει πρόσβαση στις δημοσιεύσεις κάποιου άλλου χρήστη, θα πρέπει να το έχει ζητήσει εκ των προτέρων με σχετικό αίτημα (Follow).
- Μέσω της χρήσης των #hashtags οι φωτογραφίες και τα βίντεο κατηγοριοποιούνται ανάλογα με το περιεχόμενο τους και είναι ορατά στη σελίδα των hashtags (δημόσια).
- Προτείνεται να αναφέρονται ευαίσθητες ή προσβλητικές φωτογραφίες και βίντεο, ενώ όσον αφορά στους χρήστες - πομπούς του εν λόγω περιεχομένου θα πρέπει να μπλοκάρονται και να αναφέρονται στους διαχειριστές.
- Προσβλητικά σχόλια σε φωτογραφίες ή σε βίντεο είναι δυνατόν να αφαιρεθούν από τον κάτοχο του λογαριασμού, στον οποίο συναντώνται.
- Φωτογραφία, βίντεο, σχόλιο ή ακόμα και λογαριασμός, ο οποίος έχει δημιουργηθεί με σκοπό τον εκφοβισμό ή την παρενόχληση, είναι δυνατόν να αναφερθεί μέσω του *Κέντρου Βοήθειας*.

- Σε περίπτωση που κάποιος άλλος χρήστης στέλνει προσβλητικά μηνύματα πρέπει να αναφέρεται στο Instagram και να μπλοκάρεται από τον χρήστη.
- Σε περίπτωση που κάποιος "υποδύεται" κάποιον άλλον χρήστη μέσω λογαριασμού, τον οποίο έχει δημιουργήσει στο Instagram, πρέπει να αναφέρεται στους διαχειριστές, κάνοντας κλικ στο *Κέντρο Βοήθειας* του Instagram, μετέπειτα στο *Κέντρο Απορρήτου και Ασφάλειας*, επιλέγοντας την ενότητα *"Αναφέρετε κάτι"* και έπειτα την επιλογή *"Impersonation Accounts"*. [4] [5]

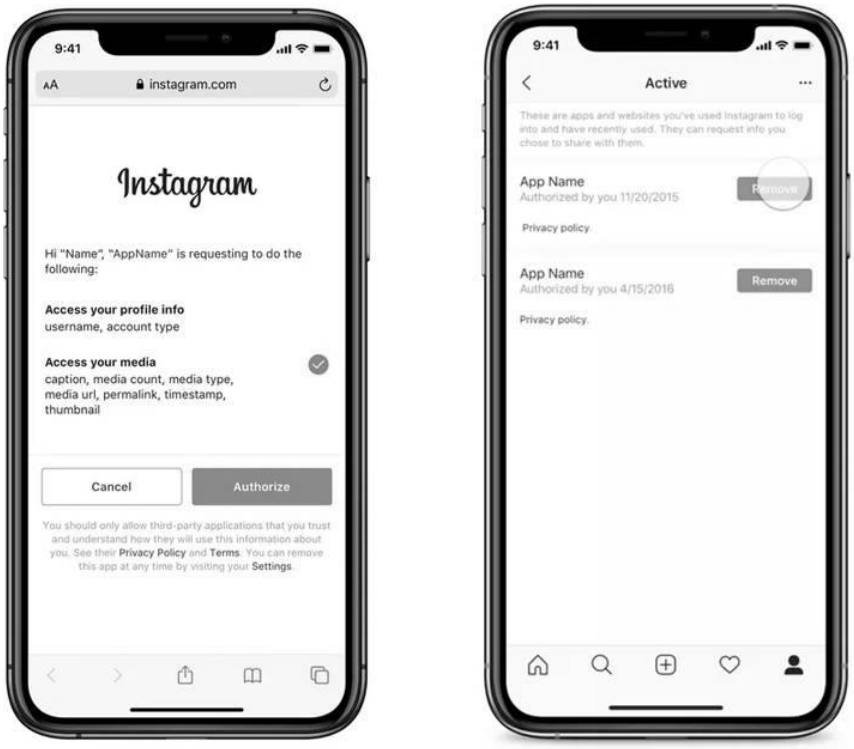
Μία ακόμα προσθήκη, η οποία έγινε πρόσφατα αντιληπτή στην εφαρμογή είναι ένα νέο χαρακτηριστικό, με το οποίο οι χρήστες έχουν τη δυνατότητα να διαχειρίζονται τα ευαίσθητα δεδομένα τους, τα οποία μοιράζονται σε τρίτες εφαρμογές μέσω του Instagram.

Κάνοντας πλοήγηση στις *Ρυθμίσεις* ο χρήστης έχει τη δυνατότητα να δει ποιες συνεργαζόμενες τρίτες υπηρεσίες έχουν πρόσβαση στα προσωπικά δεδομένα του, όπως όνομα, τηλέφωνο, πληροφορίες προφίλ, φωτογραφίες, βίντεο κ.α.

[4] **Πηγή:** https://saferinternet4kids.gr/

[5] **Βοήθεια για την ασφάλεια στο Instagram:**
**(**https://www.facebook.com/help/instagram/1372599552763476**)**

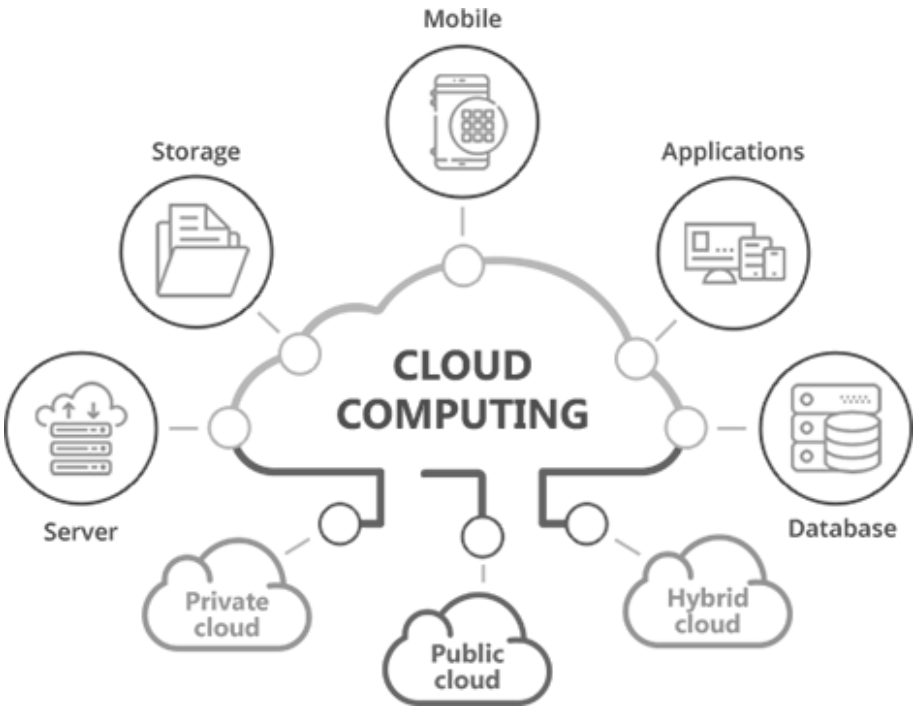
Η διαδικασία είναι η ακόλουθη: Από το προφίλ του χρήστη κάνοντας κλικ στο δεξί μέρος της οθόνης στο hamburger μενού, έπειτα στις *Ρυθμίσεις –> Ασφάλεια –> Εφαρμογές* και *Ιστότοποι*. Έτσι, ο χρήστης είναι σε θέση να επιτρέψει ή όχι την πρόσβαση στα προσωπικά στοιχεία του από εφαρμογές τρίτων.




**Εικόνα 8. Άρση δικαιωμάτων εφαρμογών τρίτων (πηγή: https://www.facebook.com/help/instagram/?rdrhc)**

## 3.2. Εισαγωγή στις εφαρμογές Cloud

Στο μοντέλο παροχής υπηρεσιών νέφους όλες οι υπηρεσίες είναι διαθέσιμες μέσω διαδικτύου. Οι υπηρεσίες νέφους παρέχονται με τέτοιο τρόπο, ώστε ο τελικός χρήστης δεν είναι σε θέση να διακρίνει τεχνικές λεπτομέρειες, όπως παραδείγματος χάρη η αρχιτεκτονική τους. Η χρηστικότητα, η διαθεσιμότητα και η αποτελεσματικότητα των υπηρεσιών αυτών αποτελούν τα κύρια κριτήρια επιλογής τους. Μερικές από τις μεγαλύτερες εταιρίες παγκοσμίως όπως η Google, η Amazon και η Microsoft έχουν υιοθετήσει τις τεχνολογίες Cloud. Η φιλοσοφία πίσω από την ανάπτυξη του Cloud ήταν το ερώτημα "πώς θα ήταν δυνατό οι εφαρμογές, τις οποίες χρησιμοποιούμε, να μας ακολουθούν όπου κι αν βρισκόμαστε;"




**Εικόνα 9. Υπηρεσίες Νέφους (πηγή: Ewere Diagboya, 2019)**

Κάποια από τα πιο σημαντικά **χαρακτηριστικά**, από τα οποία διακρίνονται οι υπηρεσίες νέφους, παρουσιάζονται παρακάτω:

**Εξυπηρέτηση κατ' απαίτηση**: άμεσα και χωρίς καθυστέρηση ο χρήστης έχει τη δυνατότητα να χρησιμοποιήσει την υπηρεσία, όποτε επιθυμεί, χωρίς να απαιτείται ανθρώπινη παρέμβαση.

**Ευρεία πρόσβαση στο δίκτυο**: οι δυνατότητες είναι διαθέσιμες σε όλο το δίκτυο και είναι προσβάσιμες μέσα από σταθερούς μηχανισμούς και από οποιαδήποτε συνδεδεμένη συσκευή.

**Διαθεσιμότητα πόρων**: η υπηρεσία χρησιμοποιεί πόρους, όπως υπολογιστικό χρόνο και αποθηκευτικό χώρο, τα οποία μοιράζονται σε πολλούς χρήστες.

**Γρήγορη ευελιξία**: υπάρχει η δυνατότητα η υπηρεσία να κλιμακωθεί γρήγορα χωρίς πρόβλημα και να αντιμετωπίσει περιόδους ιδιαίτερα αυξημένου φόρτου.

**Υπηρεσία μέτρησης**: η υπηρεσία καταγράφει τη χρήση και βάσει της τελευταίας γίνεται χρέωση ή βελτιστοποίηση της υπηρεσίας.

Οι υπηρεσίες νέφους παρέχονται σε τρία μοντέλα, καθένα από τα οποία είναι προσανατολισμένο σε ειδικές κατηγορίες χρηστών.

**Υποδομές ως υπηρεσία (Infrastructure as a service – IaaS)**: αφορά στην παροχή υπολογιστικών πόρων και απευθύνεται σε ειδικούς διαχείρισης δικτύων και υπολογιστικών συστημάτων.

**Πλατφόρμα ως υπηρεσία (Platform as a service – PaaS)**: αφορά στην παροχή υπολογιστικών πλατφορμών και απευθύνεται σε προγραμματιστές.

**Λογισμικό ως υπηρεσία (Software as a service – SaaS)**: αφορά στην παροχή εφαρμογών στους τελικούς χρήστες και απευθύνεται σε όλους τους χρήστες **(Πανσεληνάς et al., 2015)**.

### 3.2.1. Θέματα ασφάλειας σε εφαρμογές Cloud

Οι υπηρεσίες υπολογιστικού νέφους, σε όλο το φάσμα τους, αναμφισβήτητα παρέχουν σημαντικά οφέλη στις επιχειρήσεις, οι οποίες κάνουν χρήση των υπηρεσιών του, χάρη στο χαμηλό κόστους τους και στα νέα ευέλικτα επιχειρηματικά μοντέλα**.** Ωστόσο, θα έπρεπε σε αυτό το σημείο να σημειωθούν και οι ανησυχίες, οι οποίες προκύπτουν, σχετικά με την ασφάλεια των δεδομένων, τα οποία μεταφέρονται στο Cloud. "Η ασφάλεια και η διατήρηση των δεδομένων αποτελεί πρωταρχικό προβληματισμό" **(Pandith, 2014)**.

Τα πιο σημαντικά σημεία του Cloud, στα οποία απαιτείται ιδιαίτερη προσοχή, είναι:

- η ασφάλεια των δεδομένων σε αδράνεια
- η ασφάλεια των δεδομένων κατά τη διαβίβαση
- η επαλήθευση χρηστών, εφαρμογών, διαδικασιών
- ο ισχυρός διαχωρισμός μεταξύ δεδομένων διαφορετικών χρηστών
- τα νομικά και ρυθμιστικά θέματα του νέφους
- η διαχείριση προβλημάτων.

Γενικά, σε σχέση με τα δεδομένα σε αδράνεια (data in rest), τα οποία χρησιμοποιούνται από μία εφαρμογή βασισμένη σε cloud, ισχύει ότι δεν είναι κατά κανόνα κρυπτογραφημένα, καθώς η κρυπτογράφηση θα απέτρεπε την ευρετηρίαση ή την αναζήτηση τους. Τα δεδομένα αυτά είναι αποθηκευμένα σε αποθήκες δεδομένων μαζί με δεδομένα άλλων χρηστών.

Αν και χρησιμοποιούνται τεχνικές ασφάλειας, όπως είναι οι ετικέτες δεδομένων, με στόχο την παρεμπόδιση μη εξουσιοδοτημένης πρόσβασης σε αναμειγμένα δεδομένα, τα τελευταία εξαιτίας της φύσης τους δεν αποκλείεται να παραβιαστούν από "hackers". Παραδείγματος χάρη, κάτι αντίστοιχο συνέβη τον Μάρτιο του 2009 με τη διαρροή μη εξουσιοδοτημένων δεδομένων μεταξύ

των χρηστών των εφαρμογών Documents και Spreadsheets της Google. Συγκεκριμένα, έως τον Ιούνιο του 2009 δεν είχε αναπτυχθεί κάποια τεχνική ολικής επεξεργασίας κρυπτογραφημένων δεδομένων.

Η κρυπτογράφηση των δεδομένων και η χρήση ενός ακαταστατικού πρωτοκόλλου (π.χ. HTTP) παρέχουν "ανωνυμότητα" στον χρήστη· τα παραπάνω όμως δεν είναι ικανά να διασφαλίσουν την ακεραιότητα των δεδομένων. Κατά τη χρήση μίας IaaS υπηρεσίας cloud (σε δημόσιο ή και σε ιδιωτικό επίπεδο) για απλή αποθήκευση, η κρυπτογράφηση δεδομένων σε αδράνεια είναι δυνατή και προτείνεται. Αντίθετα, κατά τη χρήση μίας PaaS ή SaaS υπηρεσίας cloud (π.χ. Google Apps, Salesforce) η κρυπτογράφηση δεδομένων σε αδράνεια δεν είναι πάντα εφικτή. Επομένως, θεωρείται ότι τα δεδομένα θα κρυπτογραφηθούν τουλάχιστον σε κάποιο στάδιο του κύκλου ζωής τους, είτε είναι in-transit είτε είναι in-rest, κατά τη διάρκεια επεξεργασίας τους στο νέφος, εκτός κι αν τα δεδομένα βρίσκονται στο cloud μόνο για απλή αποθήκευση.

Όσον αφορά στην ικανοποιητική διαχείριση κλειδιών στο cloud, δεν υπάρχει εγγύηση για αυτήν, καθώς είναι πέραν των σύγχρονων δυνατοτήτων των εφαρμογών cloud και απαιτείται μελλοντική έρευνα στην κρυπτογράφηση και τη διαχείριση κλειδιών.

Εν συνεχεία, καθώς ο χρήστης ή μία επιχείρηση δεν είναι δυνατόν να γνωρίζει την τοποθεσία όπου βρίσκονται αποθηκευμένα τα δεδομένα του, υπάρχει νομοθεσία απορρήτου στα περισσότερα κράτη, σύμφωνα με την οποία προστατεύεται η μεταφορά προσωπικών δεδομένων σε άλλες χώρες. Επίσης, ο πάροχος υπηρεσιών νέφους οφείλει να ειδοποιεί τους χρήστες σε περίπτωση παραβίασης των δεδομένων και να τους ενημερώνει σχετικά με το κόστος, το οποίο συνεπάγεται, και την ανάληψη ευθύνης - εάν υπάρχει αντίστοιχη αναφορά στο συμβόλαιο.

Συνοψίζοντας, οι ανησυχίες για την ασφάλεια των δεδομένων στις εφαρμογές cloud δεν αναιρούν τα πλεονεκτήματα της αξιοποίησης της αποθήκευσης στο νέφος για μη ευαίσθητα ή μη ελεγχόμενα δεδομένα. Εάν οι χρήστες επιθυμούν

απλή αποθήκευση δεδομένων στο cloud, θα πρέπει τουλάχιστον να επαληθεύσουν ότι ο πάροχος θα παρέχει επαρκώς τις υπηρεσίες, οι οποίες απαιτούνται για την ασφαλή αποθήκευση των δεδομένων στο νέφος **(Pandith, 2014)**.

#### 3.2.1.1. Η ασφάλεια προσωπικών δεδομένων στο Dropbox

Το Dropbox είναι μία δημοφιλής εφαρμογή αποθηκευτικού χώρου ή πιο απλά μία υπηρεσία, η οποία επιτρέπει την αποθήκευση, τον συγχρονισμό και την κοινή χρήση αρχείων μεταξύ διαφορετικών συσκευών, τις οποίες ο χρήστης έχει επιλέξει να συνδέσει με τον λογαριασμό του. Η υπηρεσία προσφέρει από 2GB έως 16GB δωρεάν αποθηκευτικό χώρο, ενώ είναι διαθέσιμη σε διάφορα λειτουργικά συστήματα, όπως Windows, Mac, Linux, iPhone, iPad, Android και BlackBerry. Βασικοί ανταγωνιστές του Dropbox είναι το OneDrive της Microsoft και το Google Drive της Google. Συγκεκριμένα, το Dropbox προσφέρει αρχικά 2GB δωρεάν αποθηκευτικό χώρο, ενώ το OneDrive και το Google Drive προσφέρουν 5GB και 15GB αντίστοιχα **(Edtech, 2020)**.

Η απλή χρήση την οποία προσφέρει το Dropbox συνδυάζεται με ένα πολύ απλό πρόγραμμα, το οποίο εγκαθίσταται πολύ εύκολα στον τοπικό υπολογιστή. Το πρόγραμμα αυτό δημιουργεί έναν φάκελο στον τοπικό υπολογιστή, ο οποίος θα επικοινωνεί με τον εξυπηρετητή μέσω του διαδικτύου. Έτσι, μετακινώντας αρχεία ή φακέλους σε αυτόν τον φάκελο αυτόματα αυτά συγχρονίζονται με τον λογαριασμό, τον οποίο έχει ο χρήστης στο διαδίκτυο. Ο συγχρονισμός των αρχείων αντικατοπτρίζει την αντιγραφή αρχείων από έναν φάκελο σε έναν άλλον. Με την υπηρεσία του συγχρονισμού επομένως, το Dropbox μοιράζει τα αρχεία σε πολλούς υπολογιστές. Για πάνω από 3 συσκευές ο χρήστης θα πρέπει να αγοράσει το πιο ακριβό πακέτο. Τέλος, έχουν δημιουργηθεί εφαρμογές, οι οποίες επεκτείνουν τις δυνατότητες του Dropbox προσφέροντας τη μέγιστη αξιοποίηση του αποθηκευτικού χώρου **(Pcsteps, 2020)**.

Για μεγαλύτερη ασφάλεια ο χρήστης είναι σε θέση να ενεργοποιήσει την ταυτοποίηση 2 παραγόντων. Έτσι, το Dropbox θα απαιτεί έναν εξαψήφιο κωδικό ασφάλειας (εκτός από τον κωδικό πρόσβασης) κάθε φορά που ο χρήστης θα συνδέεται στον λογαριασμό του ή θα συνδέεται από έναν νέο υπολογιστή, τηλέφωνο ή tablet **(Dropbox, 2020)**.

### 3.2.1.2. Η ασφάλεια προσωπικών δεδομένων στο Google Drive

Το Google Drive αποτελεί μία υπηρεσία αποθήκευσης και συγχρονισμού αρχείων, η οποία παρέχεται από την Google και επιτρέπει τη χρήση αποθηκευτικού νέφους, τον διαμοιρασμό αρχείων και τη συνεργατική επεξεργασία από τον χρήστη. Τα αρχεία, τα οποία μοιράζονται δημόσια στο Google Drive, είναι ανακτήσιμα από τις μηχανές αναζήτησης. Επιπρόσθετα, το Google Drive περιλαμβάνει την υπηρεσία Google Docs, τη γνωστή σουίτα γραφείου με εφαρμογές παραγωγικότητας, η οποία προσφέρει τη συνεργατική επεξεργασία εγγράφων, υπολογιστικών φύλλων και παρουσιάσεων. Αρχικά, προσφέρει σε όλους τους χρήστες του έναν αρχικό online χώρο αποθήκευσης χωρητικότητας 15 GB ικανό να χρησιμοποιηθεί από τις διαδιδόμενες υπηρεσίες του Google Drive και του Gmail **(Wikipedia, 2020)**.

