ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΣΧΕΔΙΑΣΜΟΣ ΙΣΤΟΣΕΛΙΔΑΣ ΜΕ ΕΜΦΑΣΗ ΣΤΗΝ ΑΝΤΙΜΕΤΩΠΙΣΗ ΤΥΠΙΚΩΝ ΕΥΠΑΘΕΙΩΝ ΚΑΙ ΣΤΗΝ ΙΔΙΩΤΙΚΟΤΗΤΑ

ΜΠΑΤΣΙΚΑΣ ΣΤΕΦΑΝΟΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

ΔΑΔΑΛΙΑΡΗΣ ΑΝΤΩΝΙΟΣ
ΕΠΙΚΟΥΡΟΣ ΚΑΘΗΓΗΤΗΣ

Λαμία έτος 2024

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΣΧΕΔΙΑΣΜΟΣ ΙΣΤΟΣΕΛΙΔΑΣ ΜΕ ΕΜΦΑΣΗ ΣΤΗΝ ΑΝΤΙΜΕΤΩΠΙΣΗ ΤΥΠΙΚΩΝ ΕΥΠΑΘΕΙΩΝ ΚΑΙ ΣΤΗΝ ΙΔΙΩΤΙΚΟΤΗΤΑ

ΜΠΑΤΣΙΚΑΣ ΣΤΕΦΑΝΟΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

ΔΑΔΑΛΙΑΡΗΣ ΑΝΤΩΝΙΟΣ
ΕΠΙΚΟΥΡΟΣ ΚΑΘΗΓΗΤΗΣ

Λαμία έτος 2024

UNIVERSITY OF
THESSALY

SCHOOL OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE & TELECOMMUNICATIONS

Website design with attention to dealing with typical vulnerabilities and privacy

BATSIKAS STEFANOS

FINAL THESIS

ADVISOR

DADALIARIS ANTONIOS
ASSISTANT PROFESSOR

Lamia year 2024

«Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις [(1)], που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

*1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί **χωρίς να τα περικλείω σε εισαγωγικά** και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάσθηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.*

*2. Δέχομαι ότι η αυτολεξεί **παράθεση χωρίς εισαγωγικά**, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.*

3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια

4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

Ημερομηνία: 31/10/2024

Ο Δηλών.

(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση
του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων
σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.»

ΠΕΡΙΛΗΨΗ

Η ραγδαία και συνεχή εξέλιξη της επιστήμης της πληροφορικής και του διαδικτύου έχει ως αποτέλεσμα τον σχεδιασμό και την ανάπτυξη πληθώρας διαδικτυακών εφαρμογών και ιστοσελίδων για την κάλυψη και διευκόλυνση πολλών αναγκών. Είναι απολύτως αυτονόητο πως η ασφάλεια πρέπει να αποτελεί μια από τις βασικότερες προτεραιότητες για τους προγραμματιστές.

Σκοπός αυτής της εργασίας είναι να παρουσιαστεί ο επιμελής σχεδιασμός μιας ιστοσελίδας της οποίας το κύριο μέλημα είναι η ασφάλεια έναντι γνωστών ευπαθειών καθώς και η προστασία της ιδιωτικότητας των χρηστών.

Στα αρχικά κεφάλαια θα γίνει αναφορά σε βασικές έννοιες, ορολογίες καθώς και σε λειτουργικά ζητήματα. Έπειτα θα αναλυθεί ο σχεδιασμός και η υλοποίηση μιας ιστοσελίδας με τη χρήση Django. Τέλος, θα γίνει μελέτη ευπαθειών της συγκεκριμένης ιστοσελίδας και θα παρουσιαστούν τρόποι αντιμετώπισης.

ABSTRACT

The rapid and continuous development of computer science and the internet has resulted in the design and development of many network applications and websites to meet and facilitate many needs. It goes without saying that security should be one of the top priorities for developers.

The purpose of this thesis is to present the careful design of a website which main concern is security against known vulnerabilities as well as the protection of user privacy.

In the initial chapters, reference will be made to basic concepts, terminologies as well as operational issues. Then the design and the implementation of a website using Django will be analyzed. Finally, there will be a vulnerability analyze of the specific website and ways to deal with them will be presented.

Table of Contents

ΛΙΣΤΑ ΣΥΝΤΟΜΟΓΡΑΦΙΩΝ

ΑΠ – Ασφάλεια Πληροφοριών

Η/Υ – Ηλεκτρονικός Υπολογιστής

ARPANET – Advanced Research Projects Agency Network

ASGI – Asynchronous Server Gateway Interface

CIA – Confidentiality Integrity Availability

CSS – Cascading Style Sheets

CVE – Common Vulnerabilities and Exposures

DNS – Domain Name System

GDPR – General Data Protection Regulation

HTML – Hyper Text Markup Language

HTTP – HyperText Transfer Protocol

HTTPS – HyperText Transfer Protocol Secure

IP – Internet Protocol

MVT – Model View Template

NIST – National Institute of Standards and Technology

ORM – Object-Relational Mapper

OSI – Open Systems Interconnection

PBKDF2 – Password-Based Key Derivation Function 2

SSL – Secure Sockets Layer

TCP – Transmission Control Protocol

TLS – Transport Layer Security

URL – Uniform Resource Locator

WSGI – Web Server Gateway Interface

XSS – Cross Site Scripting

ΚΕΦΑΛΑΙΟ 1 - ΠΑΓΚΟΣΜΙΟΣ ΙΣΤΟΣ

Ενότητα 1.1 - Ορισμός

Το Διαδίκτυο υλοποιήθηκε από κυβερνητικούς ερευνητές οι οποίοι είχαν ως στόχο την μεταφορά πληροφοριών μεταξύ δύο ή παραπάνω ηλεκτρονικών υπολογιστών (Η/Υ) κατά την δεκαετία του 1960. Εκείνα τα χρόνια οι Η/Υ ήταν εξαιρετικά ογκώδης και για τη μεταφορά δεδομένων μεταξύ τους έπρεπε οι σκληροί δίσκοι της εποχής , δηλαδή οι μαγνητικές κασέτες υπολογιστή, να μεταφερθούν με το ταχυδρομείο. Αυτή η μέθοδος ήταν χρονοβόρα, δαπανηρή και αρκετά παρακινδυνευμένη.