Αντί της παραδοσιακής λίστας ελέγχου για όλους τους χρήστες, ο Έλεγχος Ασφάλειας της Google είναι πλέον ένας εξατομικευμένος οδηγός σε σχέση με τη διασφάλιση των δεδομένων των χρηστών, αντικατοπτρίζοντας έναν προσωπικό σύμβουλο ασφάλειας. Όταν οι χρήστες επισκέπτονται τον Έλεγχο Ασφάλειας, ενημερώνονται αυτόματα σε σχέση με την κατάσταση ασφάλειας των δεδομένων τους **(PCMag, 2020)**. Επίσης, η Google δημιούργησε το *Πρόγραμμα Σύνθετης Προστασίας.*

Για περισσότερη ασφάλεια η Google διαθέτει τη ρύθμιση της Σύνθετης Προστασίας, η οποία διαθέτει μία σειρά από δικλίδες ασφάλειας με στόχο να προστατεύσει τους χρήστες από στοχευμένες επιθέσεις στο διαδίκτυο.

Συγκεκριμένα:

- αποτρέπει τη μη εξουσιοδοτημένη πρόσβαση στον λογαριασμό Google του χρήστη.
- παρέχει επιπλέον προστασία από επιβλαβείς λήψεις.
- για την παρεμπόδιση της μη εξουσιοδοτημένης πρόσβασης, επιτρέπει μόνο σε εφαρμογές Google και σε επαληθευμένες εφαρμογές τρίτου μέρους την πρόσβαση στα δεδομένα του λογαριασμού, κατόπιν αδειοδότησης από τον χρήστη.
- δεν επιτρέπει σε "εισβολείς" να προβούν σε κλοπή των στοιχείων της ταυτότητας του χρήστη με σκοπό να αποκτήσουν πρόσβαση στον λογαριασμό του. Εάν κάποιος προσπαθήσει να ανακτήσει τον λογαριασμό του, η Σύνθετη Προστασία θα ακολουθήσει κάποια επιπλέον βήματα με στόχο την επαλήθευση της ταυτότητας του **(Google, 2020)**.

### 3.2.1.3. Η ασφάλεια των προσωπικών δεδομένων στο OneDrive

Το Microsoft Office 365 είναι μία σουίτα εφαρμογών με μηνιαία συνδρομή. Για τους απλούς χρήστες περιλαμβάνει τη χρήση των Office εφαρμογών (Word, Excel, PowerPoint, Access, Outlook, OneNote), αποθηκευτικό χώρο στο OneDrive και 60 λεπτά ομιλίας στο Skype. Για τις εταιρίες προσφέρει εκτός αυτών, ηλεκτρονικό ταχυδρομείο, υπηρεσίες κοινωνικού δικτύου μέσω Skype και Exchange Server και ενσωμάτωση με το Yammer. Είναι διαθέσιμο στα λειτουργικά συστήματα Windows και OS X. **(Microsoft, 2020)**.

Η Microsoft συνιστά στους χρήστες της:

- να δημιουργούν έναν ισχυρό κωδικό πρόσβασης.
- να προσθέσουν πληροφορίες ασφάλειας στον λογαριασμό τους, όπως τον τηλεφωνικό αριθμό τους, μία εναλλακτική διεύθυνση ηλεκτρονικού ταχυδρομείου και μία ερώτηση ασφάλειας και την απάντηση της. Με αυτόν

τον τρόπο, το σύστημα χρησιμοποιεί τις πληροφορίες ασφάλειας σε σχέση με την επαλήθευση της ταυτότητας του χρήστη, σε περίπτωση που ξεχάσει τον κωδικό πρόσβασης.

- να χρησιμοποιούν την επαλήθευση δύο παραγόντων. Υπάρχει η δυνατότητα η επαλήθευση να πραγματοποιηθεί είτε με μία τηλεφωνική κλήση ή ένα μήνυμα κειμένου είτε μέσω μίας εφαρμογής.
- να ενεργοποιήσουν την κρυπτογράφηση στις κινητές συσκευές τους**.**
- να εγγραφούν στο Microsoft 365. Με τη συνδρομή Microsoft 365 παρέχεται προηγμένη προστασία από ιούς και παραβιάσεις του συστήματος, όπως επίσης και τρόποι ανάκτησης αρχείων μετά από κακόβουλες επιθέσεις.

Προκειμένου να μειώσουν την πιθανότητα παραβίασης προσωπικών δεδομένων και να εντοπίσουν και να μετριάσουν σε σύντομο χρονικό διάστημα τις συνέπειες της παραβίασης, το OneDrive και το Office 365 ακολουθούν πολιτικές όπως:

- έλεγχος πρόσβασης και τήρηση της Πολιτικής "μηδενικής πρόσβασης" για τους μηχανικούς, το οποίο σημαίνει ότι οι μηχανικοί δεν έχουν πρόσβαση στην υπηρεσία, εκτός κι αν έχει εκχωρηθεί ρητά ως απόκριση σε ένα συγκεκριμένο συμβάν, το οποίο απαιτεί προβιβασμό πρόσβασης **(Microsoft, 2020)**.
- έλεγχος πρόσβασης των μηχανικών με έξυπνες κάρτες και έλεγχο βιομετρικών χαρακτηριστικών.
- ισχυρά συστήματα παρακολούθησης ασφάλειας σε πραγματικό χρόνο.
- κρυπτογράφηση ασφάλειας επιπέδου μεταφοράς TLS (**T**ransport **L**ayer **S**ecurity) κατά τη μεταφορά δεδομένων.
- έλεγχος ταυτότητας μόνο με HTTPS (**H**yper**T**ext **T**ransform **P**rotocol **S**ecure).

Επιπρόσθετα, η Microsoft προβαίνει σε πρόσθετες ενέργειες όπως:

- ανίχνευση ιών κατά τη λήψη εγγράφων για γνωστές απειλές
- παρακολούθηση ύποπτης δραστηριότητας σε πραγματικό χρόνο

- ανάκτηση αρχείων πριν επηρεαστούν, έως και 30 ημέρες μετά από μία επίθεση
- ιστορικό εκδόσεων για όλους τους τύπους αρχείων
- προστασία με κωδικό πρόσβασης & λήξη συνδέσμων κοινής χρήσης
- δυνατότητα ανάκτησης μαζικών αρχείων – σε περίπτωση διαγραφής εκ παραδρομής μεγάλου όγκου αρχείων

Τέλος, οι μηχανικοί λογισμικού της Microsoft, οι οποίοι διαχειρίζονται το OneDrive, χρησιμοποιούν μία κονσόλα του Windows PowerShell, η οποία απαιτεί έλεγχο ταυτότητας δύο παραγόντων **(Microsoft, 2020)**.

## 3.3. Εισαγωγή στις εφαρμογές E-mail

Ένα απλό παράδειγμα υπολογιστικού νέφους είναι οι δημοφιλείς εφαρμογές ηλεκτρονικού ταχυδρομείου Yahoo, Gmail, Outlook κ.α. Το λογισμικό του παρόχου και του διαχειριστή των ηλεκτρονικών μηνυμάτων βρίσκεται μέσα σε αυτό το νέφος και ρυθμίζεται από τον αντίστοιχο πάροχο υπηρεσιών Yahoo, Google κ.λπ. [6]

Τα συστήματα ασφάλειας και οι τεχνολογίες, οι οποίες χρησιμοποιούνται από τον πάροχο υπηρεσιών, είναι κοινά για τις υπηρεσίες cloud, ηλεκτρονικού ταχυδρομείου κ.α. Έτσι, οι τεχνικές, οι οποίες έχουν αναφερθεί παραπάνω για τις εταιρείες Google και Microsoft συνεχίζουν να ισχύουν και για το ηλεκτρονικό ταχυδρομείο των παρόχων αυτών. Με βάση αυτό, έχει δοθεί έμφαση σε πιθανούς κινδύνους, οι οποίοι συναντώνται με τη χρήση του ηλεκτρονικού ταχυδρομείου, στις τεχνικές κρυπτογράφησης, οι οποίες εφαρμόζονται συγκεκριμένα σε αυτήν την περίπτωση, και τέλος, στις προτεινόμενες

[6] **Η χρήση του υπολογιστικού νέφους (Cloud Computing) στο ηλεκτρονικό επιχειρήν. (πηγή: http://businessescloudcomputing.blogspot.gr/p/cloud-computing-web.html)**

συμβουλές ασφάλειας των δύο παρόχων προς τους χρήστες των εφαρμογών αυτών.

### 3.3.1. Θέματα ασφάλειας υπηρεσιών E-mail

Πολύ συχνά ανυποψίαστοι χρήστες ηλεκτρονικού ταχυδρομείου γίνονται στόχος επίθεσης κυβερνο-εγκληματιών. Οι τρόποι επίθεσης τους περιλαμβάνουν τις παρακάτω κατηγορίες:

- τη μη εξουσιοδοτημένη χρήση του λογαριασμού ηλεκτρονικού ταχυδρομείου
- την προσβολή του υπολογιστή με ιούς, με σκοπό την κλοπή δεδομένων, ταυτότητας κ.α.

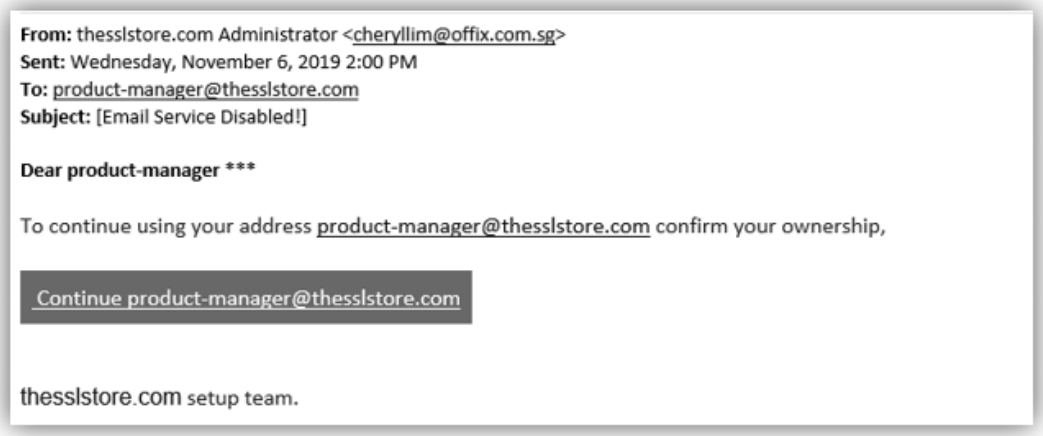
Η πρώτη κατηγορία αφορά στις απάτες μέσω E-mail, οι οποίες έχουν ως στόχο τον ανθρώπινο παράγοντα και προσβάλλουν την αρχή της αυθεντικότητας των μηνυμάτων. Συγκεκριμένα, υπολογίζεται ότι περίπου το **47% των κακόβουλων E-mails** στοχεύουν σε οικονομικούς συμβούλους εταιρειών – CFO. Ενώ, η δεύτερη κατηγορία αναφέρεται στα E-mails, τα οποία περιλαμβάνουν συνημμένα αρχεία ή συνδέσμους και προτρέπουν τον χρήστη να τα ανοίξει. Τα αρχεία αυτά περιέχουν κακόβουλο λογισμικό (malware), το οποίο εγκαθίστανται στον υπολογιστή του χρήστη και διευκολύνει τις ψηφιακές επιθέσεις.

Κάνοντας χρήση μίας τεχνικής, η οποία ονομάζεται **Spoofing**, οι επιτιθέμενοι εκμεταλλευόμενοι την άγνοια του παραλήπτη, προσπαθούν να κερδίσουν την εμπιστοσύνη του “υποδύοντας” κάποιον συνάδελφο, προϊστάμενο, προμηθευτή ή συνεργάτη του, τον οποίο θα εμπιστευόταν. Συχνά, ζητούν την εκτέλεση τραπεζικών εμβασμάτων ή άλλης μορφής ηλεκτρονικών συναλλαγών, μεταφορά περιουσιών, φορολογικά στοιχεία κ.α. Επίσης, είναι πιθανό να ζητούν στοιχεία σύνδεσης, στοιχεία τραπεζικών καρτών και άλλα διαπιστευτήρια.

Πριν την αποστολή του πλαστού E-mail, έχει αποδειχθεί ότι οι κυβερνο-εγκληματίες μελετούν προσεκτικά τα θύματα τους, τις εταιρείες όπου εργάζονται και την ιεραρχία αυτών μέσα από ιστοσελίδες κοινωνικής δικτύωσης, όπως το LinkedIn ή τον ιστότοπο της εταιρείας.

Το μυστικό επιτυχίας αυτών των παραβάσεων έγκειται στη χρήση E-mails, πανομοιότυπων με ένα αντίστοιχο επαγγελματικό. Προκειμένου να μην κινήσουν υποψίες ζητούν από τα ανυποψίαστα θύματα τους να εκτελέσουν καθήκοντα, τα οποία περιλαμβάνονται στην εργασία τους.


**Εικόνα 10. Παράδειγμα τεχνικής E-mail Spoofing (πηγή: Casey Crane, 2020)**




**Εικόνα 11. Παράδειγμα τεχνικής E-mail Spoofing με χρήση συνδέσμων (πηγή: Casey Crane, 2020)**

Μερικές από τις πιο συνήθεις μορφές εξαπάτησης περιγράφονται παρακάτω:

**Spoofed Name**: Αυτή η μορφή εξαπάτησης υπολογίζεται ότι αποτελεί το 75% των επιθέσεων. Χρησιμοποιεί ένα μη υπαρκτό όνομα ή οικείο στο θύμα στο πεδίο "From" **(*Εικόνα 11*)**. Ωστόσο, η διεύθυνση αποστολής του E-mail προέρχεται από άγνωστη υπηρεσία ηλεκτρονικού ταχυδρομείου, η οποία ανήκει στον επιτιθέμενο.

**Reply–To Spoofing**: Αυτή η τεχνική χρησιμοποιεί το πραγματικό όνομα και E-mail του θύματος. Ως όνομα στο "Reply-To" χρησιμοποιεί επίσης το όνομα του "μη υπαρκτού" αποστολέα του μηνύματος. Όμως, η διεύθυνση E-mail, από όπου αποστέλλονται οι απαντήσεις, δεν είναι η πραγματική και ανήκει στον επιτιθέμενο.

**Spoofed Sender (with No Reply–to Address)**: Αυτή η μορφή εξαπάτησης μέσω E-mail χρησιμοποιεί το όνομα και το E-mail του "μη υπαρκτού" ατόμου. Ωστόσο, το μήνυμα δεν περιλαμβάνει διεύθυνση αποστολής, οπότε δεν είναι δυνατή η απάντηση σε αυτό. Το μήνυμα συχνά περιλαμβάνει όλες τις οδηγίες σε σχέση με τη μεταφορά του τραπεζικού εμβάσματος ή της αποστολής ευαίσθητων προσωπικών δεδομένων κ.α.

**Lookalike Domain**: Με αυτήν την τεχνική μέσω E-mail, η διεύθυνση "From" του επιτιθέμενου είναι όμοια με αυτήν της διεύθυνσης E-mail του "μη υπαρκτού" ατόμου. Το domain ενδεχομένως να παρουσιάζει πολύ μικρές διαφορές από την πραγματική διεύθυνση E-mail **(Θωδωμάκης, 2018)**.

Με στόχο να εξασφαλισθεί ότι τα μηνύματα, τα οποία φτάνουν σε έναν χρήστη, είναι από έγκυρες διευθύνσεις, χρησιμοποιούνται τεχνολογίες αυθεντικοποίησης E-mail, οι οποίες ονομάζονται SPF (**S**ender **P**olicy **F**ramework) και DKIM (**D**omain **K**eys **I**dentified **M**ail). Επίσης, έχει αναπτυχθεί ένα ακόμα πιο εξελιγμένο εργαλείο ταυτοποίησης, το οποίο ονομάζεται DMARC (**D**omain based **M**essage **A**uthentication **R**eporting & **C**onformance) και ενισχύει την ασφάλεια, την οποία παρέχουν τα εργαλεία SPF και DKIM.

### 3.3.1.1. Ασφάλεια προσωπικών δεδομένων στο Gmail

Με στόχο την προστασία των υπηρεσιών E-mail της Google, χρησιμοποιούνται όλες οι τεχνικές ασφάλειας και κρυπτογράφησης μηνυμάτων, όπως έχουν περιγραφεί παραπάνω, και επιπρόσθετα χρησιμοποιείται:

- η Κρυπτογράφηση s/MIME (**M**ultipurpose **I**nternet **M**ail **E**xtensions) και
- η Κρυπτογράφηση TLS **(Google, 2020)**.

Το πρωτόκολλο S/MIME πρόκειται για μία εξειδίκευση του πρωτοκόλλου MIME, το οποίο χρησιμοποιείται για υποστήριξη ενισχυμένης κρυπτογράφησης κατά τη μεταφορά των μηνυμάτων. Κρυπτογραφεί αυτόματα, όπου είναι δυνατό, τα εξερχόμενα μηνύματα με ένα δημόσιο κλειδί και τα αποκρυπτογραφεί κατά τη λήψη με ιδιωτικό κλειδί, με στόχο την προστασία του απορρήτου του περιεχομένου του μηνύματος. Προκειμένου να είναι δυνατή η αποκρυπτογράφηση ενός μηνύματος πρέπει το κλειδί του χρήστη να μεταφορτωθεί με το μήνυμα κατά την παράδοση του . Κατά την αποστολή ή τη λήψη μηνυμάτων στο Gmail, υπάρχει επίσης η δυνατότητα ο χρήστης να δει το επίπεδο κρυπτογράφησης ενός μηνύματος. Το χρώμα του εικονιδίου διαφοροποιείται ανάλογα με το επίπεδο κρυπτογράφησης. Τέλος, με την ψηφιακή υπογραφή, η οποία υποστηρίζεται, διασφαλίζεται ότι τα μηνύματα δεν έχουν αλλοιωθεί.

Το TLS αποτρέπει κατά την επικοινωνία ενός server με μία εφαρμογή client, την υποκλοπή ή την αλλοίωση ενός μηνύματος από τρίτους. Για να εφαρμοστεί το TLS κατά την παράδοση ενός μηνύματος θα πρέπει οι υπηρεσίες παράδοσης μηνυμάτων τόσο του αποστολέα όσο και του παραλήπτη να χρησιμοποιούν το TLS.

Μερικές ακόμα συμβουλές από την Google προς τους χρήστες του Gmail είναι:

- να ελέγχουν τακτικά τα μηνύματα ηλεκτρονικού ταχυδρομείου. Εάν δεν προστατεύονται με κρυπτογράφηση, θα έχουν την ένδειξη *"Χωρίς TLS".*
- να ελέγχουν τακτικά το ιστορικό δραστηριότητας του λογαριασμού τους.
- να κλειδώνουν από απόσταση τη συσκευή τους, σε περίπτωση απώλειας, μέσα από τον λογαριασμό τους **(Google, 2020)**.

### 3.3.1.2. Ασφάλεια προσωπικών δεδομένων στο Outlook

Με στόχο την προστασία της ασφάλειας των υπηρεσιών ηλεκτρονικού ταχυδρομείου της, η Microsoft αξιοποιεί όλες τις τεχνικές ασφάλειας, όπως έχουν περιγραφεί παραπάνω, και επιπρόσθετα:

- **την Κρυπτογράφηση s/MIME**: Το άνοιγμα ενός μηνύματος με ψηφιακή κρυπτογράφηση είναι δυνατό μόνο από τους παραλήπτες, οι οποίοι διαθέτουν το σωστό ιδιωτικό κλειδί. Η ψηφιακή υπογραφή διασφαλίζει στους παραλήπτες των μηνυμάτων ότι το μήνυμα δεν έχει αλλοιωθεί.
- **την Κρυπτογράφηση TLS**, όπως περιγράφηκε παραπάνω.
- την **Microsoft 365** κρυπτογράφηση μηνυμάτων, για τους χρήστες του Outlook**.** Για να χρησιμοποιηθεί η κρυπτογράφηση μηνυμάτων Microsoft 365, ο αποστολέας πρέπει να έχει Microsoft 365 κρυπτογράφηση μηνυμάτων, η οποία περιλαμβάνεται στην άδεια χρήσης του Office 365 για μεγάλες επιχειρήσεις E3 **(Microsoft, 2020)**.

Προτείνεται επίσης να χρησιμοποιείται το **Outlook Web App** μόνο σε ασφαλή δίκτυα και τοποθεσίες, τις οποίες εμπιστεύεται ο χρήστης.

Γενικά, όλοι οι πάροχοι ηλεκτρονικού ταχυδρομείου συμβουλεύουν τους χρήστες τους:

- να συνδέονται με τη λειτουργία ανώνυμης περιήγησης και να μην ξεχνούν ανοιχτό τον λογαριασμό Google στον περιηγητή (browser) τους σε άλλες συσκευές.
- να χρησιμοποιούν μοναδικό κωδικό πρόσβασης για κάθε λογαριασμό.
- να προστατεύουν τις συσκευές τους (μοτίβο, PIN, δακτυλικό αποτύπωμα κ.α.).
- να κρατούν τις δυνητικά επιβλαβείς εφαρμογές εκτός της συσκευής τους.
- να αποφεύγουν τις απόπειρες ηλεκτρονικού ψαρέματος (Phising).
- να μην απαντούν σε spam E-mails.
- να χρησιμοποιούν ασφαλή δίκτυα και συνδέσεις.

Συνοψίζοντας, εκτός των τεχνολογικών εργαλείων, τα οποία προσφέρονται σε σχέση με την προστασία του χρήστη από παραβιάσεις μέσω ηλεκτρονικού ταχυδρομείου, είναι πολύ σημαντικό να εκπαιδευτούν οι χρήστες σε σχέση με το πως να αναγνωρίζουν από μόνοι τους τις απειλές αυτές και να τις αποφεύγουν. Εξίσου σημαντικό επίσης είναι να αναφέρονται στον πάροχο τα E-mails, τα οποία προέρχονται από άγνωστες πηγές και διαθέτουν τα χαρακτηριστικά ενός πλαστογραφημένου E-mail, όπως αυτά περιγράφηκαν παραπάνω.

# Κεφάλαιο 4. Αδυναμίες ασφάλειας διαδικτυακών εφαρμογών

Η συνεχόμενη αύξηση της ζήτησης σε σχέση με την ασφάλεια των διαδικτυακών εφαρμογών είχε ως αποτέλεσμα διάφοροι οργανισμοί να ασχολούνται αποκλειστικά με αυτό. Ένας από αυτούς είναι και η OWASP, η οποία έχει αναπτύξει δική της μεθοδολογία για penetration testing και είναι η πιο γνωστή μεταξύ άλλων. Επίσης, η OWASP παρέχει virtual machines για πρακτική εξάσκηση στο penetration testing, οδηγούς εκμάθησης καθώς και εργαλεία, τα οποία είναι χρήσιμα για τη διεξαγωγή των δοκιμών. Το penetration testing ορίζεται ως μία προσομοίωση επίθεσης με στόχο τον εντοπισμό αδυναμιών και ευπαθειών σε ένα σύστημα, ένα δίκτυο, μία εφαρμογή, έναν ιστότοπο κ.α. και θα περιγραφεί αναλυτικά στο Κεφάλαιο 6 **(PurpleSec, 2021)**.

## 4.1. OWASP Top 10

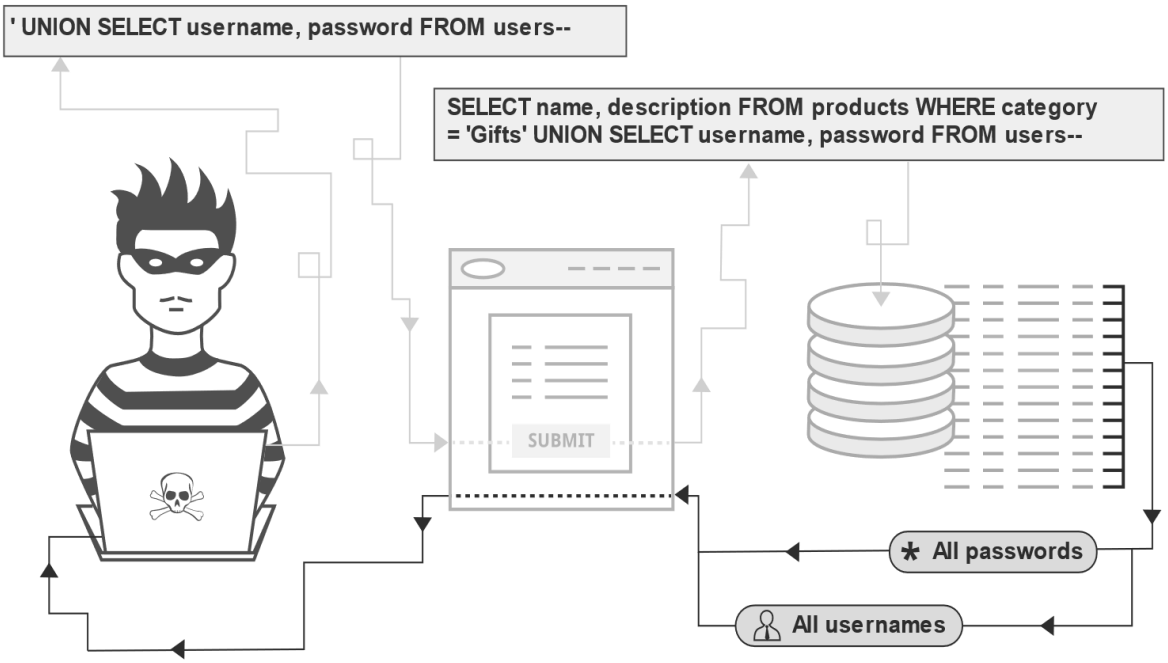
Το OWASP Top 10 είναι μία συλλογή από τις 10 πιο σημαντικές αδυναμίες των διαδικτυακών εφαρμογών. Ανανεώνεται κάθε χρόνο, συλλέγοντας αναφορές και στατιστικά στοιχεία από τις επιθέσεις, τα updates και τις αδυναμίες, οι οποίες βρέθηκαν. Ο κάθε penetration tester οφείλει να έχει υπόψιν του το OWASP TOP 10, καθώς οι αδυναμίες αυτές είναι υψηλού κινδύνου για οποιονδήποτε οργανισμό ή επιχείρηση.


**Εικόνα 12. OWASP (πηγή: https://owasp.org/)**

### 4.1.1. Injection

Αδυναμίες Injection υπάρχουν όταν ο χρήστης έχει τη δυνατότητα να εισάγει κάποια δεδομένα και αυτά να μεταφερθούν σε κάποια βάση δεδομένων ή και στο λειτουργικό σύστημα. Ειδικά, η δεύτερη περίπτωση είναι ιδιαίτερα σημαντική, καθώς δύναται να εκτελεστούν εντολές στο λειτουργικό, οι οποίες ενδεχομένως να αποβούν καταστροφικές για το σύστημα και την ομαλή λειτουργία της εφαρμογής. Σε αυτές τις περιπτώσεις ο επιτιθέμενος θα προσπαθήσει να διαβάσει ευαίσθητα αρχεία, να αντλήσει πληροφορία από τη βάση ή ακόμα και να δημιουργήσει χρήστη, με σκοπό να συνδεθεί.

Οι επιθέσεις Injection χωρίζονται γενικά σε 2 κατηγορίες: τις SQL Injection και τις Command Injection. Όταν υπάρχει κάποια φόρμα σύνδεσης και εγγραφής χρήστη ή κάποια μπάρα αναζήτησης, είναι βέβαιο ότι η συνάρτηση, η οποία είναι υπεύθυνη για τη λειτουργία αυτού του χαρακτηριστικού, θα έχει πρόσβαση στη βάση δεδομένων.




**Εικόνα 13. SQL Injection (πηγή: https://portswigger.net/web-security/sql-injection)**

Σε περίπτωση σύνδεσης (ή αναζήτησης), η εφαρμογή αφού δεχτεί την είσοδο από τον χρήστη θα στείλει ένα query στη βάση δεδομένων με στόχο να επαληθεύσει εάν υπάρχει αυτή η εγγραφή και να του επιστρέψει την κατάλληλη απάντηση. Το query αυτό θα έχει την εξής μορφή:

```
SELECT * FROM users WHERE username='XXX' and password='YYY';
```

όπου XXX και YYY το όνομα και ο κωδικός αντίστοιχα, τα οποία έδωσε ο χρήστης. Εάν τα δεδομένα εισόδου δεν ελέγχονται για χαρακτήρες, οι οποίοι είναι δυνατόν να προκαλέσουν απρόβλεπτη λειτουργία στη βάση, ο χρήστης θα ήταν σε θέση να δώσει ως username: ‘ or 1=1 - - και τότε το query θα ήταν:

```
SELECT * FROM users WHERE username='' or 1=1 - - and password='';
```

Αυτό το query θα επιστρέψει οτιδήποτε (*) εμπεριέχεται στον πίνακα users, με μόνη προϋπόθεση το username να είναι κενό ή 1=1. Οι χαρακτήρες - - υποδηλώνουν σχόλια. Συνεπώς, οτιδήποτε βρίσκεται μετά σε αυτό θα αγνοηθεί.

Η Command Injection αδυναμία είναι πιο σπάνια και συνήθως είναι δυνατόν να διαπιστωθεί εάν είναι ευπαθής η εφαρμογή, χρησιμοποιώντας κάποια εντολή, η οποία εκτυπώνει κάποιο μήνυμα. Μία ακόμα τεχνική είναι να χρησιμοποιηθεί κάποιο εργαλείο, το οποίο ελέγχει την κίνηση του δικτύου - όπως το netcat ή το wireshark, και να δώσει στην εφαρμογή την εντολή ping με παράμετρο την IP του. Εάν ο επιτιθέμενος διαπιστώσει κίνηση, η οποία προέρχεται από την εφαρμογή, τότε είναι σε θέση να επαληθεύσει ότι είναι ευπαθής.

### 4.1.2. Broken Authentication

Κάθε εφαρμογή, η οποία δίνει τη δυνατότητα διατήρησης λογαριασμού χρήστη, πρέπει να υποστηρίζει και τρόπο πιστοποίησης του. Αδυναμίες τέτοιου τύπου σχετίζονται με λογικά σφάλματα ή μη ορθή λειτουργία στη διαδικασία της πιστοποίησης. Έτσι, κάποιος έχει τη δυνατότητα να συνδεθεί σαν νόμιμος χρήστης ή ακόμα και να αποκτήσει δικαιώματα διαχείρισης στην εφαρμογή (administrator user), ενώ υπό φυσιολογικές συνθήκες δεν θα έπρεπε. Εάν συνδεθεί σαν user είναι ικανός να επηρεάσει την υστεροφημία της εφαρμογής, αφού εκθέτει τα προσωπικά δεδομένα των χρηστών, ενώ σαν administrator την ίδια την ακεραιότητα της εφαρμογής.

Σε κάθε εφαρμογή με χρήστες υπάρχει ο επισκέπτης - ο οποίος δεν έχει λογαριασμό και έχει τα λιγότερα δικαιώματα -, ο client - ο οποίος έχει δικαιώματα σε κάποιες λειτουργίες και σελίδες της εφαρμογής και απόλυτο έλεγχο στον λογαριασμό του - και ο administrator, ο οποίος είναι σε θέση να αλλάζει το περιεχόμενο και τα δεδομένα της εφαρμογής καθώς και να επεξεργάζεται μηνύματα και άλλους χρήστες. Η δυνατότητα ένας χρήστης να πάρει δικαιώματα κάποιου άλλου χρήστη ονομάζεται horizontal privilege escalation (οριζόντια κλιμάκωση δικαιωμάτων), ενώ όταν παίρνει περισσότερα δικαιώματα από αυτά τα οποία του επιτρέπονται, ονομάζεται vertical privilege escalation (κάθετη κλιμάκωση δικαιωμάτων). Η ευπάθεια αυτή συναντάται και στα λειτουργικά συστήματα και είναι ο τελικός σκοπός κάθε επιτιθέμενου ή penetration tester.

Το Basic HTTP Authentication είναι ένας τρόπος πιστοποίησης, τον οποίο παρέχει το ίδιο το πρωτόκολλο HTTP. Σε κάποιες εκδόσεις του Apache Server εάν άλλαζε η μέθοδο από GET σε OPTIONS, επιτρεπόταν η πρόσβαση χωρίς την πιστοποίηση της ταυτότητας του χρήστη.

### 4.1.3. Sensitive Data Exposure

Sensitive Data Exposure χαρακτηρίζεται η πρόσβαση μη εξουσιοδοτημένων χρηστών σε “ευαίσθητα” δεδομένα. Στοιχεία χρηστών ή στοιχεία για τη λειτουργία της εφαρμογής, όπως ονόματα αρχείων, συναρτήσεων κ.λπ. πρέπει να αποκρύπτονται. Η ανάκτηση τέτοιας πληροφορίας εξοπλίζει τον επιτιθέμενο με γνώσεις, όπου ο συνδυασμός τους ενδεχομένως να διευρύνει τις επιλογές επιθέσεων.

Σύνηθες φαινόμενο είναι στον κώδικα HTML, όπου κάθε επισκέπτης έχει πρόσβαση, να βρίσκονται σχόλια, τα οποία αναφέρουν ονόματα συναρτήσεων, κρυφά directories της εφαρμογής ακόμα και username και password κάποιου χρήστη. Οι προγραμματιστές συνηθίζουν να γράφουν σχόλια σε κάθε εφαρμογή την οποία αναπτύσσουν, είτε για να εξηγούν κάποιο περίπλοκο κομμάτι κώδικα είτε για να σημειώσουν το τι πρέπει να κάνουν. Πολλές φορές, οι προσωπικές σημειώσεις συνοδεύονται με το keyword: TODO (να κάνω).

Επίσης, κατά τη δημιουργία της εφαρμογής θα χρειαστεί να προσθέσουν ή να αλλάξουν κρίσιμα κομμάτια κώδικα. Για αυτόν τον λόγο δημιουργούνε backup αρχεία, τα οποία ενδεχομένως να μην διαγραφούν ποτέ. Ένας penetration tester ελέγχει και για backup αρχεία, ειδικά εάν έχει υπόψιν του κάποια ήδη υπάρχοντα.

### 4.1.4. XXE (**X**ML E**x**ternal **E**ntities**)**

Εάν δεν έχει ρυθμιστεί σωστά η επεξεργασία των XML στον κώδικα, είναι δυνατόν κάποιος να χρησιμοποιήσει τα external entities, με σκοπό να διαβάσει αρχεία, να εκτελέσει απομακρυσμένο κώδικα ή ακόμα και να κάνει **επίθεση άρνησης εξυπηρέτησης (denial of services)**. Θεωρούνται εύκολα κατανοητές και απλές σαν επιθέσεις.

Η XML (E**x**tensible **M**arkup **L**anguage) είναι μία γλώσσα κανόνων, η οποία παρέχει συγκεκριμένη διαδικασία πρόσβασης στην πληροφορία. Πολλές φορές χρησιμοποιείται στον παγκόσμιο ιστό, καθώς βοηθάει στην πιο αποδοτική χρήση, μετάδοση, αποθήκευση και προβολή δεδομένων.

### 4.1.5. Broken Access Control

Για κάθε ομάδα χρηστών επιτρέπονται συγκεκριμένες σελίδες, όπου επιτρέπεται να έχουν πρόσβαση καθώς και συγκεκριμένα δεδομένα. Τέτοιες αδυναμίες επιτρέπουν σε έναν απλό χρήστη να έχει πρόσβαση σε σελίδες, οι οποίες είναι έξω από το πεδίο δράσης (scope) του, όπως παραδείγματος χάρη, κάποιου άλλου χρήστη ή ενός διαχειριστή, βάσει της **Αρχής του Ελάχιστου Προνομίου**.

Το περιεχόμενο, στο οποίο θα έχει πρόσβαση κάποιος χρήστης από προγραμματιστική σκοπιά, ελέγχεται μέσα από cookies. Όταν κάποιος συνδέεται σε μία εφαρμογή, αυτή του ορίζει ένα session ανάλογα με τον ρόλο του σε αυτήν. Όταν προσπαθήσει να δει το περιεχόμενο κάποιας σελίδας, η εφαρμογή θα ελέγξει τον ρόλο του καθώς και τον ρόλο τον οποίο θα πρέπει να έχει κάποιος για αυτήν τη σελίδα ή για τη συνάρτηση της εφαρμογής. Πολλές φορές όμως, κάποιος κακόβουλος έχει τη δυνατότητα να αλλάξει τον ρόλο του και να δηλώσει έναν διαφορετικό επιτρέποντας του τελικά πρόσβαση σε κρίσιμες λειτουργίες της εφαρμογής.

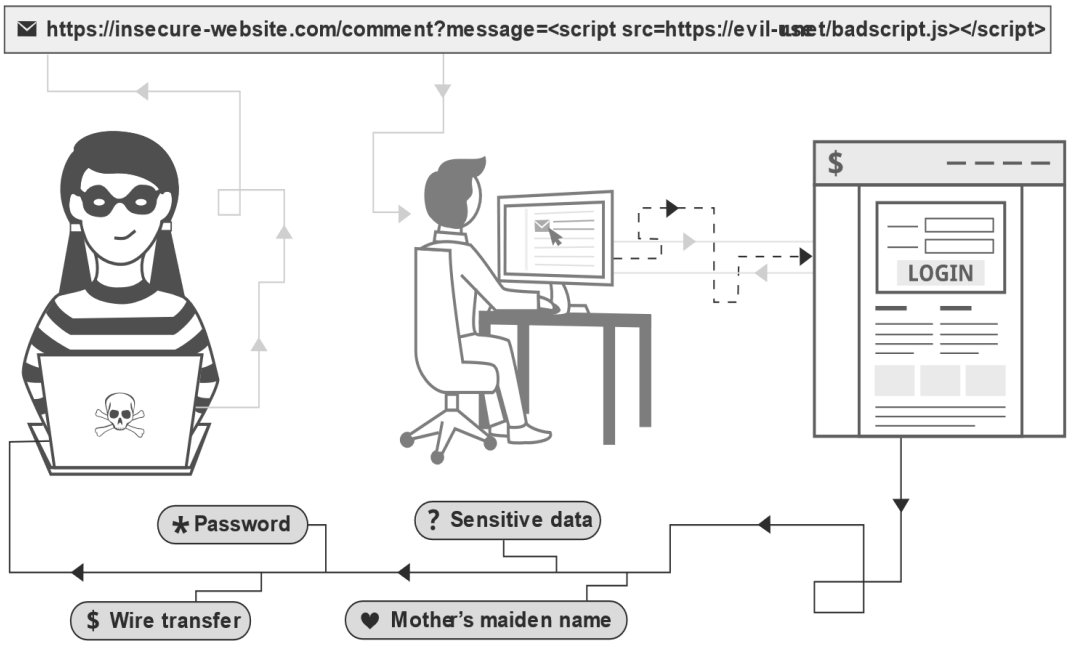
Αν και υπάρχουν τρόποι να “κρυφτούν” αυτές οι σελίδες από μη εξουσιοδοτημένους χρήστες, πολύ εύκολα κάποιος έχει τη δυνατότητα να τις ανακαλύψει είτε με τη χρήση αυτοματοποιημένων εργαλείων είτε με τον συνδυασμό διάφορων επιθέσεων.

### 4.1.6. Security Misconfiguration

Κακώς ορισμένες τεχνικές ασφάλειας, οι οποίες συνδέονται με τον κώδικα της εφαρμογής ή με τις ρυθμίσεις του συστήματος/ server, όπου τρέχει η εφαρμογή, ενδεχομένως να οδηγήσουν σε πλήθος επιθέσεων. Εάν συνδυαστούν με την αδυναμία Sensitive Data Exposure, υπάρχει η πιθανότητα ο επιτιθέμενος να αποκτήσει πρόσβαση σε αρχεία .config, τα οποία περιέχουν ρυθμίσεις για το σύστημα.

### 4.1.7. XSS (Cross-Site Scripting)

Η επίθεση XSS έχει να κάνει με την εκτέλεση Javascript κώδικα. Καθώς η Javascript εκτελείται στην πλευρά του client, αυτό ενδεχομένως να οδηγήσει στην ανάγνωση cookies και στην εκτέλεση ρουτίνων χωρίς τη συγκατάθεση του χρήστη. Ειδικά όταν ο κώδικας Javascript αποθηκεύεται, όπως παραδείγματος χάρη σε κάποιο forum, ενδεχομένως να υπάρξουν σημαντικές συνέπειες. Κάθε χρήστης, ο οποίος θα επιθυμεί να διαβάσει κάποιο thread σε ένα forum, μαζί με το περιεχόμενο του thread θα του αποστέλλεται και ο κώδικας Javascript, ο οποίος θα τρέχει τοπικά στον υπολογιστή του εν αγνοία του.




**Εικόνα 14. XSS (πηγή: https://portswigger.net/web-security/cross-site-scripting)**

### 4.1.8. Insecure Deserialization

Η επίθεση αυτή θεωρείται από πολλούς σύνθετη, καθώς απαιτεί καλή γνώση σε σχέση με το πως δουλεύει ο αντικειμενοστραφής προγραμματισμός (object-oriented programming) καθώς και τον τρόπο με τον οποίο περνάνε τα αντικείμενα (objects) μέσα από ένα HTTP πακέτο. Ενδεχομένως κάποιος να διαβάσει χρήσιμα αρχεία ή ακόμα και να αποκτήσει παραπάνω δικαιώματα στη λειτουργικότητα της εφαρμογής. Τέτοιες επιθέσεις υφίστανται, όταν δεν πραγματοποιείται κατάλληλος έλεγχος των δεδομένων εισόδου, τα οποία εισάγει ο χρήστης.

Κάθε γλώσσα προγραμματισμού έχει μέθοδο σειριακής αναπαράστασης των δεδομένων της. Με αυτόν τον τρόπο, είναι δυνατό να σταλεί μέσω του HTTP πακέτου. Η μέθοδος αυτή στην PHP ονομάζεται “Serialize”, στην Python “Pickle” και στην Java “writeObject”.

### 4.1.9. Using Components with Known Vulnerabilities

Πολύ συχνά οι προγραμματιστές χρησιμοποιούν, όπως είναι φυσικό, έτοιμες βιβλιοθήκες και συναρτήσεις οι οποίες ίσως να είναι γνωστές για κάποια ευπάθεια τους. Κάποιος κακόβουλος χρήστης ενδεχομένως να εκμεταλλευτεί το παραπάνω, συνήθως με έτοιμο διαθέσιμο κώδικα ή μέθοδο. Επίσης, τα ίδια τα frameworks ανάλογα με την έκδοση τους είναι πιθανό να είναι ευπαθή σε κάποιον τύπο επίθεσης.

Υπάρχουν πολλά αυτοματοποιημένα και υψηλής απόδοσης εργαλεία, τα οποία ελέγχουν την εφαρμογή για τέτοιες ευπάθειες (scanning). Εάν ανακαλυφθεί κάποια ευπάθεια, ο penetration tester είναι σε θέση να την εκμεταλλευτεί χρησιμοποιώντας το Metasploit framework μαζί με το κατάλληλο exploit.