Η αρχική μορφή του Διαδικτύου ήταν το ARPANET(Advanced Research Projects Agency Network) το 1969 ήταν το πρώτο δίκτυο μεταγωγής πακέτου (packet switching) δεδομένων μεταξύ πολλαπλών Η/Υ οι οποίοι βρίσκονται συνδεδεμένοι στο ίδιο δίκτυο. Το ARPANET χρησιμοποιήθηκε μόνο από το Υπουργείο Άμυνας των Ηνωμένων Πολιτειών , ορισμένα εργαστήρια ερευνών και πανεπιστημίων στα οποία παρέχονταν ειδική άδεια πρόσβασης.

Το Διαδίκτυο παραδόθηκε στο ευρύ κοινό για χρήση την πρώτη Ιανουαρίου του 1983 και αυτό το γεγονός κατέστη δυνατόν λόγω της ανάπτυξης του πρωτοκόλλου επικοινωνίας Transmission Control Protocol/Internet Protocol (TCP/IP) το οποίο επέτρεψε κάθε είδος Η/Υ σε οποιοδήποτε δίκτυο να επικοινωνεί με τους υπολοίπους. Αυτό το τεχνολογικό επίτευγμα άλλαξε ριζικά την ανθρωπότητα και την διαμόρφωσε στη σημερινή εποχή της.

Εκτιμάται ότι το 2000 οι χρήστες του Παγκοσμίου Ιστού ανερχόταν στους 361 εκατομμύρια χρήστες ενώ σύμφωνα με την Statista, τον Απρίλιο του 2024 οι χρήστες έχουν φτάσει τους 5,44 δισεκατομμύρια.

Ενότητα 1.2 - Πρωτόκολλα Διαδικτύου

Κάθε μορφή διαδικτυακής επικοινωνίας επιβάλλεται να ακολουθεί ορισμένους κανόνες που ορίζονται από τα πρωτόκολλα διαδικτύου (IP). Αυτοί οι κανόνες ακολουθούνται πιστά από κάθε ηλεκτρονική συσκευή για την βέλτιστη επικοινωνία με το διαδίκτυο. Τα πιο γνωστά πρωτόκολλα διαδικτύου είναι το Μοντέλο Αναφοράς Ανοιχτής Διασύνδεσης Συστημάτων (OSI reference model) και το TCP/IP.

Υποενότητα 1.2.1 – OSI Model

Το μοντέλο OSI αποτελείται από επτά διαφορετικά επίπεδα το καθένα με τις λειτουργίες του. Παρακάτω αναφέρεται επιγραμματικά οι λειτουργία κάθε επιπέδου από το υψηλότερο στο χαμηλότερο.

- **Επίπεδο Εφαρμογής**

  Το επίπεδο στο οποίο ο χρήστης αλληλοεπιδράει με εφαρμογές οι οποίες του δίνουν την δυνατότητα να έχει πρόσβαση στο δίκτυο, να μεταφέρει και να βλέπει τα δεδομένα που τον ενδιαφέρουν.

- **Επίπεδο Παρουσίασης**

  Το επίπεδο το οποίο μεταφράζει τα δεδομένα που λαμβάνει σε κατάλληλη μορφή , κρυπτογραφεί και συμπιέζει δεδομένα του επιπέδου εφαρμογής. Αντιθέτως αποκρυπτογραφεί και αποσυμπιέζει δεδομένα του επιπέδου συνεδρίας.

- **Επίπεδο Συνεδρίας**

  Το επίπεδο το οποίο είναι υπεύθυνο για την δημιουργία, διατήρησης και τερματισμού συνεδρίας.

- **Επίπεδο Μεταφοράς**

Το επίπεδο το οποίο είναι υπεύθυνο για τη μεταφορά των δεδομένων. Ελέγχει την ολοκληρωμένη με σωστή σειρά, χωρίς σφάλματα και διπλοτυπίες μεταφορά δεδομένων.

- **Επίπεδο Δικτύου**

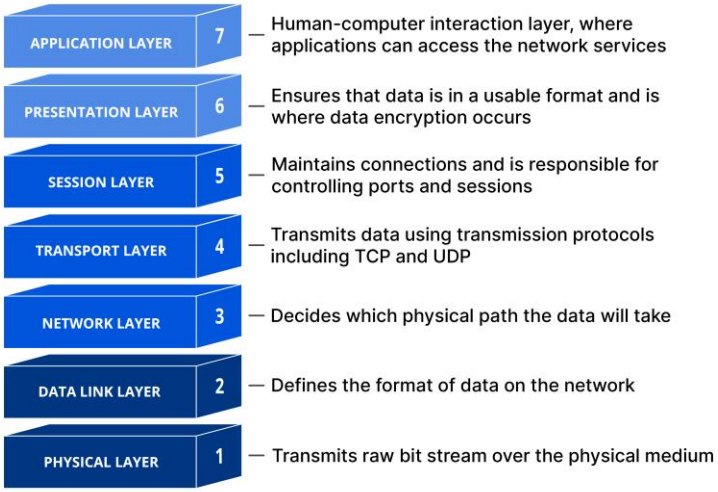
  Το επίπεδο το οποίο είναι υπεύθυνο για τη διαβίβαση δεδομένων μεταξύ χρηστών οι οποίοι βρίσκονται σε διαφορετικά δίκτυα. Επιλέγει τη συντομότερη διαδρομή για την μεταφορά των δεδομένων, ελέγχει την κίνηση και αναλαμβάνει την διευθυνσιοδότηση των δεδομένων.