### 4.1.10. Insufficient Logging and Monitoring

Κάθε εφαρμογή ανεξάρτητα με το αν είναι διαδικτυακή ή όχι πρέπει να κρατάει logs, τα οποία είτε κρατάνε progress για τη λειτουργία της ή για στοιχεία σύνδεσης των χρηστών είτε γράφουν κάποια errors ή warnings, τα οποία ίσως να προκύψουν από την ίδια την εφαρμογή από ασυμβατότητα με κάποια εξωτερική βιβλιοθήκη ή από πρόβλημα του ίδιου του συστήματος. Αυτό βοηθάει τους developers να βελτιώνουν την εφαρμογή και να την αποσφαλματώνουν (debugging) ελέγχοντας απλά τα logs, τα οποία εκείνη τους παρέχει.

Επιπρόσθετα, εάν παρατηρηθεί κάποια ύποπτη κίνηση ή κάποια απώλεια/ παραποίηση δεδομένων, οι διαχειριστές και οι υπεύθυνοι για θέματα ασφάλειας θα πρέπει να μελετήσουν τις τελευταίες καταγραφές και να ελέγξουν εάν αυτό είναι αποτέλεσμα κακόβουλης ενέργειας ή δυσλειτουργίας της εφαρμογής.

Η πληροφορία, η οποία θα αποθηκεύεται, συνήθως ορίζεται από τους διαχειριστές και τους υπεύθυνους ασφάλειας. Εξαρτάται όμως και από παράγοντες υλικού, όπως την ταχύτητα ρολογιού του επεξεργαστή, την ταχύτητα ανάγνωσης και εγγραφής του αποθηκευτικού μέσου (HDD/ SSD) καθώς και τη χωρητικότητα του **(OWASP, 2020)**.

# Κεφάλαιο 5. Σχεδίαση και ανάπτυξη εφαρμογής *"UniStudent"*

## 5.1. Περιγραφή εφαρμογής

Πρόκειται για μία εφαρμογή, τύπου φοιτητολογίου – estudy, η οποία έχει αναπτυχθεί με βάση την ασφάλεια. Απευθύνεται στους φοιτητές, οι οποίοι είναι εγγεγραμμένοι σε ένα τμήμα. Η εφαρμογή βρίσκεται online στη διεύθυνση ***https://www.unistudent.eu/***.

Αποτελείται από τρεις υπηρεσίες/ λειτουργίες (σελίδες):

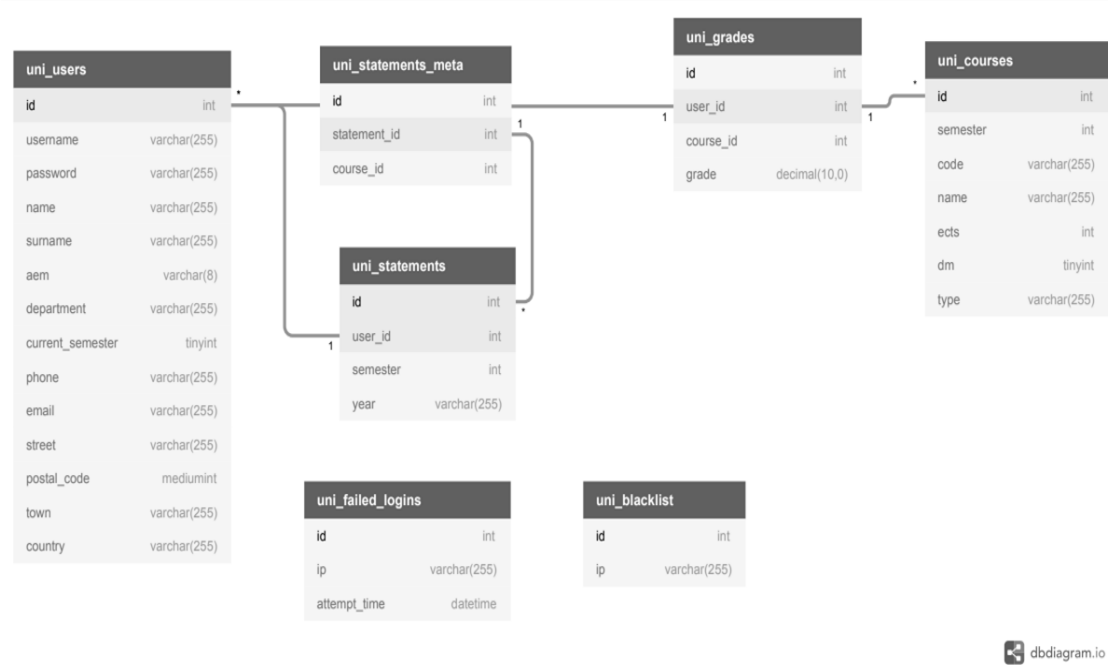
- *Προφίλ*,
- *Δηλώσεις*,
- *Βαθμολογίες*.

- Στη σελίδα ***"Προφίλ"*** ο χρήστης θα έχει τη δυνατότητα να δει τα προσωπικά στοιχεία του (π.χ. ονοματεπώνυμο, ΑΜ, E-mail, εξάμηνο φοίτησης, τμήμα φοίτησης κ.α.), τα οποία υπάρχουν αποθηκευμένα στη βάση της εφαρμογής.
- Στη σελίδα ***"Δηλώσεις"*** θα έχει τη δυνατότητα να δει τις δηλώσεις μαθημάτων του καθ' όλη τη διάρκεια των σπουδών του και επιλέγοντας το εξάμηνο φοίτησης να δει τα μαθήματα, τα οποία είχαν δηλωθεί το συγκεκριμένο εξάμηνο.
- Στη σελίδα ***"Βαθμολογίες"*** θα έχει τη δυνατότητα να ενημερωθεί σχετικά με τις καταχωρημένες βαθμολογίες του ανά μάθημα.

## 5.2. Περιγραφή βάσης

Το σχεσιακό σχήμα δημιουργήθηκε με τη βοήθεια του εργαλείου σχεδίασης Dbdiagram, το οποίο βρίσκεται στον ιστότοπο *https://dbdiagram.io/*. Η σχεσιακή βάση της εφαρμογής ***"UniStudent"*** αποτελείται από πέντε πίνακες: **1.** uni_users, **2.** uni_statements_meta, **3.** uni_statements, **4.** uni_grades και **5.** uni_courses, οι οποίοι διαχειρίζονται τα δεδομένα του χρήστη (προσωπικά στοιχεία, μαθήματα, βαθμοί κ.α.), τα οποία εμφανίζονται στις σελίδες "*Προφίλ*", "*Δηλώσεις*" και "*Βαθμολογίες*".

Στη βάση υπάρχουν επίσης οι πίνακες uni_failed_logins και uni_blacklist, οι οποίοι είναι ανεξάρτητοι και αφορούν στην καταγραφή των αποτυχημένων προσπαθειών σύνδεσης και στις IP διευθύνσεις των συσκευών, από όπου έχουν πραγματοποιηθεί οι προσπάθειες επίθεσης, αντίστοιχα. Στο σχεσιακό σχήμα, όπως φαίνεται στην Εικόνα 15, είναι εμφανείς οι σχέσεις μεταξύ των πινάκων καθώς και το κύριο και το δευτερεύον κλειδί κάθε πίνακα.



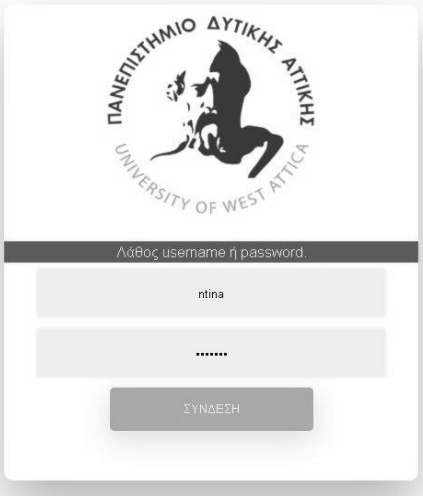

**Εικόνα 15. Σχεσιακό σχήμα της βάσης**

## 5.3. Σχεδίαση εφαρμογής

Η εφαρμογή αναπτύχθηκε με το πρωτόκολλο HTTPS και διαθέτει εγκατεστημένο SSL πιστοποιητικό και ιδιωτικό κλειδί τύπου RSA. Ο RSA είναι ισχυρός κρυπτογραφικός αλγόριθμος ασύμμετρου κλειδιού, ο οποίος επιτρέπει την κρυπτογράφηση μηνυμάτων και χρησιμοποιείται και ως ψηφιακή υπογραφή. Τα πιστοποιητικά SSL, τα οποία εκδίδονται από ανεξάρτητες πηγές (Αρχές Πιστοποίησης – Certification Authorities), ανάλογα με τον τύπο τους πιστοποιούν την κατοχύρωση ενός domain, όπως επίσης την ύπαρξη και τα στοιχεία του ιδιοκτήτη της ιστοσελίδας. Τα πιστοποιητικά SSL διατίθενται είτε μέσω πληρωμής είτε δωρεάν από τον οργανισμό **Let's Encrypt**.

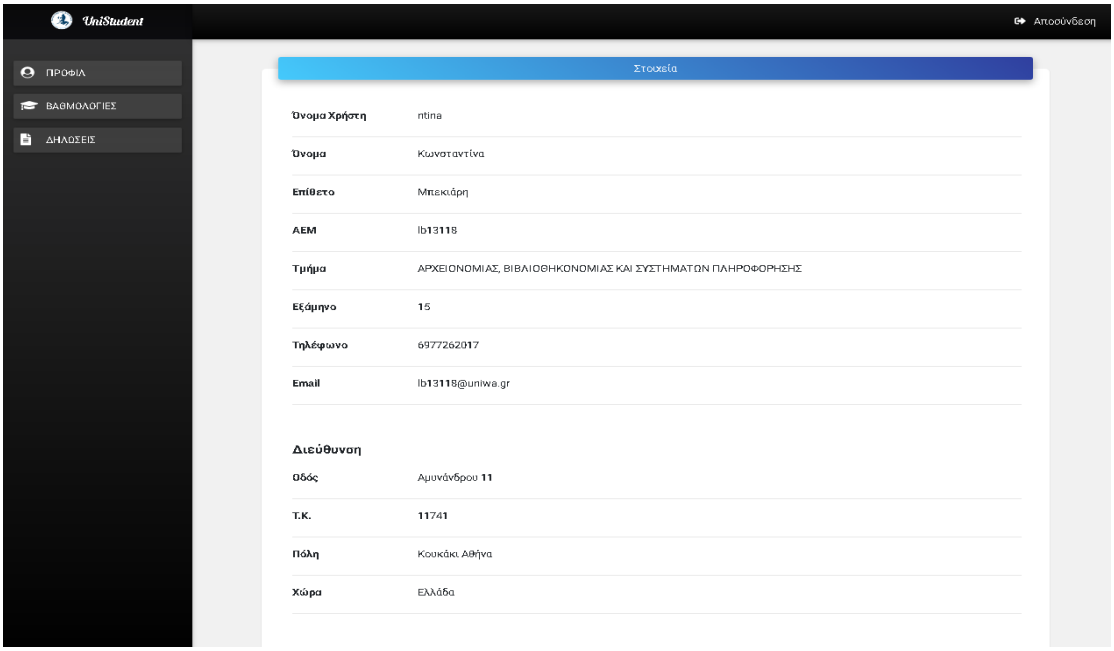
Αρχικά, εμφανίζεται η σελίδα σύνδεσης του χρήστη, στην οποία θα συμπληρώνει τα στοιχεία του (username και password) για να συνδεθεί **(*Εικόνα 16*)**. Σε αυτό το σημείο δεν θα εμφανίζεται τίποτα παραπάνω, εκτός από τα απαραίτητα πεδία, σύμφωνα με την **Αρχή της Ελάχιστης Επιφάνειας Επίθεσης**. Έπειτα από λανθασμένη πληκτρολόγηση στοιχείων εισόδου θα εμφανίζεται αντίστοιχο μήνυμα, το οποίο θα προτρέπει τον χρήστη να πληκτρολογήσει ξανά τα στοιχεία του **(*Εικόνα 17*)**.


**Εικόνα 16. Σελίδα "Login" για τη σύνδεση του φοιτητή στο σύστημα**




**Εικόνα 17. Μήνυμα προτροπής έπειτα από αποτυχημένη σύνδεση**

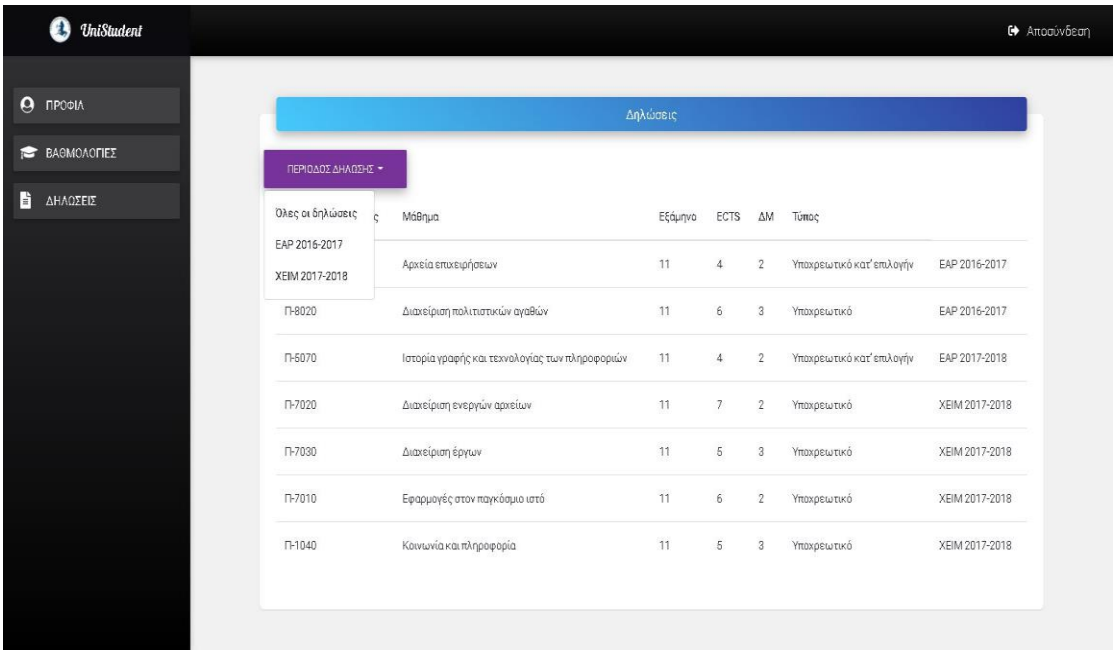
Αφού πραγματοποιηθούν οι απαραίτητοι έλεγχοι σε σχέση με την ορθότητα των στοιχείων, τα οποία έδωσε ο χρήστης, ο τελευταίος θα εισέρχεται στο σύστημα με πρώτη εμφάνιση τη σελίδα *“Προφίλ”*.



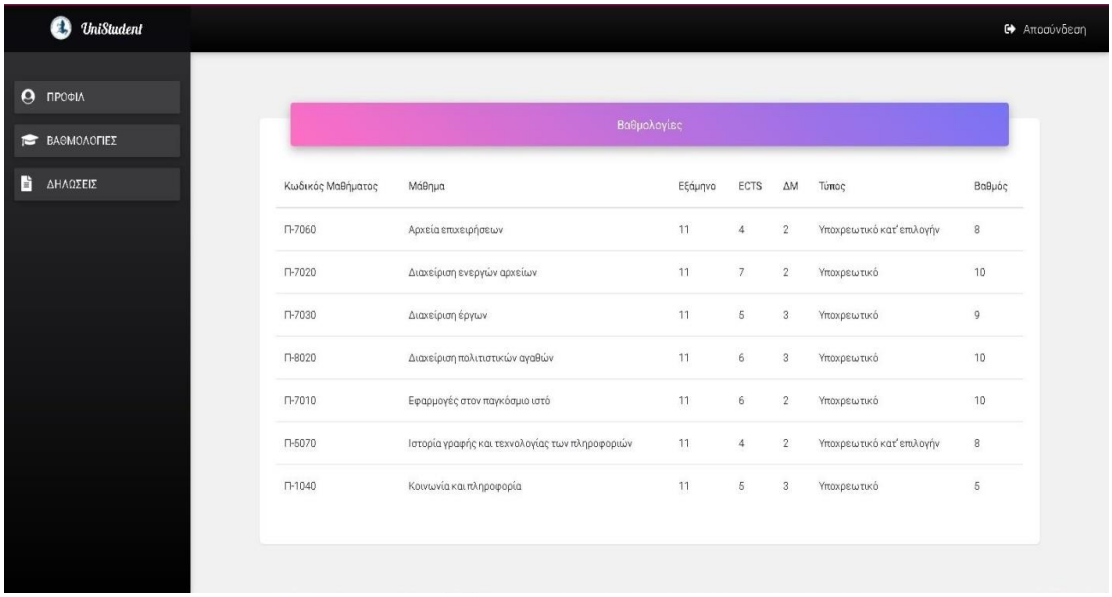

**Εικόνα 18. Σελίδα “Προφίλ” του φοιτητή**

Ο φοιτητής δεν έχει το δικαίωμα να κάνει τροποποιήσεις στα προσωπικά του στοιχεία στη σελίδα "*Προφίλ*", μόνο έπειτα από επικοινωνία με τον διαχειριστή του συστήματος, εφαρμόζοντας την **Αρχή του Ελάχιστου Προνομίου**.

Εν συνεχεία, του δίνεται η επιλογή να πλοηγηθεί από το πλαϊνό μενού στις σελίδες *"Προφίλ", "Βαθμολογίες"* και *"Δηλώσεις".*




**Εικόνα 19. Σελίδα "Δηλώσεις" του φοιτητή**




**Εικόνα 20. Σελίδα "Βαθμολογίες" του φοιτητή**

Τέλος, υπάρχει κουμπί "Έξοδος" για την αποσύνδεση του χρήστη από την εφαρμογή, το οποίο βρίσκεται στο δεξί μέρος του navigation bar της εφαρμογής.

Για την ανάπτυξη της εφαρμογής χρησιμοποιήθηκαν οι παρακάτω τεχνολογίες:

i. **HTML 5.0:** για τη δόμηση των πληροφοριών στην πλευρά του client
ii. **CSS 3.0:** για την εμφάνιση των πληροφοριών στην πλευρά του client
iii. **Bootstrap 4:** HTML, CSS και JavaScript framework
iv. **Javascript:** για την ομαλή λειτουργία της εφαρμογής και τη διάδραση του χρήστη
v. **jQuery 3.3.1:** δημοφιλής βιβλιοθήκη για χρήση της Javascript
vi. **PHP 7.3.6:** για την ανάπτυξη server-side κώδικα
vii. **MySQL:** για την ανάπτυξη της βάσης δεδομένων
viii. **MariaDB 10.2:** για τη διαχείριση της βάσης δεδομένων

Η εφαρμογή έχει αναπτυχθεί με στόχο να δεχθεί και να αμυνθεί σε επιθέσεις τύπου:

i. SQL Injection
ii. PHP Injection
iii. Broken Authentication
iv. Sensitive Data Exposure
v. Security Misconfiguration
vi. Brute Force [7]
vii. XSS

[7] **Η Brute Force επίθεση αναφέρεται στην εξαντλητική δοκιμή πιθανών κλειδιών, τα οποία παράγουν ένα κρυπτογράφημα.**

Δεν έχουν πραγματοποιηθεί προσπάθειες ενίσχυσης της άμυνας της εφαρμογής σε σχέση με τις παρακάτω επιθέσεις, καθώς:

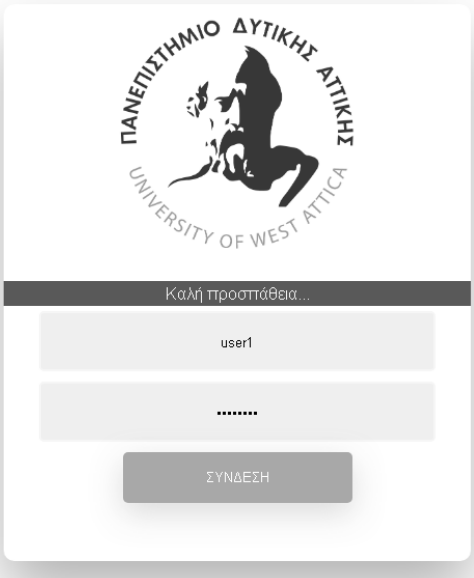
- η επίθεση **X**ML E**x**ternal **E**ntities (XXE) απαιτεί την ύπαρξη XML κώδικα.
- η επίθεση **Insufficient Logging and Monitoring** απαιτεί την ύπαρξη αρχείων logs.
- η επίθεση **Using Vulnerable Components** απαιτεί την ύπαρξη σύνθετου κώδικα (π.χ. έτοιμες βιβλιοθήκες/ συναρτήσεις).
- η επίθεση **Broken Access Control** απαιτεί την ύπαρξη χρηστών με διαφορετικά δικαιώματα πρόσβασης.
- η επίθεση **Insecure Deserialization** απαιτεί την ύπαρξη αντικειμενοστραφούς προγραμματισμού.

### 5.3.1. Τεχνικές ασφάλειας εφαρμογής

Για την αποφυγή **XSS**, **SQL Injections** και **PHP Injections** έχει αναπτυχθεί κώδικας στο ***authenticate.php***, με στόχο να φιλτράρεται η εισαγωγή δεδομένων, με τον οποίο αποκόπτονται:

- κενά και
- ειδικοί χαρακτήρες.

Έτσι εάν πληκτρολογηθεί κάποιο από τα παραπάνω, εμφανίζεται κατάλληλο μήνυμα προτροπής, το οποίο ενημερώνει τον επιτιθέμενο ότι απορρίφθηκε η προσπάθεια επίθεσης **(*Εικόνα 21*)**. Είναι αποδεκτή μόνο η χρήση γραμμάτων - κεφαλαίων/ πεζών - και αριθμών κατά την πληκτρολόγηση του κωδικού πρόσβασης. Ο κωδικός πρόσβασης ενός φοιτητή ορίζεται από τη γραμματεία του τμήματος.




**Εικόνα 21. Μήνυμα προτροπής μετά από αποτυχημένη προσπάθεια επίθεσης στο πεδίο password**

Εν συνεχεία, για περισσότερη ασφάλεια, οι κωδικοί πρόσβασης είναι αποθηκευμένοι στη βάση δεδομένων εντός του πίνακα ***uni_users*** και

κρυπτογραφημένοι με τη χρήση του κρυπτογραφικού αλγορίθμου **Bcrypt**. Χαρακτηριστικό γνώρισμα του αλγορίθμου Bcrypt είναι ότι το «$2υ$» βρίσκεται στην αρχή κάθε hash που δημιουργεί. Το κρυπτογράφημα που δημιουργεί αντιστοιχεί στον κωδικό πρόσβασης, ο οποίος έχει αρχικά οριστεί από τον διαχειριστή του συστήματος.

Η εφαρμογή προστατεύεται επίσης από **Επιθέσεις αποκάλυψης πληροφοριών**, παραδείγματος χάρη αποτρέποντας τον επιτιθέμενο από το να δει τον κατάλογο αρχείων. Ο επιτιθέμενος είναι ικανός να αντιληφθεί ότι υπάρχει ένα σύστημα αρχείων· δεν έχει όμως τη δυνατότητα πρόσβασης στα αρχεία, καθώς σε κάθε προσπάθεια ανακατευθύνεται σε ένα κενό αρχείο ***info.php***, το οποίο υπάρχει σε κάθε φάκελο στον κατάλογο αρχείων.

Σχετικά με την παραβίαση **Broken authentication**, έχει αποφευχθεί η χρήση Cookies, καθώς η ασφαλής χρήση Cookies αμφισβητείται από την κοινότητα. Έτσι, το session id αποθηκεύεται στον server, οπότε είναι αδύνατο να κλαπούν πληροφορίες της session (π.χ. session id, username, password κ.α.).

Με στόχο την παρεμπόδιση **Brute Force** επιθέσεων, έχει αναπτυχθεί κώδικας, ο οποίος (α) στην περίπτωση που η προσπάθεια σύνδεσης περιέχει αλφαριθμητικούς χαρακτήρες, μετά από πέντε αποτυχημένες προσπάθειες εντός μίας ώρας, ανακατευθύνει τον χρήστη στο αρχείο ***blocked.php***. Συγκεκριμένα, καταγράφεται η IP διεύθυνση και η χρονική στιγμή της πρώτης λανθασμένης εισόδου στον πίνακα ***uni_failed_logins,*** ο οποίος ανανεώνεται κάθε 6 ώρες διαγράφοντας το περιεχόμενο του· (β) στην περίπτωση που η επίθεση περιέχει ειδικούς χαρακτήρες θα εντάξει την IP διεύθυνση, από την οποία πραγματοποιείται η επίθεση, σε μία ***Blacklist*** και θα αποτρέπει τον επιτιθέμενο στο εξής από το να δει την ιστοσελίδα. Η Blacklist βρίσκεται στη βάση με την ονομασία ***uni_blacklist*** και ανανεώνεται αυτόματα, διαγράφοντας το περιεχόμενο της κάθε 24 ώρες περίπου, με τη χρήση του εργαλείου Cron Job.

Ακόμα μία τεχνική ασφάλειας, η οποία έχει χρησιμοποιηθεί, είναι η δημιουργία ενός **honeypot** στη σελίδα σύνδεσης. "Ένα honeypot είναι ένας πόρος

πληροφοριακών συστημάτων, του οποίου η αξία έγκειται στη μη εξουσιοδοτημένη ή παράνομη χρήση αυτού**" (Πάγκαλος & Μαυρίδης, 2002)**. Τα honeypots αποτελούν δηλαδή τεχνικές, οι οποίες έχουν ως στόχο να αποσπάσουν την προσοχή των επιτιθέμενων, καταγράφοντας ταυτόχρονα τα στοιχεία τους **(The Greek Honeynet Project, 2016)**. Στην περίπτωση της συγκεκριμένης εφαρμογής, όταν ο επιτιθέμενος (π.χ. bot) συμπληρώσει οτιδήποτε στο πεδίο "Don't fill if human." η IP διεύθυνση του μπλοκάρεται από τον server της εφαρμογής και καταγράφεται στην Blacklist.

Για περισσότερη ασφάλεια έχει απενεργοποιηθεί η δυνατότητα μεταφόρτωσης αρχείων (file upload) από τις ρυθμίσεις του server. Έτσι, ελαχιστοποιείται η επιφάνεια επίθεσης.

Εν συνεχεία, θα πρέπει να αναφερθεί ότι έχει ληφθεί υπόψιν και το περιθώριο λάθους κατά την πληκτρολόγηση των στοιχείων σύνδεσης του χρήστη. Έπειτα από πέντε συνεχόμενες αποτυχημένες προσπάθειες σύνδεσης στο Login, κάνοντας λανθασμένη πληκτρολόγηση του username ή password, η IP διεύθυνση του χρήστη μπλοκάρεται για 1 ώρα από την πρώτη αποτυχημένη σύνδεση, ανακατευθύνοντας τον χρήστη στο αρχείο ***blocked.php***. Όπως αναφέρθηκε και παραπάνω, καταγράφεται η IP διεύθυνση και η χρονική στιγμή της πρώτης λανθασμένης εισόδου στον πίνακα ***uni_failed_logins*** της βάσης. Μετά το πέρας της 1 ώρας, ο χρήστης θα έχει τη δυνατότητα να προσπαθήσει να συνδεθεί ξανά στην εφαρμογή. Για αυτόν τον λόγο έχει δημιουργηθεί μία Javascript function, η οποία ενημερώνει τον χρήστη για τον υπολειπόμενο χρόνο με κατάλληλο μήνυμα π.χ. *"Ξαναπροσπάθησε σε 0 ώρες 57 λεπτά 51 δευτερόλεπτα."*.

Στη συνέχεια παρατίθεται μέρος του κώδικα ανάπτυξης της ιστοσελίδας σε PHP από το αρχείο ***functions.php***, το οποίο καλείται κατά την αυθεντικοποίηση ελέγχοντας εάν η IP διεύθυνση βρίσκεται (α) στον πίνακα ***uni_blacklist***, ο οποίος προκύπτει, όπως αναφέρθηκε παραπάνω, από κάποια προσπάθεια επίθεσης, ή (β) στον πίνακα ***uni_failed_logins***, εφόσον έχουν προηγηθεί πάνω από πέντε αποτυχημένες προσπάθειες σύνδεσης. Στην (α) περίπτωση όταν προσπαθήσει να εισέλθει στο ***login.php***, εμφανίζεται το μήνυμα: *"HTTP/1.1 400 Bad Request"*. Ενώ εάν ισχύει η (β) περίπτωση, αυτό έχει ως αποτέλεσμα να αποκλειστεί η IP

διεύθυνση του χρήστη για 1 ώρα, η οποία εάν δεν έχει επέλθει τη χρονική στιγμή του ελέγχου, ανακατευθύνει τον χρήστη στη σελίδα ***blocked.php***. Και στις δύο περιπτώσεις η εφαρμογή αποτρέπει τον χρήστη από το να έχει πρόσβαση στη σελίδα σύνδεσης.

```
<?php

require_once './db-config.php';

function is_ip_blocked()
{
   global $mysqli;
    $ip = $_SERVER['REMOTE_ADDR'];
    $query = "SELECT COUNT(ip) AS ips FROM uni_blacklist WHERE ip LIKE
INET_ATON(?)";


    if( $stmt = $mysqli->prepare( $query ) )
    {
        // Bind parameters (s = string, i = int, b = blob, etc)
        $stmt->bind_param('s', $ip);
        $stmt->execute();
        $res = $stmt->get_result();
        $stmt->close();
        $response = $res->fetch_all(MYSQLI_ASSOC);
        $response = $response[0];

        if( $response['ips'] > 0)
        {
            header("HTTP/1.1 400 Bad Request");
            $mysqli->close();
            die();
        }
    }
    else
    {
        echo 'ERROR';
        header("HTTP/1.1 400 Bad Request");
        $mysqli->close();
        die();
    }
    $query = "SELECT ip, attempt_time + INTERVAL 1 HOUR AS time FROM
uni_failed_logins WHERE ip LIKE INET_ATON(?) AND now()<attempt_time +
INTERVAL 1 HOUR ORDER BY attempt_time ASC";

    if( $stmt = $mysqli->prepare( $query ) )
    {
        // Bind parameters (s = string, i = int, b = blob, etc)
        $stmt->bind_param('s', $ip);
        $stmt->execute();
        $res = $stmt->get_result();
        $stmt->close();
        $ret = $res->fetch_all(MYSQLI_ASSOC);
```

```
        $now = date("Y-m-d H:i:s");

        if( count($ret) > 4 && $ret[0]['time'] > $now )
        {
            header("Location: ./blocked.php");
            $mysqli->close();
            die();
        }
    }
    else
    {
        echo 'ERROR';
        header("HTTP/1.1 400 Bad Request");
        $mysqli->close();
        die();
    }
}
```

Παρακάτω παρατίθεται κώδικας από το αρχείο ***blocked.php***, μέσα στο οποίο βρίσκεται κώδικας Javascript, με τον οποίο υπολογίζεται και προβάλλεται στον χρήστη πόση ώρα απομένει από την πρώτη αποτυχημένη προσπάθεια σύνδεσης αφότου η IP διεύθυνση του αποκλείστηκε για 1 ώρα και έχει καταγραφεί στον πίνακα ***uni_failed_logins***. Μετά το πέρας αυτού του χρόνου θα επιτραπεί στον χρήστη να έχει πρόσβαση στη σελίδα *"Login"*. Αυτό ελέγχεται κατά τη διαδικασία της αυθεντικοποίησης.

```
<?

require_once './db-config.php';

$ip = $_SERVER['REMOTE_ADDR'];
$query = "SELECT attempt_time + INTERVAL 1 HOUR AS time FROM
uni_failed_logins WHERE ip LIKE INET_ATON(?) ORDER BY attempt_time DESC";

if( $stmt = $mysqli->prepare( $query ) )
{
    // Bind parameters (s = string, i = int, b = blob, etc)
    $stmt->bind_param('s', $ip);
    $stmt->execute();
    $res = $stmt->get_result();
    $stmt->close();
    $response = $res->fetch_all(MYSQLI_ASSOC);
    $response = $response[0];

    $time = $response['time'];
    $str_time = strtotime($time);
}
else
{
```

```
    echo 'ERROR';
    header("HTTP/1.1 400 Bad Request");
    $mysqli->close();
    die();
}
?>
<!DOCTYPE HTML>
<html>
    <head>
        <meta name="viewport" content="width=device-width, initial-
scale=1">
        <style>
            p {
              text-align: center;
              font-size: 30px;
              margin-top: 0px;
            }
        </style>
    </head>
    <body>

        <p id="demo"></p>

        <script>
            var countdown = new Date(<?php echo '"' . $time  . '"';
?>).getTime();

            var ref = setInterval(function() {

                var now = new Date().getTime();
                var interval = countdown - now;
                var hrs = Math.floor( ( interval % ( 1000 * 60 * 60 * 24 )
) / (1000 * 60 * 60) );
                var min = Math.floor( ( interval % ( 1000 * 60 * 60 ) ) /
(1000 * 60) );
                var sec = Math.floor( ( interval % ( 1000 * 60 ) ) / 1000
);

                var message = "Ξαναπροσπάθησε σε " + hrs + " ώρες " + min +
" λεπτά " + sec + " δευτερόλεπτα.";
                document.getElementById("demo").innerHTML = message;

                if( interval < 0 )
                {
                    clearInterval(x);
                    window.location.replace("https://unistudent.eu");
                }
            }, 1000);
        </script>

    </body>
</html>
```

Τέλος, σε ένα κρυφό αρχείο, το οποίο ονομάζεται **.htaccess**, έχει οριστεί να αποτρέπεται η χρήση αιτημάτων DELETE PUT UPDATE.

```
<Limit DELETE PUT UPDATE>
        Deny from all
</Limit>
```

Επίσης, το **.htaccess** ανακατευθύνει όλες τις σελίδες στο HTTPS:

```
RewriteEngine On
RewriteCond %{SERVER_PORT} 80
RewriteRule ^(.*)$ https://www.unistudent.eu/$1 [R,L]
```

# Κεφάλαιο 6. Αξιολόγηση ασφάλειας διαδικτυακών εφαρμογών

## 6.1. Διεξαγωγή ελέγχου

Όταν πραγματοποιηθεί η ανάπτυξη της διαδικτυακής εφαρμογής υπάρχει μία διαδικασία για την εύρεση των ευπαθειών της και την αντιμετώπιση τους. Θα διεξαχθούν έλεγχοι σε αυτήν και αφού γραφτεί η αναφορά, θα επιστρέψει ξανά στην ομάδα ανάπτυξης με στόχο να διορθωθούν οι ευπάθειες. Στο τελικό στάδιο ο penetration tester θα ελέγξει εάν όντως διορθώθηκαν οι ευπάθειες και θα οριστικοποιήσει τη φάση ελέγχου.

Οι μεθοδολογίες του penetration testing είναι αρκετά συγκεκριμένες, αν και ενδεχομένως να διαφέρουν σε λεπτομέρειες ή στη χρήση εργαλείων. Χωρίζεται σε τρεις βασικές κατηγορίες ανάλογα με την αρχική γνώση, την οποία έχει κάποιος σε σχέση με τη σελίδα ή τον οργανισμό. Οι κατηγορίες είναι Black Box, Gray Box και White Box. Επιγραμματικά, στο Black Box ο penetration tester δεν έχει καμία γνώση για τον κώδικα ή τις τεχνολογίες, οι οποίες χρησιμοποιούνται. Στο Gray Box έχει εν μέρει κάποια γνώση, όπως παραδείγματος χάρη, οι διάφοροι τύποι χρηστών ή τα frameworks, τα οποία έχουν χρησιμοποιηθεί. Τέλος, στο White Box υπάρχει πλήρης γνώση για τις τεχνολογίες, πρόσβαση στον κώδικα και στις βασικές ρυθμίσεις. Το White Box είναι ο καλύτερος τρόπος ελέγχου της εφαρμογής, καθώς ο penetration tester έχει τη δυνατότητα να δοκιμάσει επιθέσεις, κάτι το οποίο χωρίς την επίγνωση του συστήματος θα ήταν αδύνατο. Το μειονέκτημα έναντι του Black Box είναι το κόστος του σε χρόνο.

Όλες οι μεθοδολογίες έχουν συγκεκριμένα βήματα. Τα πεδία ελέγχου, τα οποία ορίζονται από τον κάτοχο ή τον υπεύθυνο ασφάλειας της εφαρμογής, είναι τα παρακάτω: **1.** Scoping, **2.** έλεγχος (αξιολόγηση συστήματος) και **3.** αναφορά.

### 6.1.1. Scoping

Σε αυτό το στάδιο, ο δικαιούχος της εφαρμογής θα ορίσει ποια σημεία της επιθυμεί να ελεγχθούν. Οποιαδήποτε κακόβουλη πράξη ή έλεγχος έξω από το scope του penetration tester είναι παράνομη και διώκεται ποινικά. Επίσης, συνήθως σε αυτό το σημείο ορίζεται και το είδος του penetration test (Black Box, White Box ή Gray Box). Όλα τα παραπάνω περιγράφονται αναλυτικά σε σύμβαση, η οποία υπογράφεται από τους εμπλεκόμενους, καλύπτοντας έτσι νομικά τον penetration tester.

**Συγκέντρωση πληροφοριών**

Στη φάση ελέγχου της εφαρμογής (penetration testing) το πιο σημαντικό στάδιο είναι η συγκέντρωση πληροφοριών (ειδικά όταν αναφερόμαστε σε Black Box). Ο penetration tester καλείται να συλλέξει όσες περισσότερες πληροφορίες είναι δυνατόν σε σχέση με την εφαρμογή, όπως τη γλώσσα προγραμματισμού με την οποία έχει αναπτυχθεί, τυχόν βάση δεδομένων την οποία χρησιμοποιεί, διαδικτυακές πόρτες, οι οποίες υπάρχουν και ενδεχομένως να υποδηλώνουν επιπλέον services, χρήστες, directories τα οποία οδηγούν σε άλλες σελίδες, backup αρχεία και frameworks.

Όσες περισσότερες πληροφορίες συλλεχθούν τόσο πληρέστερη θα είναι η ανάλυση, αυξάνοντας έτσι τις πιθανότητες εντοπισμού ευπαθειών. Σε αυτό το στάδιο (Reconnaissance ή Recon) χρησιμοποιούνται συνήθως αυτοματοποιημένα εργαλεία και ο penetration tester ελέγχει manual το περιεχόμενο των HTML σελίδων που θα ανακαλύψει. Πολλές φορές στα σχόλια υπάρχουν πληροφορίες για τη γλώσσα προγραμματισμού, κάποιο username (συνήθως administrator), βάση δεδομένων κ.α. Τα πάντα καταγράφονται σε σημειώσεις και ίσως χρησιμοποιηθούν μετέπειτα.

Γνωστά εργαλεία σε αυτό το στάδιο είναι το nmap, το οποίο ελέγχει για ανοιχτές πόρτες στον server ή και firewalls, τα οποία υπάρχουν, και το dirb το οποίο ελέγχει για directories και άλλες σελίδες. Το αν η χρήση του nmap είναι

νόμιμη ή όχι εξαρτάται από τον ορισμό του πεδίου ελέγχου. Το penetration test εάν αφορά αποκλειστικά στην εφαρμογή που τρέχει συνήθως θα είναι μόνο στην πόρτα 80 (HTTP) ή στην 443 (HTTPS) εάν πραγματοποιείται απομακρυσμένα. Εάν ο ενδιαφερόμενος θέλει να ελεγχθεί ο server ολοκληρωτικά, τότε η χρήση nmap είναι νόμιμη, καθώς ο penetration tester υποχρεούται να ελέγξει και άλλες υπηρεσίες, οι οποίες τρέχουν παράλληλα.

Το dirb χρησιμοποιείται με παράμετρο το domain/ IP της εφαρμογής, την πόρτα που τρέχει και κάποιο wordlist. Σε αυτό το wordlist εμπεριέχονται συνηθισμένα ονόματα σελίδων, τα οποία χρησιμοποιούνται από προγραμματιστές, όπως index.html, admin.html, login.html κ.α. Το εργαλείο αυτό κάνει αίτημα προς αυτήν τη σελίδα και εάν λάβει απάντηση ότι υπάρχει, το εμφανίζει· διαφορετικά, το αγνοεί. Ο penetration tester αφού συγκεντρώσει όλες τις σελίδες, θα τις ελέγξει εκ νέου σε σχέση με το περιεχόμενο και τον σκοπό τους.

Ιδιαίτερη σημασία πρέπει να δοθεί στην υπερπληροφορία. Από τα εργαλεία παράγεται πολύ μεγάλος όγκος πληροφορίας, τον οποίον είναι αδύνατον ένας ή και μία ομάδα ανθρώπων να ελέγξει. Αφήνεται στην εμπειρία του penetration tester να αναγνωρίσει ποια από αυτά τα ευρήματα είναι σημαντικά και πρέπει να αξιοποιηθούν και ποια όχι. Φυσικά απαιτείται εμπειρία και γνώση. Σε καμία περίπτωση δεν πρέπει να αγνοείται κάποιο κομμάτι ή να μην του δίνεται προσοχή.

### 6.1.2. Έλεγχος

Αφού ολοκληρωθεί το πρώτο στάδιο και ο penetration tester έχει στη διάθεση του το απαραίτητο υλικό, θα προχωρήσει στους ελέγχους. Ενδεχομένως να περιηγηθεί στις διάφορες σελίδες για να κατανοήσει εις βάθος τον λόγο ύπαρξης της εφαρμογής και να έχει τη δυνατότητα να εντοπίσει ευπάθειες.

Ιδιαίτερη προσοχή χρήζουν φόρμες σύνδεσης ή εγγραφής και μπάρες αναζήτησης, καθώς ενδεχομένως να υπάρχουν ευπάθειες SQL Injection,

απομακρυσμένη εκτέλεση κώδικα RCE (**R**emote **C**ode **E**xecution) και XSS. Με αυτά ο επιτιθέμενος ενδεχομένως να καταφέρει να πάρει κάποιο ρόλο, έστω και απλού χρήστη, και να προσπαθήσει να ανέβει σε ανώτερο (privilege escalation). Είναι συνηθισμένο σε αυτό το στάδιο να χρησιμοποιείται manual και αυτοματοποιημένος έλεγχος.