- **Επίπεδο Ζεύξης Δεδομένων**

  Το επίπεδο το οποίο παρέχει αξιόπιστη και αποδοτική επικοινωνία μεταξύ συσκευών, αναθέτει σε κάθε συσκευή του τοπικού δικτύου μία μοναδική ταυτότητα και καθορίζει ποια θα είναι η μορφή των δεδομένων στο δίκτυο.

- **Φυσικό Επίπεδο**

  Το φυσικό επίπεδο είναι υπεύθυνο για τη μεταφορά της πληροφορίας σε μορφή bits μέσω ενός φυσικού μέσου, όπως τα ομοαξονικά καλώδια και οι οπτικές ίνες, ανάμεσα σε δύο κόμβους. Αποτελεί το χαμηλότερο επίπεδο.



Εικόνα 1.1 Διάγραμμα μοντέλου OSI

(https://www.cloudflare.com/learning/ddos/glossary/open-systems-interconnection-model-osi/)

Υποενότητα 1.2.2 – TCP/IP Model

Το μοντέλο TCP/IP είναι μία πιο συνοπτική έκδοση του OSI. Αποτελείται από τέσσερα συνολικά επίπεδα.

- **Επίπεδο Εφαρμογής**

  Αποτελείται από τα επίπεδα Εφαρμογής, Παρουσίασης και Συνεδρίας του OSI.

- **Επίπεδο Μεταφοράς**

- **Επίπεδο Διαδικτύου**

- **Φυσικό Επίπεδο**

  Αποτελείται από το Επίπεδο Ζεύξης Δεδομένων και το Φυσικό Επίπεδο του OSI.

Συνήθως το TCP/IP πρωτόκολλο χρησιμοποιείται περισσότερο λόγω των λιγότερων επιπέδων, της πρακτικότητας του σε πραγματικών συνθήκων σενάρια δικτύωσης και της ευκολότερης εκμάθησης και κατανόησης από τους αρχάριους στο χώρο των τηλεπικοινωνιών.

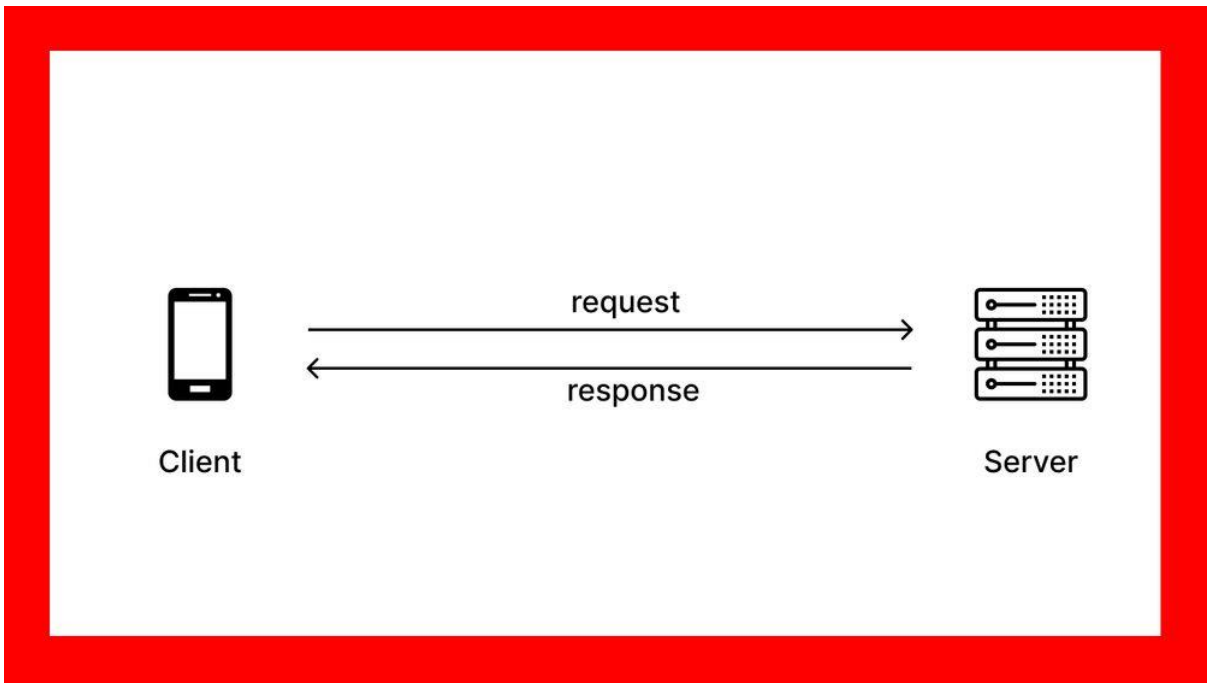
Ενότητα 1.3 - Πρωτόκολλα Επικοινωνιών

Το Πρωτόκολλο Επικοινωνίας ορίζει, τον τρόπο, το συντακτικό, τον συγχρονισμό και μεθόδους πρόληψης σφαλμάτων, δηλαδή ένα σύστημα κανόνων με απώτερο σκοπό την αλληλομεταφορά δεδομένων-πληροφοριών μεταξύ χρηστών.
Δύο από τα κυριότερα πρωτόκολλα επικοινωνιών που χρησιμοποιούνται στον παγκόσμιο ιστό για την επικοινωνία διακομιστή – πελάτη είναι το HTTP και το HTTPS.

Υποενότητα 1.3.1 – HyperText Transfer Protocol – HTTP

Το Πρωτόκολλο Μεταφοράς Υπερκειμένου (HTTP) είναι το θεμέλιο της επικοινωνίας δεδομένων για τον Παγκόσμιο Ιστό. Με την χρήση του HTTP είναι δυνατή η ανταλλαγή δεδομένων μεταξύ πελάτη και διακομιστή. Οποιοσδήποτε web browser έμμεσα χρησιμοποιεί HTTP για την επικοινωνία του. Το πρωτόκολλο HTTP είναι επιπέδου εφαρμογής.
Για να συνδεθεί ο πελάτης σε μία ιστοσελίδα μέσω του web browser το HTTP πρωτοκόλλου θα στείλει ένα αίτημα (request) στο διακομιστή , ο οποίος θα ανταποκριθεί (response) στέλνοντας πίσω το κατάλληλο αρχείο ώστε να ανοίξει η ιστοσελίδα.

Το πρωτόκολλο HTTP βασίζεται στο μοντέλο Αιτήματος – Ανταπόκρισης (Request – Response).



Εικόνα 1.2 Μοντέλο Αιτήματος – Ανταπόκρισης

(https://www.freecodecamp.org/news/what-is-http/)

Κάθε αίτημα που στέλνει ο πελάτης οφείλει να περιέχει την κατάλληλη δομή και τις απαραίτητες πληροφορίες που θα χρειαστεί ο διακομιστής για να κατανοήσει το αίτημα και η απάντηση του να είναι η επιθυμητή.

Αρχικά το HTTP αίτημα θα αναγράφει την ακριβή έκδοση του HTTP, τον Ενιαίο Εντοπιστή Πόρων (Uniform Resource Locator / URL), είναι η διεύθυνση ενός συγκεκριμένου και μοναδικού πόρου στο διαδίκτυο, έπειτα αναγράφεται η HTTP μέθοδο η οποία ορίζει ποια ενέργεια αναμένει ως απάντηση από τον διακομιστή, ακολουθεί η κεφαλίδα (header) του αιτήματος που περιέχει χρήσιμες πληροφορίες για τον διακομιστή και τελειώνει με το ‘σώμα’ (body) του μηνύματος με προαιρετικές πληροφορίες.

Οι πιο συχνοί HTTP μέθοδοι είναι:

- **GET**: Ζητείται από τον διακομιστή συγκεκριμένος πόρος.
- **POST**: Ζητείται από τον διακομιστή να δημιουργήσει νέο πόρο.
- **PUT**: Ζητείται από τον διακομιστή να επεξεργαστεί/ενημερώσει έναν ήδη προ υπάρχον πόρο.
- **DELETE**: Ζητείται από τον διακομιστή να διαγράψει ένα πόρο.

Οι υπόλοιποι μέθοδοι είναι PATCH, HEAD, OPTIONS, TRACE, CONNECT.

Εικόνα 1.3 Παράδειγμα HTTP αιτήματος.

(https://medium.com/@adilrk/http-request-and-response-e7da8eb3a00c)

Ως απάντηση σε ένα αίτημα ο διακομιστής επιστρέφει στον πελάτη ορισμένους κώδικες καταστάσεις (status codes) για να τον ενημερώσει σχετικά με το αίτημα του. Ο κωδικός αποτελείται από τρία ψηφία όπου το αρχικό ψηφίο (δεξιότερο) ορίζει την κατηγορία του κωδικού.

Αν ξεκινάει με 1XX τότε σημαίνει ότι ο διακομιστής συμφωνεί να χειριστεί το αίτημα. Στην περίπτωση που επιστρέψει 2XX σημαίνει ότι το αίτημα αποδέχθηκε και επιστρέφει το περιεχόμενο. Στην περίπτωση του 3XX υποδεικνύει στον πελάτη να γυρέψει τον πόρο σε διαφορετικό σημείο διότι έγινε μεταφορά του σε διαφορετικό σημείο. Η περίπτωση του 4XX είναι ο κωδικός ότι η αίτηση απορρίφθηκε λόγω σφάλματος του πελάτη ή η σελίδα δεν υπάρχει, το γνωστό σφάλμα 404. Τελική κατηγορία είναι κωδικοί 5XX στην οποία ο διακομιστής ενημερώνει τον πελάτη ότι αντιμετώπισε μια κατάσταση που του ήταν αδύνατη να χειριστεί.

Υποενότητα 1.3.2 – HyperText Transfer Protocol Secure – HTTPS

Το πρωτόκολλο HTTP διαδραμάτισε σημαντικότατο ρόλο στην εξέλιξη του διαδικτύου και της μεταφοράς των δεδομένων. Δυστυχώς παρατηρήθηκε μία αρκετά σοβαρή αδυναμία που ήταν ικανή να προκαλέσει τεράστια προβλήματα στους χρήστες. Τα HTTP μηνύματα ήταν απλά κείμενα που σημαίνει ότι σε περίπτωση που κάποιος κατάφερνε να αιχμαλωτίσει ένα μήνυμα μπορούσε με ευκολία να διαβάσει το περιεχόμενο του. Όπως αναφέρθηκε το πρωτόκολλο χρησιμοποιείται για μεταφορά δεδομένων και σε αρκετές περιπτώσεις μεταφέρονται ευαίσθητα προσωπικά δεδομένα όπως ονόματα χρήστη, διαφόρων ειδών κωδικοί και τραπεζικά στοιχεία. Αυτή την αδυναμία κατάφερε να αντιμετωπίσει το

πρωτόκολλο HTTPS. Όπως το υποδηλώνει και το τελευταίο γράμμα είναι η ασφαλή (Secure) έκδοση του HTTP. Το πρωτόκολλο HTTPS είναι επιπέδου μεταφοράς.