#### 6.1.2.1. Αυτοματοποιημένο Penetration Test

Όπως έχει αναφερθεί παραπάνω, για την εύρεση SQL Injection ευπαθειών, XSS και RCE θα άξιζε κάποιος να χρησιμοποιήσει αντίστοιχα εργαλεία. Για SQL (**S**tructured **Q**uery **L**anguage) επιθέσεις πολύ γνωστό είναι το Sqlmap και για γενικότερη χρήση υπάρχει και το Nikto. Άλλα εργαλεία είναι τα Metasploit και ZAP.

Με το Sqlmap ελέγχεται μία μεγάλη συλλογή εισόδων, η οποία ενδεχομένως να προκαλέσει μη ορθή συμπεριφορά στη βάση. Το εργαλείο ανιχνεύει και εκτυπώνει τις εισόδους, τις οποίες χρειάζεται ο επιτιθέμενος.

Το Metasploit χρησιμοποιείται για την εύρεση και εκμετάλλευση αδυναμιών, καθώς διαθέτει μία μεγάλη συλλογή από βιβλιοθήκες, εργαλεία και exploits. Έχει ενσωματωμένο το nmap και υπάρχει πολλή υποστήριξη από κοινότητες και ερασιτέχνες, οι οποίοι ασχολούνται με τον χώρο της κυβερνο-άμυνας. Επίσης, υπάρχει και το Armitage, το οποίο ουσιαστικά είναι το Metasploit με γραφικά.

Το αυτοματοποιημένο penetration test από πολλούς θεωρείται κακή πρακτική, καθώς αφήνει πολλά logs στον server και ειδικά τη σύγχρονη εποχή είναι πολύ πιθανό να διακόψει την κίνηση κάποιο IDS/IPS (**I**ntrusion **D**etection **S**ystem/ **I**ntrusion **P**revention **S**ystem) πριν προλάβει να φτάσει στην εφαρμογή. Επίσης, ένα ακόμα μειονέκτημα του αυτοματοποιημένου ελέγχου είναι ότι πολλές εφαρμογές θέλουν συγκεκριμένα πακέτα HTTP· διαφορετικά, θα αποτύχουν.

Το πιο σημαντικό πλεονέκτημα είναι η εξοικονόμηση χρόνου και το ότι ο penetration tester έχει τη δυνατότητα να εξετάσει λειτουργίες της εφαρμογής, τις οποίες κανένα εργαλείο δεν θα ήταν ικανό να εξετάσει.

### 6.1.2.2. Manual Penetration Test

Το manual penetration test απαιτεί περισσότερο χρόνο και πιο βαθιά γνώση σε σχέση με το πως δουλεύει το πρωτόκολλο HTTP και οι εφαρμογές καθώς επίσης και δεξιότητες, όπως η υπομονή, η παρατηρητικότητα και η δημιουργικότητα. Το εργαλείο το οποίο χρησιμοποιείται ευρέως εδώ είναι το Burp Suite.

Σε ένα HTTP πακέτο υπάρχουν οι Headers, οι οποίοι περιγράφουν τη μέθοδο (π.χ. get, post, trace), τον browser ο οποίος έκανε το αίτημα, το βασικό domain της εφαρμογής κ.α.

Το Burp Suite είναι ένας proxy, όπου ρυθμίζοντας κατάλληλα τον browser μας, οδηγούμε την ροή των δεδομένων να “περνάει” μέσα από αυτόν. Ο proxy “δεσμεύει” το πακέτο και δεν το αποστέλλει, μέχρι να το αποφασίσει ο penetration tester.


**Εικόνα 22. Burp Suite**

Με το Burp Suite δίνεται η δυνατότητα επεξεργασίας των headers και ο έλεγχος για τη συμπεριφορά της εφαρμογής με μη αναμενόμενο πακέτο. Κλασσικό παράδειγμα αποτελεί το HTTP basic authentication, το οποίο, εάν δεν είναι σωστά ορισμένη η ασφάλεια, αλλάζοντας τον header από GET σε OPTIONS θα οδηγήσει σε Broken Authentication και θα είναι σε θέση κάποιος να δει περιεχόμενο, το οποίο κανονικά δεν θα έπρεπε. Μία επίθεση, η οποία θέλει τη χρήση Burp Suite είναι η smuggling. Εάν ο server δεν είναι σωστά ρυθμισμένος και λάβει ένα HTTP πακέτο, όπου ο επιτιθέμενος έχει προσθέσει χαρακτήρες στο τέλος του και έχει αλλάξει το μέγεθος του, ενδέχεται να προκαλέσει απροσδιόριστη συμπεριφορά.

Υπάρχουν επίσης και πολλές επικεφαλίδες, οι οποίες αν και δεν είναι τόσο γνωστές, με τη μελέτη του HTTP πρωτοκόλλου και τη σωστή χρήση τους είναι ικανές να οδηγήσουν σε πλήθος επιθέσεων. Παράδειγμα τέτοιων είναι το X-Origin-URL, το οποίο ενδεχομένως να οδηγήσει σε Broken Access Control και το X-Forwarded-For, το οποίο ενδεχομένως να “πείσει” τον server για τη διεύθυνση προέλευσης του πακέτου· κάτι το οποίο όμως σπάνια συναντάται στην πράξη.

Μία πολύ σημαντική ευπάθεια έχει να κάνει με το αν είναι σε θέση ο penetration tester να ανεβάσει (upload) αρχεία στην εφαρμογή. Όποτε υπάρχει τέτοια δυνατότητα, εξετάζεται σε βάθος εάν ο χρήστης είναι σε θέση να ανεβάσει διαφορετικού είδους αρχεία από αυτά, τα οποία αναμένεται. Παραδείγματος χάρη, το σύνηθες μορφότυπο αρχείων για εικόνες είναι: .jpg/ .png, ενώ για έγγραφα: .pdf/ .odt/ .docx. Με τη χρήση του Burp Suite είναι ικανός κάποιος να πείσει την εφαρμογή, χρησιμοποιώντας τα κατάλληλα magic bytes, ότι το αρχείο, το οποίο ανέβασε, είναι του αναμενόμενου μορφοτύπου ενώ στην πραγματικότητα ενδεχομένως να είναι ακόμα και αρχείο γλώσσας προγραμματισμού.

Στο ίδιο το Burp Suite υπάρχουν αυτοματοποιημένα εργαλεία για Brute Force επιθέσεις, decoders κ.α., τα οποία ενδέχεται να βοηθήσουν και να εξοικονομήσουν χρόνο κατά τη διεξαγωγή του ελέγχου. Γνωστά είναι ο Decoder

- ο οποίος χρησιμοποιείται για την κωδικοποίηση και αποκωδικοποίηση κωδικών και strings - και ο Intruder, ο οποίος εξυπηρετεί στην αυτοματοποίηση επιθέσεων, όπως SQL Injection **(PortSwigger, 2020)**.

Ακόμα ένα εργαλείο – browser, όχι όμως τόσο γνωστό, είναι το mantra-ff της OWASP, το οποίο δίνει τη δυνατότητα στον επιτιθέμενο να βλέπει live τα αιτήματα μεταξύ client – server. Συνδυάζεται με πολλά extensions, τα οποία διευκολύνουν το έργο του.

Με το manual penetration test έχει τη δυνατότητα κάποιος να ελέγξει την εφαρμογή σε μεγαλύτερο βάθος και να αποκτήσει μία πιο σφαιρική άποψη σε σχέση με την ασφάλεια της. Το μειονέκτημα, όπως έχει αναφερθεί παραπάνω, είναι ότι απαιτείται περισσότερος χρόνος, εμπειρία και βαθιά γνώση σε διάφορα πεδία.

### 6.1.3. Αναφορά

Αφού ολοκληρωθεί και ο έλεγχος, ο penetration tester καλείται να συντάξει αναφορά με τις αδυναμίες τις οποίες βρήκε, τις επιθέσεις τις οποίες κατάφερε να διεκπεραιώσει με επιτυχία, την κρισιμότητα της κάθε επίθεσης και τα σημεία, τα οποία βρέθηκαν.

Η κρισιμότητα αλλάζει σε κάθε εφαρμογή και σε κάθε εταιρεία/ οργανισμό. Σχετίζεται άμεσα με την αξία της πληροφορίας και τη σημαντικότητα της λειτουργίας, η οποία επηρεάζεται από την επίθεση. Συνήθως, την τελική αναφορά για τη σημαντικότητα (risk assessment) την δίνει κάποιος σύμβουλος ασφάλειας, ο οποίος είναι υπάλληλος του οργανισμού, ο οποίος εξετάζεται, και έχει άμεση σχέση με το αντικείμενο του.

Τέλος, η αναφορά θα μελετηθεί από την ομάδα προγραμματιστών, οι οποίοι ανέπτυξαν την εφαρμογή, και με τη σειρά τους θα διορθώσουν (patch) τις

αδυναμίες. Είναι πολύ σημαντικό να υπάρχει στην ομάδα ανάπτυξης κάποιος, ο οποίος να κατέχει γνώσεις από το πεδίο της ασφάλειας και να είναι ικανός να “μεταφράσει” σε κώδικα το report του penetration tester.

Μόλις ολοκληρωθούν οι διορθώσεις, η εφαρμογή θα εξετασθεί ξανά από τον penetration tester αναφορικά με τις συγκεκριμένες ευπάθειες. Δεν αποτελεί μία χρονοβόρα διαδικασία, καθώς ο penetration tester απλώς θα επαναλάβει τα βήματα και τις επιθέσεις όπως προηγουμένως. Εάν δει ότι η εφαρμογή είναι σωστά ενημερωμένη και ότι δεν υπάρχει κάποιο πρόβλημα, θα ενημερώσει ότι ο έλεγχος πρέπει να τερματίσει. Αυτό το στάδιο σηματοδοτεί και την έναρξη διαθεσιμότητας της εφαρμογής.

## 6.2. Τεχνικές ενίσχυσης ασφάλειας

Υπάρχουν διάφορες τεχνικές και μέθοδοι σχετικά με την ασφάλεια μίας σελίδας. Αν και ξεφεύγουν από το scope της "ασφάλειας εφαρμογών" αξίζει να αναφερθούν και να περιγραφούν, καθώς έχουν άμεση σχέση.

### 6.2.1. Back-end κώδικας

Όπως αναφέρθηκε και σε προηγούμενο κεφάλαιο, οι προγραμματιστές έχουν τη μεγαλύτερη ευθύνη για την ασφάλεια. Υπάρχουν πρακτικές και βιβλιοθήκες, οι οποίες εξυπηρετούν αυτόν τον σκοπό. Ένας προγραμματιστής δεσμεύεται να τις έχει υπόψιν του ανάλογα με τη γλώσσα προγραμματισμού, την οποία χρησιμοποιεί.

**Παραδείγματα**

Παρακάτω θα αναφερθούν οι αδυναμίες και αναφορικά με αυτές, θα περιγραφούν κάποιες βασικές αρχές και τρόποι αποφυγής τους.

**Injection:** ο καλύτερος τρόπος να αποφεύγεται είναι ο έλεγχος των δεδομένων εισόδου (sanitation). Τα δεδομένα θα πρέπει να ελέγχονται για κάθε κατηγορία αδυναμιών. Παρακάτω φαίνονται επιγραμματικά μερικές γνωστές κακόβουλες είσοδοι:

- **SQL Injection:** ‘, “, #, -, or, and, union, select, delete
- **XSS:** <, >, /, ;, script, alert
- **RCE:** /, ;, ‘, echo
- **XXE:** [, ], !, external

Εάν ο αλγόριθμος εντοπίσει τέτοιους χαρακτήρες και λέξεις θα πρέπει είτε να μην εκτελέσει το query είτε να το παραμετροποιήσει, έτσι ώστε να είναι

ακίνδυνο για την εφαρμογή. Συνήθως, εμφανίζονται μηνύματα λάθους, τα οποία ειδοποιούν όταν κάποιος χαρακτήρας απαγορεύεται να χρησιμοποιηθεί.

**Insecure Deserialization:** η γενική συμβουλή είναι να αποφεύγεται. Πολύ σπάνια, η χρήση του είναι μονόδρομος και εάν είναι απαραίτητο πρέπει να γίνεται έλεγχος των δεδομένων, τα οποία ελήφθησαν.

**Sensitive Data Exposure:** είναι αρκετά δύσκολο να διορθωθεί. Κατά τη σχεδίαση και δημιουργία μίας σελίδας, οι προγραμματιστές χρησιμοποιούν σχόλια, backup αρχεία και δοκιμαστικά directories, τα οποία ξεχνούν στη φάση της παραγωγής. Πρέπει να ελέγχονται και να αφαιρούνται ή να μην γράφονται αρχικά. Σημειωματάρια, όπως το CheeryTree, χρησιμοποιούνται από πολλούς, προκειμένου να αποφεύγονται τέτοια λάθη.

**Broken Authentication:** η χρήση ορθών λογικών συναρτήσεων πιστοποίησης ταυτότητας είναι απαραίτητη προϋπόθεση για την αποφυγή τέτοιων αδυναμιών. Πολύ συχνά, οι προγραμματιστές δημιουργούν λογικά σφάλματα σε κομμάτια κώδικα ή βασίζονται σε τιμές, οι οποίες είναι εύκολα επεξεργάσιμες. Όσο πιο πολλοί έλεγχοι πραγματοποιούνται, τόσο πιο ασφαλείς θα είναι οι εφαρμογές. Με στόχο την αποφυγή Broken Authentication χρησιμοποιείται η τεχνική Two-Factor Authentication, η οποία θα περιγραφεί ξεχωριστά.

**Broken Access Control:** η χρήση σωστά ορισμένων sessions και πολλοί διαφορετικοί έλεγχοι κατά την πρόσβαση ενός χρήστη σε κάποια σελίδα ή συνάρτηση της εφαρμογής θα λύσουν το πρόβλημα αυτό. Σε αυτές τις περιπτώσεις, χρησιμοποιείται και το LDAP (**L**ightweight **D**irectory **A**ccess **P**rotocol), το οποίο θα περιγραφεί ξεχωριστά παρακάτω.

**Security Misconfiguration:** προγραμματιστές και System Administrators πρέπει να εξασφαλίζουν τη χρήση σωστών ρυθμίσεων και πρακτικών.

**Using Vulnerable Components:** η ομάδα ανάπτυξης πρέπει πάντα να ζητάει updates σε βιβλιοθήκες, οι οποίες χρησιμοποιούνται εντός του κώδικα ή ακόμα και διαφορετικών software. Εάν ένα ευπαθές κομμάτι ή βιβλιοθήκη πρέπει να χρησιμοποιηθεί και δεν υπάρχει πρόσφατη ασφαλή έκδοση του, πρέπει να μελετηθεί εις βάθος η αδυναμία του και να επιλυθεί από τον developer καθώς και από τον System Administrator σε επίπεδο εφαρμογής.

**Insufficient Logging and Monitoring:** η σημαντικότητα του να κρατούνται αρχεία, τα οποία περιγράφουν διάφορες καταστάσεις κατά την εκτέλεση της εφαρμογής καθώς και τον έλεγχο τους, έχει περιγραφεί προηγουμένως. Ανεξάρτητα από τα logs τα οποία θα κρατηθούν από άλλους - π.χ. διαχειριστές - οι προγραμματιστές πρέπει να φτιάχνουν αντίστοιχες συναρτήσεις μέσα στην εφαρμογή. Παραδείγματος χάρη, όταν ένας χρήστης προσπαθήσει να κάνει login, είτε ήταν αποτυχημένο είτε όχι, θα πρέπει να καταγραφεί. Στην Python θα έμοιαζε κάπως έτσι:

```
def keep_logs(username, IP, time ):
f = open('logs.txt', 'a')
string =username+":"+IP+":"+time
f.write(string)
f.close()
```

Ο παραπάνω κώδικας αποθηκεύει κάθε προσπάθεια σύνδεσης. Εάν παρατηρηθεί κάποια ύποπτη δραστηριότητα, ο υπεύθυνος θεμάτων ασφάλειας θα πρέπει να ελέγξει τα logs σε ένα εύρος ημερομηνιών και να διαπιστωθεί εάν κάποιος προσπάθησε Brute Force επιθέσεις ή ακόμα κι αν κατάφερε να αποκτήσει πρόσβαση.

### 6.2.2. Waf (**W**eb **A**pplication **F**irewall)

Μία τάση αυτής της εποχής φαίνεται να είναι το Web Application Firewall. Εξαιτίας της υψηλής απόδοσης των επεξεργαστών και των μνημών οι εφαρμογές έχουν ένα δικό τους firewall σε υψηλό επίπεδο (επίπεδο εφαρμογής), το οποίο τις προστατεύει από επιθέσεις όπως SQL Injection, XSS, Cross-site forgery κ.α.

Κρίσιμα κομμάτια του κώδικα, όπως φόρμες σύνδεσης, εγγραφής, μπάρες αναζήτησης και ανέβασμα αρχείων, μόλις παίρνουν κάποια δεδομένα διακόπτουν την κανονική τους ροή και επιστρέφουν την εκτέλεση κώδικα στο firewall τους, προκειμένου να ελέγξει αυτήν την είσοδο.

Αυτό εξυπηρετεί στη συντήρηση και κατανόηση του κώδικα, καθώς υπάρχει μία κεντρική συνάρτηση επεξεργασίας για διάφορα ζητήματα ασφάλειας **(Cloudfare, 2020)**.

### 6.2.3. Two-Factor Authentication

Το Two-Factor Authentication (διπλός έλεγχος ταυτότητας) είναι ένας τρόπος να γίνει πιο ισχυρή η ταυτοποίηση του χρήστη. Σε εφαρμογές οι οποίες προσφέρουν “ευαίσθητες” υπηρεσίες, όπως η μεταφορά χρημάτων, αλλά και σε κοινωνικά δίκτυα όπως Facebook, Instagram κ.α. συναντάται συχνά αυτή η τεχνική.

Αφού ο χρήστης εισάγει το username και το password του και επιβεβαιωθεί η εγκυρότητα των στοιχείων του από την εφαρμογή, του αποστέλλεται ένας δεύτερος κωδικός είτε με E-mail είτε με μήνυμα στο προσωπικό του κινητό, τον οποίο πρέπει να εισάγει στην εφαρμογή. Μόλις ολοκληρωθεί και αυτή η επαλήθευση, του επιτρέπεται η πρόσβαση.

Με αυτήν την τεχνική, για να αποκτήσει κάποιος πρόσβαση στα δεδομένα ενός χρήστη, θα πρέπει αφενός να έχει στη διάθεση του το password και το username του και αφετέρου να έχει πρόσβαση στο E-mail του ή ακόμα και στο κινητό του. Για διπλό έλεγχο ταυτότητας δύναται να χρησιμοποιηθούν και βιομετρικά στοιχεία, όπως παραδείγματος χάρη το δακτυλικό αποτύπωμα, η αναγνώριση του προσώπου ή της ίριδας του ματιού.

### 6.2.4. Κρυπτογραφία

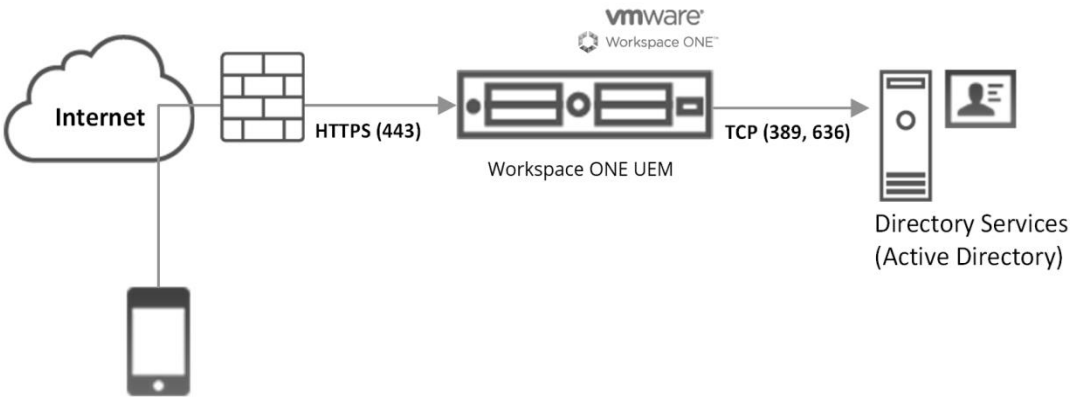
Η κρυπτογραφία σήμερα έχει διάφορες χρήσεις. Η πιο βασική από αυτές είναι η προστασία των δεδομένων. Όσο ασφαλής και να είναι μία εφαρμογή, ποτέ δεν είναι σε θέση κάποιος να είναι απόλυτα βέβαιος σε σχέση με την ακεραιότητα των δεδομένων της.

Πολλές εφαρμογές αποθηκεύουν δεδομένα, όπως username και password, στη βάση με hash μορφή. Το hash δεν είναι 1 – 1 συνάρτηση. Αυτό σημαίνει ότι γνωρίζοντας το hash δεν έχει κάποιος τη δυνατότητα να επιστρέψει στο αρχικό μήνυμα. Με αυτόν τον τρόπο, ούτε οι ίδιοι οι προγραμματιστές δεν είναι σε θέση να τα γνωρίζουν. Κατά την εγγραφή, ο χρήστης εισάγει username και password και αυτά κωδικοποιούνται με κάποια hash συνάρτηση, όπως md5 ή sha256. Όταν ο χρήστης προσπαθήσει να συνδεθεί και δώσει τα συνθηματικά του, αυτά θα κωδικοποιηθούν και θα ελεγχθούν με τα αντίστοιχα στη βάση. Τα υπόλοιπα δεδομένα, τα οποία πιθανόν να απαιτούνται, όπως όνομα, επίθετο, E-mail και κινητό τηλέφωνο θα κρυπτογραφηθούν με τον κωδικό του, τον οποίο ουσιαστικά μόνο εκείνος έχει στη διάθεση του.