Για την ασφαλή μετάδοση δεδομένων στο διαδίκτυο αναπτύχθηκαν διάφορα πρωτόκολλα κρυπτογράφησης, τα οποία μετέτρεπαν ένα ευανάγνωστο κείμενο σε μία νέα μορφή κειμένου που ήταν ακατανόητο στον άνθρωπο. Δύο τέτοια πρωτόκολλα είναι το Secure Socket Layer (SSL) και το Transport Layer Security (TLS) και δραστηριοποιούνται σε κατώτερα επίπεδα από το επίπεδο εφαρμογής. Το TLS είναι η εξέλιξη του SSL και δημιουργήθηκε για να προσφέρει ιδιωτικότητα και ακεραιότητα δεδομένων στις διαδικτυακές επικοινωνίες. Το HTTPS μπορεί να χρησιμοποιήσει εξίσου και τα δύο προαναφερόμενα πρωτόκολλα ωστόσο συνηθίζεται περισσότερο η χρήση του πιο προηγμένου TLS.

Η διαφορά στη λειτουργία των πρωτοκόλλων HTTP και HTTPS, μην λαμβάνοντας υπόψιν την κρυπτογράφηση, έγκειται στο γεγονός ότι η δεύτερη πιστοποιεί την ταυτότητα του συνομιλητή και δημιουργεί μεταξύ του έναν κρυπτογραφημένο ασφαλή δίαυλο επικοινωνίας με απώτερο σκοπό να είναι δύσκολη η κλοπή δεδομένων. Η ενέργεια της ταυτοποίησης προσθέτει χρόνο με αποτέλεσμα το HTTPS να είναι πιο αργό από το HTTP.

ΚΕΦΑΛΑΙΟ 2 – ΑΣΦΑΛΕΙΑ ΚΑΙ ΙΔΙΩΤΙΚΟΤΗΤΑ

Ενότητα 2.1 - Ιστότοπος και ιστοσελίδα

Το έγγραφο (document) το οποίο είναι γραμμένο στη γλώσσα προγραμματισμού Hyper Text Markup Language (HTML) και προβάλλεται από το φυλλομετρητή (browser) ονομάζεται Ιστοσελίδα (webpage). Ένα website έχει τη δυνατότητα να εμπεριέχει διάφορους πόρους όπως προγράμματα που εκτελούν μία σειρά εντολών τα επονομαζόμενα ως scripts και επίσης υπάρχει και η περίπτωση να προβάλλονται εικόνες και να αναπαράγονται ήχοι και βίντεο.
Η συλλογή μιας ομάδας από ιστοσελίδες που είναι σε άμεση σύνδεση μεταξύ τους κατονομάζεται ως Ιστότοπος (website). Οποιοδήποτε website στο χώρο του διαδικτύου έχει και το προσωπικό του "όνομα" που στη γλώσσα της πληροφορικής λέγεται Domain Name. Η πρόσβαση σε ένα website μπορεί να επιτευχθεί με την είσοδο του domain name στην μπάρα διεύθυνσης του browser.
Πιο συγκεκριμένα στη μπάρα διεύθυνσης του browser εισάγεται ο Ενιαίος Εντοπιστής Πόρων (URL) ο οποίος έχει καθορισμένη ανατομία.
Για παράδειγμα για πρόσβαση στην ιστοσελίδα της Google ο url που θα πρέπει να εισαχθεί στη μπάρα διεύθυνσης είναι ο εξής:
https://www.google.com
Αρχικά ορίζεται το πρωτόκολλο επικοινωνίας https (υποενότητα 1.3.2), ακολουθεί ο subdomain www, έπειτα το domain name και τέλος ο top-level domain com που χρησιμοποιείται από εμπορικούς οργανισμούς. Η παραπάνω είναι η απλή μορφή ενός URL καθώς μπορούν να τοποθετηθούν και επιπλέον παράμετροι για ποιο εξεζητημένες αναζητήσεις.
Την στιγμή που ο URL δοθεί στο browser ο ίδιος θα αποστείλει αίτημα στο εξυπηρετητή (web server) ο οποίος περιέχει τον επιθυμητό ιστότοπο αυτός με τη σειρά του θα αποστείλει πίσω τα περιεχόμενα του ιστότοπου.

Ενότητα 2.2 - Τύποι Ιστοσελίδων

Υπάρχουν δύο είδη ιστοσελίδων οι στατικές ιστοσελίδες (static) και οι δυναμικές ιστοσελίδες (dynamic). Ο διαχωρισμός τους αναγνωρίζεται στον τρόπο που ο server θα επιστρέψει τις πληροφορίες στο browser.

Υποενότητα 2.2.1 – Στατική Ιστοσελίδα

Μια Στατική Ιστοσελίδα αποτελείται από μία ομάδα αρχείων που περιέχουν τον πηγαίο κώδικα κυρίως σε γλώσσες προγραμματισμού HTML ,CSS και JavaScript και είναι αποθηκευμένα σε έναν web server. Η στατική ιστοσελίδα έχει σταθερό αριθμό σελίδων και συγκεκριμένη δομή και περιεχόμενο. Αυτό σημαίνει ότι ο server στέλνει τα HTML αρχεία στους πελάτες ακριβώς όπως διαμορφώθηκαν από τους προγραμματιστές της ιστοσελίδας. Κάθε πελάτης λαμβάνει το ίδιο περιεχόμενο.
Επιπλέον δεν υπάρχει αλληλεπίδραση με καμία βάση δεδομένων και καμία παραποίηση από τον server. Ο μοναδικός τρόπος για να επεξεργαστεί μια στατική ιστοσελίδα είναι με τον προγραμματισμό του πηγαίου κώδικα της.

Πλεονεκτήματα

- Το κόστος δημιουργίας και συντήρησης είναι μικρό εξαιτίας της απουσίας βάσης δεδομένων και της απλότητας της ιστοσελίδας.