Μία ακόμα χρήση της κρυπτογραφίας σε μία εφαρμογή είναι η κρυπτογράφηση όλης της κίνησης. Με αυτόν τον τρόπο επιθέσεις σε επίπεδο δικτύου όπως Man In The Middle Attack, όπου κάποιος ο οποίος είναι στο ίδιο τοπικό δίκτυο με έναν άλλον χρήστη έχει τη δυνατότητα να “παρακολουθήσει” την κίνηση και τα δεδομένα του τελευταίου, αποφεύγονται. Όλη η κίνηση είναι κρυπτογραφημένη με ισχυρά κλειδιά και χρησιμοποιώντας RSA.

### 6.2.5. LDAP (**L**ightweight **D**irectory **A**ccess **P**rotocol**)**

Το LDAP είναι ένα πρωτόκολλο για την πρόσβαση στην πληροφορία. Με τη χρήση του, δύναται κάποιος να ορίσει ομάδες χρηστών, όπου η κάθε μία έχει συγκεκριμένα δικαιώματα πρόσβασης σε καταλόγους. Με αυτόν τον τρόπο προσφέρεται στην εφαρμογή ακόμα ένα επίπεδο ασφάλειας, ειδικά για αδυναμίες τύπου Broken Access Control **(Wikipedia, 2020)**.



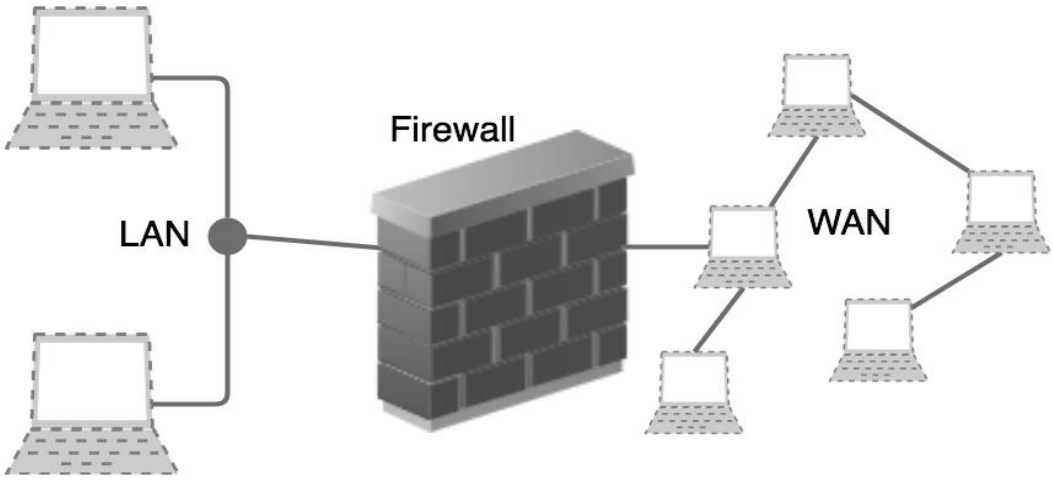

**Εικόνα 23. LDAP (πηγή: https://docs.vmware.com/en/VMware-Workspace-ONE-UEM/services/UEM_ConsoleBasics/GUID-AWT-AD-LDAP USERAUTHENTICATIONTYPES.html)**

### 6.2.6. Firewall

Το firewall (τείχος προστασίας) είναι η πρώτη γραμμή άμυνας. Δεν έχει τόσο σχέση με την εφαρμογή όσο με τον ίδιο τον server. Το firewall είναι ένα set από κανόνες και πολιτικές, τα οποία υπάρχουν στον εξυπηρετητή. Είναι πιθανό το firewall να επιτρέπει συγκεκριμένο τύπο κίνησης, να απαγορεύει κίνηση από κάποιες IPs ή και ολόκληρες χώρες καθώς και να κρατάει κλειστές διαδικτυακές θύρες, τις οποίες δεν χρειάζεται.

Μόλις έρχεται εισερχόμενη κίνηση στο δίκτυο, ελέγχεται πρώτα από το firewall. Αυτό θα ελέγξει τις επικεφαλίδες του πακέτου και με την προϋπόθεση ότι εκείνες δεν παραβιάζουν την πολιτική και τους κανόνες, θα αναδρομολογήσει το πακέτο στον σωστό εξυπηρετητή.

Διαθέσιμο δωρεάν και open source firewall είναι το pfsense, το οποίο χρησιμοποιείται από εξειδικευμένο προσωπικό και μη.




**Εικόνα 24. Firewall (πηγή: https://www.geeksforgeeks.org/introduction-of-firewall-in-computer-network/)**

### 6.2.7. IDS/IPS (**I**ntrusion **D**etection **S**ystem/ **I**ntrusion **P**revention **S**ystem)

Το IDS/IPS είναι ένα σύστημα, το οποίο ελέγχει όλη την κίνηση του δικτύου, αναγνωρίζει περιστατικά ασφάλειας, τα καταγράφει και τα εμποδίζει από το να πάνε στην εφαρμογή. Επίσης, χρησιμοποιείται και για να ελέγξει εάν κάποιος εντός του εσωτερικού δικτύου παραβιάζει πολιτικές ασφάλειας, εκθέτοντας συνολικά το δίκτυο ή τα δεδομένα σε κινδύνους.

Τα IDS συνήθως λειτουργούν αναγνωρίζοντας συγκεκριμένα patterns εντός κάποιου πακέτου/ αιτήματος. Το IDS είναι υπεύθυνο για την αναγνώριση κακόβουλης κίνησης και το IPS για την εμπόδιση της.

Αν και αποτελούν πολύ σημαντικό πλεονέκτημα στη συνολική ασφάλεια του δικτύου και κατ' επέκταση της εφαρμογής, τα μειονεκτήματα είναι ότι αφενός θέλουν συνεχή ενημέρωση, με νέους κανόνες και patterns, καθώς και υποστήριξη από κάποιον διαχειριστή και αφετέρου αυξάνουν την καθυστέρηση του δικτύου. Σαφώς αυτά είναι μηδαμινά σε σχέση με αυτά, τα οποία προσφέρουν.

Τέλος, αξίζει να σημειωθεί ότι αν και αποτελούν τη βασική γραμμή άμυνας και ότι ένα πλήρως ενημερωμένο και άρτια λειτουργικό τέτοιο σύστημα είναι ικανό να προστατεύσει την ομαλή λειτουργία οποιασδήποτε εφαρμογής, δεν πρέπει να παραμελείται η ίδια η ασφάλεια της. Παρατηρείται το φαινόμενο ότι όσο πιο αποδοτικά γίνονται τα IDS/IPS τόσο εξασθενεί ο έλεγχος και η βελτίωση ασφάλειας της εφαρμογής. Και τα δύο συνδυαστικά θα έχουν το βέλτιστο δυνατό αποτέλεσμα. Σε περίπτωση κάποιας επίθεσης - η οποία δεν έχει ανακαλυφθεί μέχρι τότε (zero day exploit) - εάν αποτύχει το ένα, η επίθεση θα πρέπει να εντοπιστεί από το άλλο.

Παραδείγματος χάρη, ένα IDS ίσως να μην εντοπίσει ένα query για SQL Injection· η εφαρμογή ενδεχομένως όμως να καταφέρει να το αναγνωρίσει και να το διαχειριστεί με τέτοιο τρόπο, έτσι ώστε να μην επηρεαστεί η ακεραιότητα της.

Σκοπός κάθε συστήματος είναι να προσφέρονται πολλαπλά επίπεδα ασφάλειας, ακόμα και για την ίδια αδυναμία. Ο συνδυασμός ενός firewall, IDS/IPS με έναν σωστά γραμμένο κώδικα, χωρίς να αφήνει περιθώρια για ευπάθειες, υπόσχεται υψηλό επίπεδο ασφάλειας.

Όλα αυτά θα πρέπει να συνοδεύονται από επαρκή logs και κάποια ομάδα, η οποία να επιβλέπει σε live-time διάφορα περιστατικά. Τέτοιες ομάδες ανήκουν στην κατηγορία Blue Teams του χώρου κυβερνο-άμυνας. Οι ειδικοί ονομάζονται Cyber Security Analysts και είναι υπεύθυνοι για την επιτήρηση της κίνησης του δικτύου, την αντιμετώπιση περιστατικών, την ανάλυση ύποπτων κινήσεων και την εξασφάλιση της ακεραιότητας της εφαρμογής.

Γνωστό IDS είναι το Snort, το οποίο συνδυάζεται άψογα με pfsense. Πιθανόν όμως να αντικατασταθεί ολοκληρωτικά από το Suricata μιας που το τελευταίο χρησιμοποιεί multi-threading και έχει τη δυνατότητα να εξυπηρετεί ταυτόχρονα πολλαπλά αιτήματα, βελτιώνοντας έτσι τη συνολική αποδοτικότητα του συστήματος **(Juniper networks, 2020)**.

# Κεφάλαιο 7. Αξιολόγηση ασφάλειας εφαρμογής *“UniStudent”*

## 7.1. Penetration Test Report

*Έλεγχος ασφάλειας εφαρμογής **UniStudent**.*

**Πληροφορίες εγγράφου**

**Document Title:** UniStudent Penetration Test Report

**Project Title:** UniStudent

**Author:** Ntina B.

**Penetration tester:** Ntina B.

**Reviewed by:**

**Approved by:**

**Classification:** Confidential

**Date:** 27/02/2021

**Περιγραφή ελέγχου**

Διεξήχθη penetration test για την εφαρμογή ***“UniStudent”***. Το report αυτό περιέχει όλα τα βήματα και τις τεχνικές που εφαρμόστηκαν με στόχο την επιτυχή διεξαγωγή του ελέγχου των αδυναμιών της εφαρμογής.

Το penetration test της εφαρμογής βασίστηκε στις OWASP Top 10 αδυναμίες διαδικτυακών εφαρμογών. Οποιαδήποτε αδυναμία βρέθηκε κατά τη διεξαγωγή του ελέγχου ασφάλειας της εφαρμογής έχει αναλυθεί παρακάτω.

Ο πίνακας 1 παρουσιάζει συνοπτικά τις επιθέσεις, οι οποίες έχουν πραγματοποιηθεί, καθώς και τα αυτοματοποιημένα εργαλεία, τα οποία έχουν χρησιμοποιηθεί σε κάθε επίθεση.

| ***Επιθέσεις εφαρμογής "UniStudent"*** | | | | |
|---|---|---|---|---|
| **OWASP Top 10 αδυναμίες διαδικτυακών εφαρμογών** | | **Αυτοματοποιημένα εργαλεία** | | |
| | | **Sqlmap** | **Nikto** | **Gobuster** |
| Injection | ✓ | ✓ | | |
| Broken Authentication | ✓ | | ✓ | |
| Sensitive Data Exposure | ✓ | | ✓ | |
| XML External Entities (XXE) | - | - | - | |
| Broken Access Control | ✓ | | ✓ | |
| Security Misconfiguration | ✓ | | ✓ | |
| Cross Site Scripting (XSS) | ✓ | | ✓ | |
| Insecure Deserialization | - | - | - | |
| Using Components with Known Vulnerabilities | - | - | - | |
| Insufficient Logging and Monitoring | - | - | - | |
| **Επιπρόσθετες Αδυναμίες** | | | | |
| Πιθανές άλλες αδυναμίες | ✓ | | | ✓ |

**Πίνακας 1. Πίνακας επιθέσεων εφαρμογής *"UniStudent"***

Οι παρακάτω επιθέσεις δεν έχουν πραγματοποιηθεί, καθώς :

- η επίθεση **X**ML E**x**ternal **E**ntities (XXE) απαιτεί την ύπαρξη XML κώδικα.
- η επίθεση **Insecure Deserialization** απαιτεί την ύπαρξη αντικειμενοστραφούς προγραμματισμού.
- η επίθεση **Using Vulnerable Components** απαιτεί την ύπαρξη σύνθετου κώδικα (π.χ. έτοιμες βιβλιοθήκες/ συναρτήσεις).
- η επίθεση **Insufficient Logging and Monitoring** απαιτεί την ύπαρξη αρχείων logs.

Η διεξαγωγή του ελέγχου ασφάλειας έγινε αρχικά Black Box και στη συνέχεια White Box. Με αυτόν τον τρόπο εξασφαλίστηκε ότι σε οποιαδήποτε περίπτωση βρέθηκαν αδυναμίες σχετικά με την εφαρμογή.

**Penetration Test – Black Box**

Στην πρώτη φάση του ελέγχου ασφάλειας της εφαρμογής θεωρήθηκε ότι το penetration test είναι Black Box. Αυτό σημαίνει ότι ο penetration tester αγνοεί οποιαδήποτε πληροφορία σε σχέση με την εφαρμογή.

**Penetration Test – White Box**

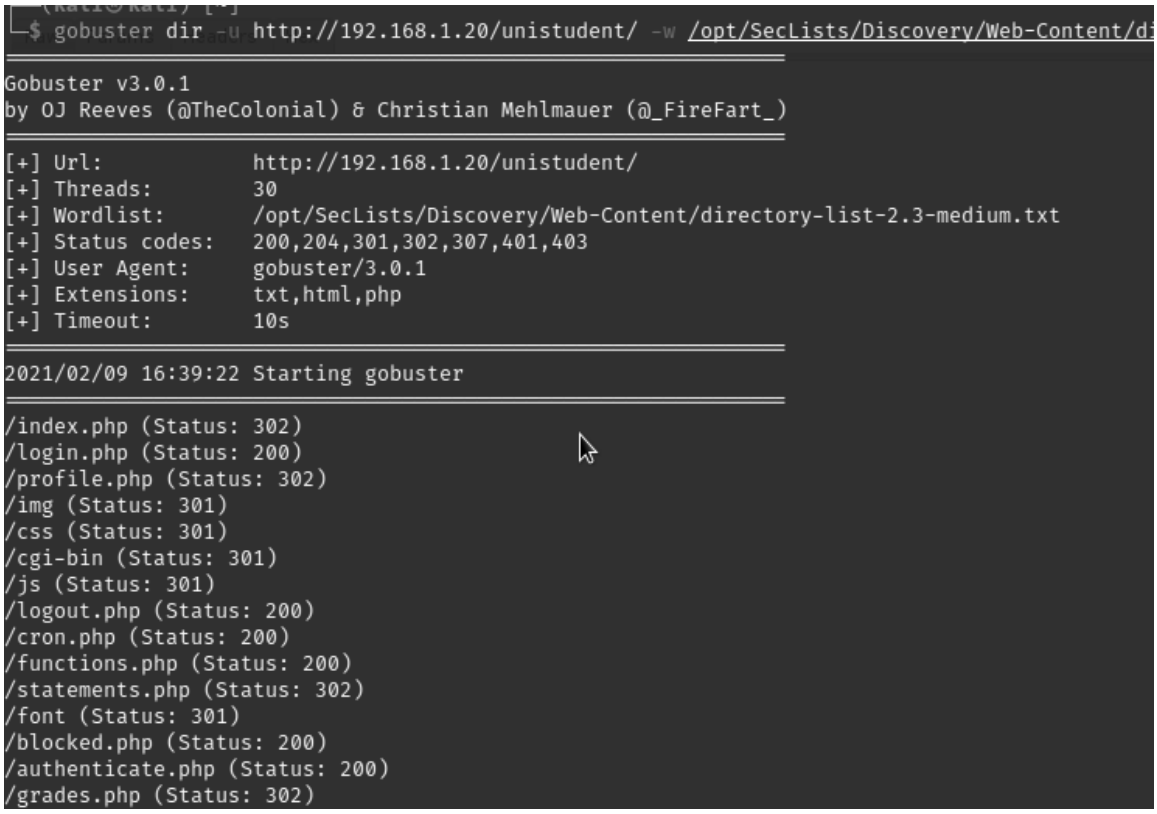
Μετέπειτα, η διεξαγωγή του ελέγχου πραγματοποιήθηκε σε White Box. Στο White Box είναι συνηθισμένο ο penetration tester να έχει πρόσβαση σε κώδικα καθώς επίσης και σε κάποιον χρήστη (username, password), έτσι ώστε να ελέγξει εάν του δίνονται παραπάνω δυνατότητες ως χρήστης ή ακόμα και ως διαχειριστής.

Χρησιμοποιήθηκαν τα παρακάτω αυτοματοποιημένα εργαλεία: **1.** Gobuster, **2.** Sqlmap, **3.** Nikto.

Η IP του server, όπου τρέχει η εφαρμογή, είναι στην τοπική διεύθυνση: *192.168.1.20* και το scope του ελέγχου είναι στην πόρτα 80, όπου τρέχει το HTTP service. Η διεύθυνση URL είναι: *192.168.1.20/unistudent*.

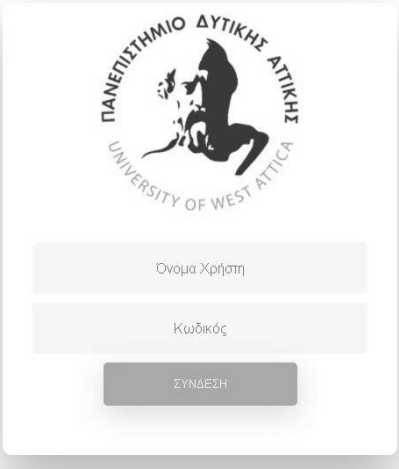
**Ευρήματα ελέγχου**

Με στόχο να συλλεχθούν πληροφορίες σε σχέση με την εφαρμογή και την έκταση της χρησιμοποιήθηκε αρχικά ένα αυτοματοποιημένο πρόγραμμα έτσι ώστε να ανακαλυφθούν διάφορες επεκτάσεις, οι οποίες ενδεχομένως να υπάρχουν. Το εργαλείο αυτό ονομάζεται Gobuster και παρακάτω φαίνονται τα αποτελέσματα του ***(Εικόνα 25)***.



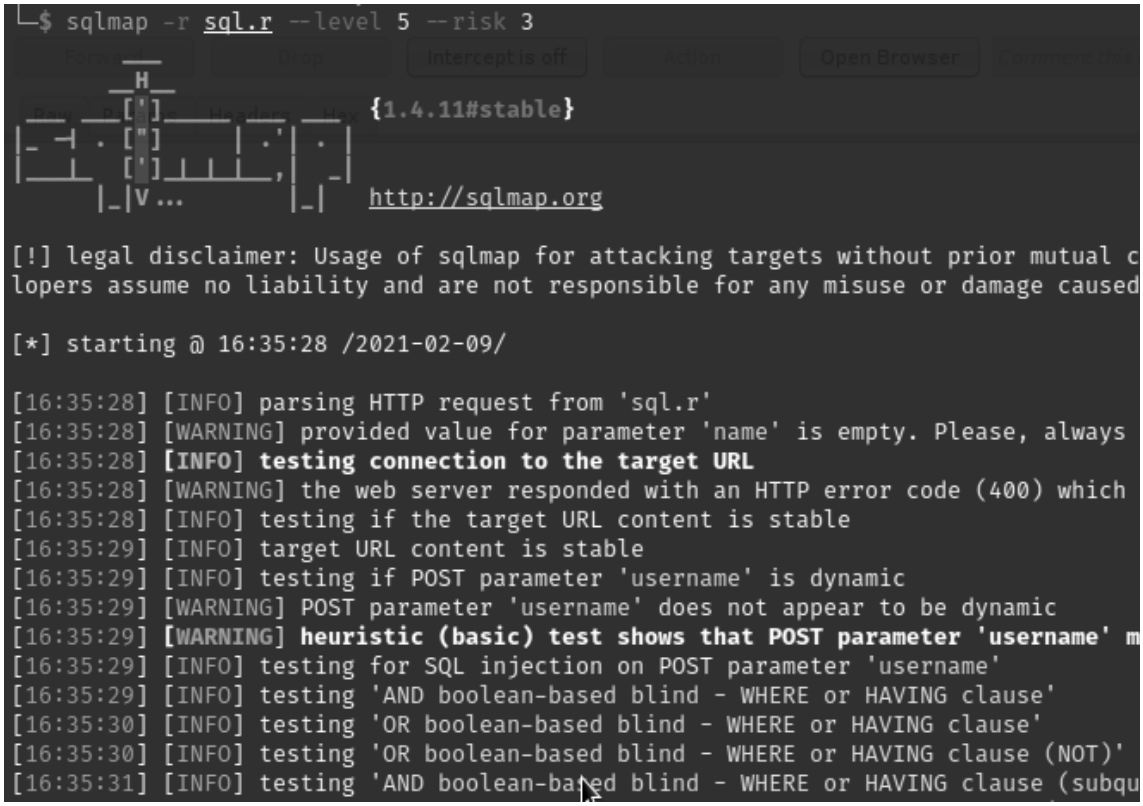

**Εικόνα 25. Gobuster**

Το πιο ενδιαφέρον από τα αποτελέσματα είναι το αρχείο ***login.php*** και γι' αυτό εξετάστηκε πρώτο. Όταν ο penetration tester επισκεφτεί τη σελίδα ***login.php***, όπως ήταν αναμενόμενο, του δίνεται η δυνατότητα σύνδεσης παρέχοντας όνομα χρήστη και κωδικό.