- Ο χρόνος φόρτωσης της ιστοσελίδας είναι μικρός καθώς το περιεχόμενο αποστέλλεται στον χρήστη χωρίς αλλοίωση.

- Η ασφάλεια των στατικών ιστοσελίδων είναι εξαιρετική εξαιτίας της απλότητας της κατασκευής ελαχιστοποιώντας τις ευπάθειες της.

Μειονεκτήματα

- Απουσία ειδοποιήσεων πραγματικού χρόνου.

- Δυσκολία στην εισαγωγή περιεχομένου διότι πρέπει να γραφτεί στον πηγαίο κώδικα της ιστοσελίδας.

Υποενότητα 2.2.2 – Δυναμική Ιστοσελίδα

Μια Δυναμική Ιστοσελίδα διαθέτει ένα πρόγραμμα διαχείρισης περιεχομένου. Η δυναμική ιστοσελίδα έχει την ικανότητα να παράγει περιεχόμενο σε πραγματικό χρόνο. Η δυναμική ιστοσελίδα είναι συνδεδεμένη με τουλάχιστον μία βάση δεδομένων από την οποία αντλεί δεδομένα σε πραγματικό χρόνο και παρουσιάζει στον κάθε πελάτη διαφορετικό περιεχόμενο. Ο πελάτης μπορεί να αλληλοεπιδράει με τον διακομιστή της ιστοσελίδας. Συνήθως οι δυναμικές ιστοσελίδες χρησιμοποιούν κάποιο CMS (Content Management System), όπως το WordPress.

Πλεονεκτήματα

- Η διαχείριση περιεχομένου είναι εύκολη υπόθεση σε μία δυναμική ιστοσελίδα.
- Οι δυναμικές ιστοσελίδες προσφέρουν εύκολη επεκτασιμότητα.

Μειονεκτήματα

- Είναι αργές εξαιτίας της παραγωγής περιεχομένου για τον κάθε χρήστη.
- Λόγω της χρήστης πολλών τεχνολογιών και της πολυπλοκότητας της ιστοσελίδας εγείρονται πολλές ανησυχίες ως προς την ασφάλεια της.
- Το κόστος κατασκευής και συντήρησης είναι τεράστιο.

Ενότητα 2.3 - Κυβερνοέγκλημα

Τις τελευταίες δεκαετίες η ραγδαία και όπως φαίνεται ασταμάτητη τεχνολογική πρόοδος έχει προκαλέσει μια νέα επανάσταση στην ανθρωπότητα. Η σύγχρονη κοινωνία έχει αλλάξει ριζικά τον τρόπο λειτουργίας της και βασίζεται όλο και περισσότερο στις ηλεκτρονικές συσκευές. Η αυξημένη χρήση των τεχνολογικών συσκευών έχει τροποποιήσει την συμπεριφορά των ανθρώπων , έχει αυξήσει την αλληλεπίδραση καθώς και έχει αυξήσει την πρόσβαση στην ενημέρωση και την μάθηση. Είναι σχεδόν αυτονόητο ότι η τεχνολογία είναι αναπόσπαστο κομμάτι την καθημερινότητας της πλειοψηφίας των ανθρώπων. Η νέα τεχνολογική περίοδος είναι απόλυτα συνυφασμένη με το διαδίκτυο. Η πρόσβαση στο διαδίκτυο αποτελεί προτεραιότητα για τον κάθε άνθρωπο και η πλειοψηφία των ηλεκτρονικών συσκευών κατασκευάζεται ώστε να έχει τη δυνατότητα σύνδεσης στο διαδίκτυο. Ο χώρος του επιχειρείν σε παγκόσμιο επίπεδο αναγκάζεται να αναβαθμιστεί και ανακαλύπτει νέους τρόπους λειτουργείας , διαφήμισης , αλληλεπίδρασης με τους πελάτες ή πιθανούς πελάτες και κάλυψη πρωτοεμφανιζόμενων αναγκών. Οι κυβερνήσεις βελτιώνουν τις υπηρεσίες τους προς τους πολίτες. Ωστόσο κάθε εξέλιξη ελλοχεύει και κινδύνους.

Οι νέες τεχνολογίες και η άνθηση του διαδικτύου δημιούργησε επίσης ευκαιρίες σε επιτήδειους για νέους τύπους εγκλημάτων. Σύμφωνα με το National Institute of Standards and Technology (NIST) ο ορισμός του Κυβερνοχώρου (Cyberspace) είναι ο παρακάτω:

'Ο παγκόσμιος τομέας που εμπεριέχεται στο πληροφοριακό περιβάλλον που αποτελείται από το αλληλεξαρτώμενο δίκτυο των υποδομών πληροφοριακών συστημάτων συμπεριλαμβανομένου του διαδικτύου , τηλεπικοινωνιακών δικτύων, συστημάτων Η/Υ και ενσωματωμένων επεξεργαστών και ελεγκτών.'[1]

Όποιες εγκληματικές ενέργειες διαπράττονται στον κυβερνοχώρο κατατάσσονται στα Κυβερνοεγκλήματα.
Όπως αναφέρθηκε λίγες γραμμές παραπάνω η ανθρώπινη κοινωνία βασίζεται υπερβολικά στις ηλεκτρονικές συσκευές και το διαδίκτυο με αποτέλεσμα το κυβερνοέγκλημα να

αυξάνεται δραματικά και να προκαλεί τεράστιες ζημίες σε κυβερνήσεις , επιχειρήσεις αλλά και τον άνθρωπο σε ατομικό επίπεδο.
Το World Economic Forum με άρθρο του παραθέτει ότι το 2023 εκτελούνταν περίπου 2.220 κυβερνοεπιθέσεις την ήμερα και μία από τις πιο μεγάλες επιθέσεις του 2023 ήταν στο US State Department που αναλογεί με το υπουργείο εξωτερικών κατά την οποία κλάπηκαν δεκάδες χιλιάδες ηλεκτρονικά μηνύματα (emails).[2]
Παρατηρείται ότι ακόμα και τα ανώτερα κυβερνητικά ιδρύματα που διαθέτουν αυξημένη κυβερνοασφάλεια πέφτουν θύματα από εγκληματίες του κυβερνοχώρου.