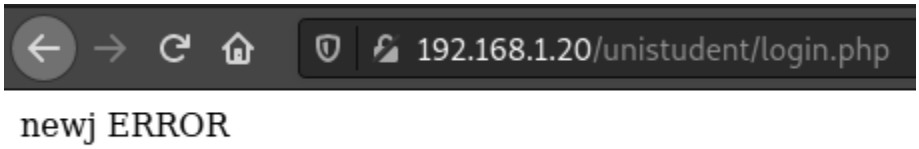

Όταν υπάρχει σελίδα, η οποία δίνει στον χρήστη τη δυνατότητα σύνδεσης, σημαίνει ότι υπάρχει κάποια βάση δεδομένων στο back-end, η οποία εξυπηρετεί την αυθεντικοποίηση. Πραγματοποιήθηκε έλεγχος για την ύπαρξη ή όχι αδυναμιών που επιτρέπουν είτε την αυθεντικοποίηση χωρίς την γνώση των σωστών αναγνωριστικών είτε την άντληση πληροφοριών από τη βάση.

Εν συνεχεία, εκτελέστηκε το Sqlmap - ένα εργαλείο, το οποίο αυτοματοποιεί τον έλεγχο για SQL Injection αδυναμίες. Ουσιαστικά, στέλνει διάφορα SQL queries στην εφαρμογή και διαβάζει το response. Σε περίπτωση λάθους απλώς αγνοεί το συγκεκριμένο query· διαφορετικά, το επιστρέφει ως επιτυχές με σκοπό να το αξιοποιήσει ο αναλυτής (penetration tester). Τα αποτελέσματα του φαίνονται παρακάτω ***(Εικόνα 26)***.



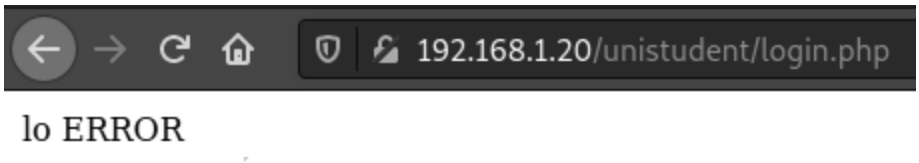

**Εικόνα 26. Sqlmap**

Το Sqlmap δεν κατάφερε να βρει κάποιο query, οπότε συνεχίστηκε ο έλεγχος εξετάζοντας τη συμπεριφορά της εφαρμογής. Παρατηρήθηκε ότι επιστρέφεται διαφορετικό μήνυμα λάθους, ανάλογα με τα δεδομένα εισόδου. Συγκεκριμένα όταν εισάγεται λανθασμένο username ή/ και password, το μήνυμα λάθους είναι το εξής ***(Εικόνα 27)***:




**Εικόνα 27. Μήνυμα λάθους μετά από εισαγωγή λανθασμένων στοιχείων εισόδου**

Ενώ όταν εισάγεται κάποιος χαρακτήρας, ο οποίος χρησιμοποιείται για ανίχνευση ευπαθειών, το μήνυμα που εμφανίζεται είναι διαφορετικό ***(Εικόνα 28)***.

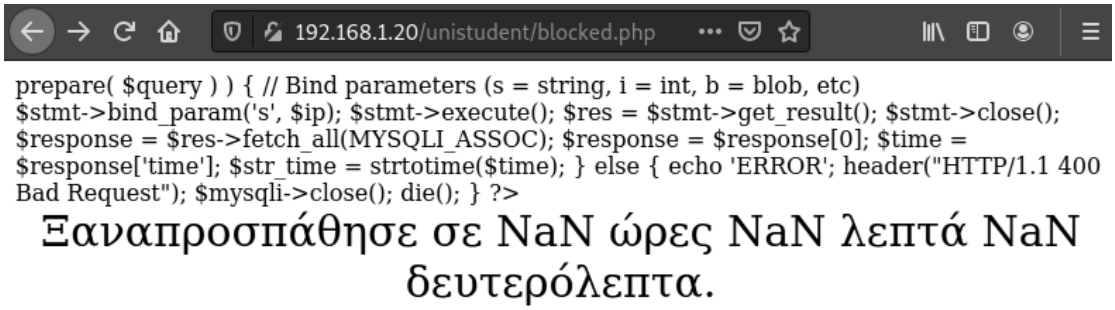



**Εικόνα 28. Μήνυμα λάθους μετά από εισαγωγή μη επιτρεπτών χαρακτήρων**

Αυτό αποτελεί μία ισχυρή ένδειξη ότι η εφαρμογή διαφοροποιεί τη συμπεριφορά της ανάλογα με την είσοδο και ενδεχομένως να διαθέτει κάποιον μηχανισμό ανίχνευσης αυτών των χαρακτήρων. Το παραπάνω αποτελεί σημαντική πληροφορία για έναν επιτιθέμενο, καθώς καθοδηγείται σε σχέση με τους ελέγχους, τους οποίους θα πρέπει να κάνει.

Στην ίδια σελίδα υλοποιήθηκαν έλεγχοι για XSS, αλλά αποδείχθηκε ότι δεν υπάρχει κάποια αδυναμία.

Όπως και στα αποτελέσματα του Gobuster αυτό που παρατηρήθηκε κι εδώ είναι **Information Disclosure**. Συγκεκριμένα εάν κάποιος επισκεφθεί τη σελίδα ***blocked.php***, θα δει το παρακάτω:

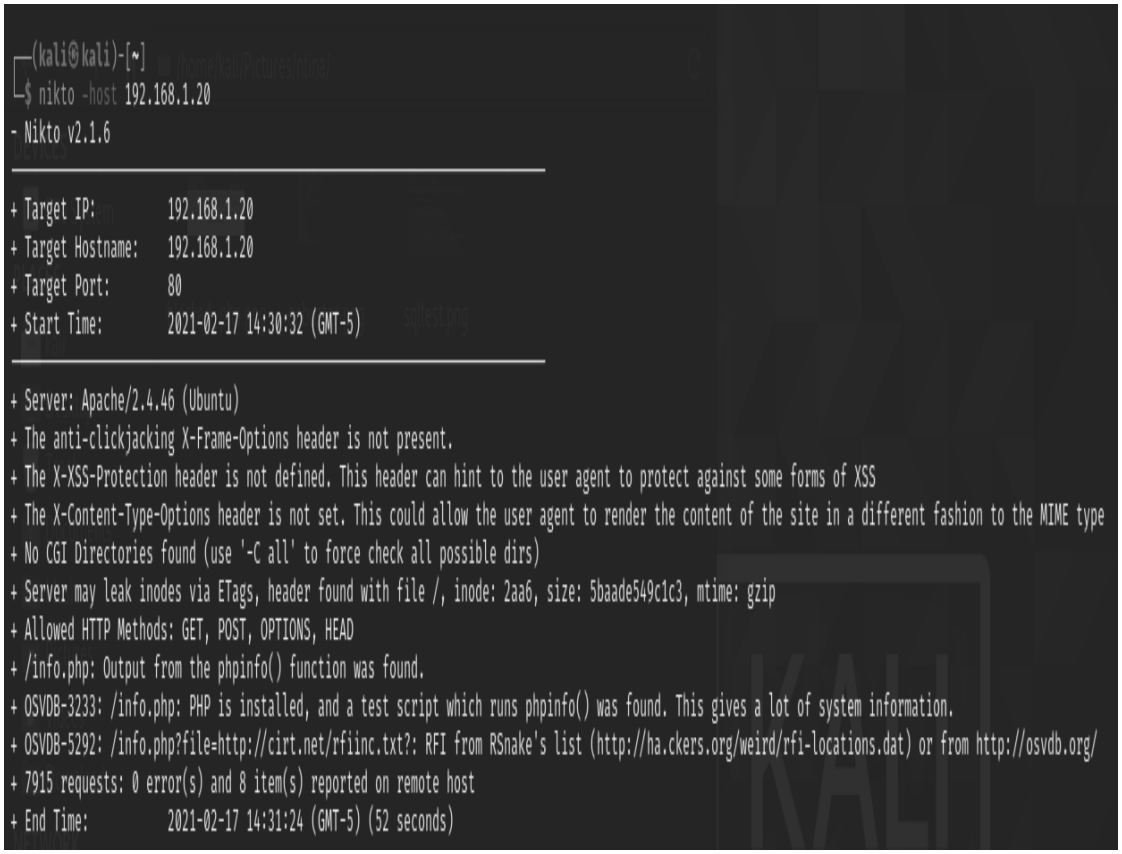



**Εικόνα 29. Σελίδα *blocked.php***

Από τον κώδικα και το μήνυμα, το οποίο εμφανίζεται ***(Εικόνα 29)*** συνάγεται ως συμπέρασμα ότι η εφαρμογή εμποδίζει συγκεκριμένες IP για συγκεκριμένο χρονικό διάστημα υπό κάποιες προϋποθέσεις. Το πιο πιθανό είναι να εμποδίζει επιθέσεις **Brute Force**.

Αναφορικά με τις υπόλοιπες σελίδες της εφαρμογής δεν βρέθηκαν αξιόλογα ευρήματα.

Στην τελική φάση του ελέγχου χρησιμοποιήθηκε το Nikto, το οποίο είναι ένα εργαλείο για αυτοματοποιημένους ελέγχους αδυναμιών. Τα αποτελέσματα του φαίνονται παρακάτω ***(Εικόνα 30)***.




**Εικόνα 30. Nikto**

Το μόνο που βρέθηκε είναι ότι υπάρχει το αρχείο ***info.php***.

**Συνοπτική παρουσίαση ευρημάτων**

Ο πίνακας 2 παρουσιάζει συνοπτικά τις αδυναμίες της εφαρμογής "***UniStudent"***, καθώς και παρατηρήσεις σε σχέση με το πως οι παραπάνω έχουν αποφευχθεί.

| ***Αδυναμίες εφαρμογής "UniStudent"*** | | |
|---|---|---|
| **OWASP Top 10 αδυναμίες διαδικτυακών εφαρμογών** | **Βρέθηκαν** | **Παρατηρήσεις** |
| Injection | × | Έλεγχος δεδομένων εισόδου |
| Broken Authentication | × | Αποφυγή χρήσης Cookies |
| Sensitive Data Exposure | × | Αποφυγή σχολίων, back-up αρχείων και δοκιμαστικών directories |
| XML External Entities (XXE) | - | *Η αδυναμία XML External Entities (XXE) απαιτεί την ύπαρξη xml κώδικα.* |
| Broken Access Control | × | Ύπαρξη μόνο 2 απλών χρηστών χωρίς αναβαθμισμένο ρόλο. |
| Security Misconfiguration | × | Χρήση σωστών ρυθμίσεων και πρακτικών |
| Cross Site Scripting (XSS) | × | Έλεγχος δεδομένων εισόδου |
| Insecure Deserialization | - | *Η αδυναμία Insecure Deserialization απαιτεί την ύπαρξη αντικειμενοστραφούς προγραμματισμού.* |
| Using Components with Known Vulnerabilities | - | *Η αδυναμία Using Components with Known Vulnerabilities απαιτεί την ύπαρξη έτοιμων βιβλιοθηκών και συναρτήσεων.* |
| Insufficient Logging and Monitoring | - | *Η αδυναμία Insufficient Logging and Monitoring απαιτεί την ύπαρξη αρχείων logs.* |
| **Επιπρόσθετες αδυναμίες** | | |
| Information Disclosure [8] | ✓ | Δεν έχει πραγματοποιηθεί κάποια προσπάθεια αποφυγής της συγκεκριμένης αδυναμίας. |

**Πίνακας 2. Πίνακας αδυναμιών εφαρμογής *"UniStudent"***

[8] **Η αδυναμία Information Disclosure αναφέρεται στην ακούσια αποκάλυψη ευαίσθητων πληροφοριών.**

**Προτάσεις**

Με βάση τα ευρήματα, παρατίθενται κάποιες προτάσεις με στόχο την ενίσχυση της ασφάλειας της εφαρμογής:

**1.** Χρήση κάποιας ασφαλούς μεθόδου κρυπτογράφησης στη βάση δεδομένων.

**2.** Έλεγχος στις σελίδες HTML και απομάκρυνση σχολίων, τα οποία δίνουν ευαίσθητες πληροφορίες.

# Επίλογος

Εν συντομία, η δημιουργία μίας διαδικτυακής εφαρμογής αποτελεί από μόνη της ένα σύνθετο έργο και απαιτεί τη συνεργασία πολλών ειδικοτήτων. Κατά τη σχεδίαση και την ανάπτυξη μίας εφαρμογής θα πρέπει να δίνεται έμφαση στην ασφάλεια της. Αναλυτές και προγραμματιστές θα πρέπει σε πρώτη φάση να περιορίσουν όσο είναι δυνατόν τα σφάλματα και τις ευπάθειες που ενδεχομένως να προκύψουν. Οι penetration testers θα πρέπει στη συνέχεια να ανακαλύψουν όσο το δυνατόν περισσότερες αδυναμίες και με βάση την καθοδήγηση τους, οι τελευταίες να διορθωθούν από το ειδικευμένο προσωπικό.

Υπάρχουν διάφορες τεχνικές και μέθοδοι ασφάλειας, τόσο σε επίπεδο εφαρμογής όσο και σε επίπεδο συστήματος και δικτύου. Με τον συνδυασμό τους επιτυγχάνεται το βέλτιστο αποτέλεσμα, καθώς η αποτυχία σε ένα επίπεδο δεν σημαίνει απαραίτητα αποτυχία του συνολικού συστήματος.

Σε αυτό το σημείο θα πρέπει να σημειωθεί ότι η ασφάλεια διαδικτυακών εφαρμογών κατά τη σχεδίαση και την ανάπτυξη τους θεωρείται πολύ σημαντική. Εξίσου σημαντική όμως είναι και η αξιολόγηση της ασφάλειας τους μέσω hacking (penetration testing). Με την εξέλιξη της τεχνολογίας, όπου οι ταχύτητες των υπολογιστών και οι ταχύτητες σύνδεσης στο Internet αυξάνονται, είναι πιθανό να αυξηθούν επιθέσεις, οι οποίες έως αυτήν τη στιγμή δεν χρησιμοποιούνταν συχνά εξαιτίας είτε των υψηλών απαιτήσεων τους σε χρόνο (π.χ. Brute Force, Password Attacks) είτε της αυξημένης δυσκολίας τους. Ακόμα, είναι πιθανό είτε να εμφανιστούν νέες επιθέσεις είτε και να σταματήσουν να χρησιμοποιούνται ήδη υπάρχουσες.

Οι επαγγελματίες διαφορετικών ειδικοτήτων, οι οποίοι συμμετέχουν στη σχεδίαση και στην ανάπτυξη μίας διαδικτυακής εφαρμογής, οφείλουν να έχουν γνώση τόσο πρακτική όσο και θεωρητική, να εξελίσσονται συνεχώς και τέλος, να είναι σε θέση να συνεργαστούν μεταξύ τους με στόχο να βρεθεί η “χρυσή τομή”, καθώς εκτός από την ασφάλεια θα πρέπει να εξεταστούν και ζητήματα, όπως η

απόδοση της εφαρμογής, ο όγκος των δεδομένων κ.α. Βασικός στόχος τους θα πρέπει να είναι η διασφάλιση της ύψιστης δυνατής ασφάλειας.

# Βιβλιογραφικές αναφορές

**Ελληνική Βιβλιογραφία**


Θωδωμάκης, Δ. (2018). Ασφάλεια Email ..οι απάτες, οι επιθέσεις & οι τρόποι προστασίας. [online] Available at: https://www.itsecuritypro.gr/asfaleia-email-oi-apates-oi-epitheseis-amp-oi-tropoi-prostasias/ (Accessed: 09 January 2021)

Λαζακίδου Α., Χατζημιτσής Δ. Γ., Ευαγγέλου Ι. Ε. (2004). *Εικονικός Κόσμος Και Νέες Τεχνολογίες.* Αθήνα: Εκδόσεις Κλειδάριθμος.

Μαυρίδης, Ι. (2015). *Ασφάλεια Πληροφοριών στο Διαδίκτυο.* [online] Available at: https://www.openbook.gr/asfaleia-pliroforiwn-sto-diadiktyo/ Accessed: 27 December 2020).

Πάγκαλος, Γ., Μαυρίδης Ι. (2002). *Ασφάλεια Πληροφοριακών Συστημάτων και Δικτύων.* Θεσσαλονίκη: Εκδόσεις Ανικούλα.

Πανσεληνάς Γ., Αγγελιδάκης Ν., Μιχαηλίδη Α., Μπλάτσιος Χ., Παπαδάκης Σ., Παυλίδης Γ., Τζαγκαράκης Ε., Τζωρμπατζάκης Α. (2015). *Εφαρμογές Πληροφορικής.* [online] Available at: http://ebooks.edu.gr/ebooks/v/pdf/8547/2594/22-0226-02_Efarmoges-Pliroforikis_A-Lykeiou-Epilogis_Vivlio-Mathiti/ (Accessed: 6 January 2021)

**Ξένη Βιβλιογραφία**


Cloudfare (2020). What is a WAF? | Web Application Firewall explained. *Cloudflare*. [online] Available at: https://www.cloudflare.com/learning/ddos/glossary/web-application-firewall-waf/ (Accessed: 1 November 2020)

Diagboya, E. (2019). How to Start using Cloud Computing as a StartUp. *MyCloudSeries.* [online] Available at: https://medium.com/mycloudseries/how-to-start-using-cloud-computing-as-a-startup-77055c60f74f (Accessed: 6 January 2021)

Dropbox (2020). How To Enable Two-Step Verification. *Dropbox* [online] Available at: https://help.dropbox.com/teams-admins/team-member/enable-two-step-verification (Accessed: 27 December 2020)

Edtech (2020). Dropbox: Εφαρμογή Αποθήκευσης Και Οργάνωσης Αρχείων. *Edtech.* [online] Available at: https://edtech.gr/dropbox/ (Accessed: 27 December 2020)

Evans, G. (2019). Two-tier Vs Three-tier Architecture. [online] Available at: https://medium.com/@gacheruevans0/2-tier-vs-3-tier-architecture-26db56fe7e9c (Accessed: 6 January 2021)

Facebook (2020). Accessing & Downloading Your Information. *Facebook Help Centre*. [online] Available at: https://www.facebook.com/help/330229433729799/ (Accessed: 27 December 2020)

Google (2020). Αποκτήστε Την Ισχυρότερη Ασφάλεια Λογαριασμού Της Google Με Το Πρόγραμμα Σύνθετης Προστασίας. *Google*. [online] Available at: https://support.google.com/accounts/answer/7519408?co=GENIE.Platform%3DAndroid&hl=el (Accessed: 27 December 2020)

Juniper networks (2020). What is IDS and IPS?. *Juniper Networks.* [online] Available at: https://www.juniper.net/uk/en/products-services/what-is/ids-ips/ (Accessed: 1 November 2020)

Microsoft (2020). Πώς Το OneDrive Προστατεύει Τα Δεδομένα Σας Στο Cloud. *Microsoft.* [online] Available at: https://support.microsoft.com/el-gr/office/πώς-το-onedrive-προστατεύει-τα-δεδομένα-σας-στο-cloud-23c6ea94-3608-48d7-8bf0-80e142edd1e1 (Accessed: 27 December 2020)

OWASP (2020). OWASP Top Ten Web Application Security Risks. *OWASP.* [online] Available at: https://owasp.org/www-project-top-ten/ (Accessed: 1 November 2020)

Pandith M. Y. (2014). Data Security and Privacy Concerns in Cloud Computing. *Internet of Things and Cloud Computing.* 2(2), pp. 6-11. [online] Available at: http://www.sciencepublishinggroup.com/journal/paperinfo.aspx?journalid=238&doi=10.11648/j.iotcc.20140202.11 (Accessed: 06 January 2021)

PCMag Greece (2020). Google: Ελέγξτε Την Ασφάλειά Σας Στο Διαδίκτυο. [online] Available at: https://gr.pcmag.com/asphaleia/29721/google-elegxte-ten-asphaleia-sas-sto-diadiktuo (Accessed: 27 December 2020)

Pcsteps (2020). 23 Δυνατότητες Του Dropbox Που Ίσως Δεν Γνωρίζατε. [online] Available at: https://www.pcsteps.gr/ 143016-οι-καλύτερες-δυνατότητες-του-dropbox (Accessed: 27 December 2020)

Petoskey, C. (2020). These are the Skills you will Need for a Superb Front End Development. [online] Available at: https://www.techgyd.com/skills-you-need-for-superb-front-end-development/43323/ (Accessed: 6 January 2021)

PortSwigger (2020). Burp Suite - Application Security Testing Software. [online] Available at: https://portswigger.net/burp (Accessed: 1 November 2020)

Raj, P., Raman A., Subramanian H. (2017). Architectural Patterns. [online] Available at: https://www.oreilly.com/library/view/architectural-patterns/9781787287495/df0e98f1-0190-42e2-a9b1-69f050a03a4e.xhtml (Accessed: 6 January 2021)

Wikipedia (2020). Google Drive. [online] Available at: https://el.wikipedia.org/wiki/Google_Drive (Accessed: 27 December 2020)

Wikipedia (2020). Lightweight Directory Access Protocol. [online] Available at: https://en.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol (Accessed: 1 November 2020)

The Greek Honeynet Project (2016). Εισαγωγή Honeypots – Honeynets. [online] Available at: https://docplayer.gr/12036445-Kefalaio-1-eisagogi-honeypots-honeynets.html (Accessed: 09 January 2021)