Υποενότητα 2.3.1 – Τύποι Κυβερνοεπιθέσεων

Το National Institute of Standards and Technology (NIST) ορίζει την Κυβερνοεπίθεση (Cyber Attack) ως:

'Οποιοδήποτε είδος κακόβουλης δραστηριότητας που επιχειρεί να συλλέξει, να διαταράξει, να αρνηθεί, να υποβαθμίσει ή να καταστρέψει πόρους του συστήματος πληροφοριών ή τις ίδιες τις πληροφορίες.'[3]

Ο επιτιθέμενος, είτε είναι ένα ή περισσότερα άτομα είτε ένας οργανισμός είτε μια κυβέρνηση, αναζητάει να επωφεληθεί από αυτή την ενέργεια.

Οι συχνότερες και πιο μελετημένες κυβερνοεπιθέσεις είναι:

- Επιθέσεις DoS και DDoS

  Η άρνηση υπηρεσίας denial of service (DoS) έχει ως σκοπό να υπερφορτώσει ένα σύστημα κυρίως ένα server με αιτήματα προς επεξεργασία ώστε να στραγγίξει τους πόρους και το σύστημα να καταρρεύσει. Η κατανεμημένη άρνηση υπηρεσίας (DDoS) είναι το ίδιο η μόνη διαφορά ότι επιστρατεύονται πολλοί Η/Υ που θα στείλουν αιτήματα στον στόχο.

- Επιθέσεις Phishing

Το ηλεκτρονικό "ψάρεμα" είναι μία τακτική κατά την οποία ένας κακόβουλος αποστέλλει σε ένα στόχο email τα οποία εμφανίζονται ως αυθεντικά και πως προέρχονται από έμπιστες και νόμιμες πηγές και αποσκοπούν ο στόχος να "τσιμπήσει το δόλωμα" και να πατήσει τον σύνδεσμο που περιέχει το email κατεβάζοντας ένα κακόβουλο λογισμικό στον Η/Υ του.

- Ransomware

Αποτελεί σύνθεση των λέξεων ransom (λύτρα) και malware (κακόβουλο λογισμικό). Το συγκεκριμένο λογισμικό αφού εισχωρήσει στην ηλεκτρονική συσκευή του θύματος κλειδώνει το σύστημα και για να το ξεκλειδώσει ζητάει από το θύμα να πληρώσει ένα χρηματικό ποσό. Αν γίνει η πληρωμή τότε παρέχεται ο κωδικός ξεκλειδώματος.

- DNS spoofing

Η επίθεση Domain Name System spoofing (πλαστογράφηση) γίνεται όταν ο κυβερνοεγκληματίας τροποποιεί τις καταγραφές του DNS με αποτέλεσμα το θύμα να μεταφέρεται σε μία παραποιημένη ιστοσελίδα από την οποία μπορεί να κατεβάσει κακόβουλα λογισμικά ή να δώσει ευαίσθητες προσωπικές πληροφορίες

- Trojan Horse

Ο λεγόμενος Δούρειος Ίππος είναι ένα κακόβουλο πρόγραμμα το οποίο εμφανίζεται ως νόμιμο τόσο στο τείχος προστασίας (firewall) της συσκευής όσο και στον χρήστη. Μέσα του όπως κρύβεται ένα άλλο malware το οποίο ανάλογα τον τύπο του θα δράσει.

- SQL Έγχυση (Injection)

Η απόπειρα ενός κυβερνοεγκληματία να εκτελέσει επιπλέον εντολές σε μια βάση δεδομένων καταλογίζεται ως επίθεση SQL injection. Οι εντολές που προσπαθούν να

εισχωρήσουν δια μέσου των εισαγωγών του χρήστη με απώτερο σκοπό την τροποποίηση λειτουργιών της εφαρμογής είναι γραμμένες στη προγραμματιστική γλώσσα SQL εξ ου και το όνομα της επίθεσης. Σε περίπτωση επιτυχίας της επίθεσης θα προκληθούν τα αποτελέσματα κυμαίνονται από διαρροές πληροφοριών, αλλαγή/διαγραφή δεδομένων ή μη εξουσιοδοτημένη πρόσβαση στη βάση δεδομένων.

- Cross Site Scripting (XSS)

Μία διαφορετική επίθεση έγχυσης όπου ο κυβερνοεγκληματίας προσπαθεί να εγχύσει κακόβουλα αρχεία δέσμης εντολών (scripts) σε έμπιστες ιστοσελίδες. Η επίθεση XSS συμβαίνει όταν ο επιτιθέμενος χρησιμοποιεί μια ιστοσελίδα για να στείλει τον κακόβουλο συνδέσμους ιστοσελίδων σε άλλον χρήστη. Αν η ιστοσελίδα δεν είναι ικανή να επεξεργαστεί σωστά τα δεδομένα και ο χρήστης πατήσει τον σύνδεσμο τότε ο κακόβουλος κώδικας θα εκτελεστεί στο σύστημα του θύματος με τελικό αποτέλεσμα την κλοπή των cookies. Τα cookies είναι μικρά αρχεία κειμένου με πληροφορίες που είναι αποθηκευμένα στον φυλλομετρητή.

Ενότητα 2.4 - Ασφάλεια Πληροφοριών

Η Ασφάλεια Πληροφοριών (ΑΠ / Information Security / InfoSec) είναι η προστασία των σημαντικών πληροφοριών ενός οργανισμού, όπως ψηφιακά αρχεία και δεδομένα , φυσικά μέσα ακόμα και ανθρώπινες συζητήσεις, ενάντια σε μη εξουσιοδοτημένη πρόσβαση, αποκάλυψη, χρήση ή τροποποίηση.

Επίσης η ΑΠ είναι υπεύθυνη και για τη μεταφορά των σωστών πληροφοριών σε υπόλοιπα εξουσιοδοτημένα μέλη.

Το National Institute of Standards and Technology (NIST) προσδιορίζει τον όρο Κυβερνοασφάλεια (Cybersecurity) ως εξής:

“Η ικανότητα της προστασίας ή της άμυνας της χρήσης του κυβερνοχώρου από κυβερνοεπιθέσεις.”[4]

Η διαφορά ΑΠ και κυβερνοασφάλειας έγκειται στο γεγονός ότι η ΑΠ προστατεύει όλες τις μορφές δεδομένων , φυσικών και ψηφιακών, από όλων των ειδών απειλών είτε είναι φυσικές καταστροφές είτε ανθρώπινη παρέμβαση . Η κυβερνοασφάλεια αποτελεί υποκατηγορία της ΑΠ διότι προστατεύει μονάχα ψηφιακής μορφής δεδομένων – πληροφοριών. Αντιμετωπίζει και μετριάζει τεχνολογικές απειλές εξασφαλίζοντας την ακεραιότητα των δεδομένων.
Το 1977 θεσπίστηκαν οι τρεις θεμελιώδης αρχές της Ασφάλειας Πληροφοριών οι οποίες διαμόρφωσαν το ακρωνύμιο CIA.

- Εμπιστευτικότητα (Confidentiality)

  Αφορά την διασφάλιση ότι θα αποτραπεί η ανάγνωση και πρόσβαση εμπιστευτικών δεδομένων και αρχείων από μη εξουσιοδοτημένα μέλη και ομάδες.

- Ακεραιότητα (Integrity)

  Αφορά την διατήρηση της ακρίβειας των δεδομένων στη βάση δεδομένων. Η ακεραιότητα εφαρμόζεται σε όλο το φάσμα, αποτρέπει την σκόπιμη τροποποίηση ή διαγραφή δεδομένων από κακοήθης αλλά και από καλοπροαίρετους χρήστες που ηθελημένα ή μη προσπάθησαν να επέμβουν με μη εξουσιοδοτημένους τρόπους.

- Διαθεσιμότητα (Availability)

  Αφορά την εξασφάλιση των εξουσιοδοτημένων μελών στην πρόσβαση των πληροφοριών και αρχείων που επιθυμούν, όποτε το επιθυμούν. Επιπλέον για να είναι δυνατό κάτι τέτοιο πρέπει να γίνεται συντήρηση των εξαρτημάτων (hardware) και τακτική ενημέρωση των συστημάτων.

Η τριάδα CIA αποτελεί τη βάση της ΑΠ όσο αναφορά την πολιτική και την λήψη αποφάσεων είναι ανάγκη να αναφερθούμε και στους υπόλοιπους παράγοντες που θα απαρτίσουν ένα ολοκληρωμένο πλάνο ΑΠ:

- Διαχείριση Ρίσκου

Κάθε επιχείρηση χρησιμοποιεί τις αρχές της διαχείρισης ρίσκου για να καθορίσει το επίπεδο ρίσκου που μπορεί να αντιμετωπίσει όταν εκτελεί ένα σύστημα. Υπάρχει η δυνατότητα να τοποθετηθούν δικλίδες ασφαλείας και ελαφρυντικά μέσα για την μείωση του ρίσκου. Ο απόλυτος στόχος είναι η αύξηση των θετικών αποτελεσμάτων και ταυτόχρονα η αποφυγή των αρνητικών.

- Ταξινόμηση Δεδομένων

  Αναφέρεται στην ταξινόμηση των άκρως εμπιστευτικών δεδομένων τα οποία είναι επιτακτική ανάγκη να παραμείνουν ασφαλή και ακέραια και ταυτόχρονα να είναι εύκολα προσβάσιμα από εξουσιοδοτημένα άτομα.

- Μέσα και εμπιστευτικά συμφωνητικά

  Αναφέρεται σε φυσικά μέσα που κατέχουν πληροφορίες όπως εκτυπωμένες σελίδες πχ συμβόλαια ,συμφάσεις, οικονομικές αναφορές.

- Εκπαίδευση Χρήστη

  Κάθε επιχείρηση προτείνεται να ενημερώνει και να εκπαιδεύσει τους υπαλλήλους της ανά τακτά χρονικά διαστήματα στην προστασία των προσωπικών τους δεδομένων καθώς και των δεδομένων της επιχείρησης για την αποτροπή μοιραίων σφαλμάτων.

- Επιχειρηματική συνοχή και αποκατάσταση από καταστροφές

  Σε περιπτώσεις σφαλμάτων και αποτυχίας του software ή του hardware είναι αναγκαίο τα δεδομένα να παραμείνουν αναλλοίωτα και προσβάσιμα. Αυτό πραγματοποιείται με την δημιουργία αντιγράφων ασφαλείας (backups) ή με την διπλοτυπία των εξαρτημάτων.

- Ευρωπαϊκοί και κρατικοί νόμοι

Η προστασία των προσωπικών δεδομένων καλύπτεται και θεσμικά από κρατικούς και κανονισμούς που ορίζει η Ευρωπαϊκή Ένωση με σκοπό να προτρέπει τους πολίτες και τις επιχειρήσεις ποια προσωπικά δεδομένα να προφυλάσσουν και τις ποινές που προβλέπονται σε περιπτώσεις διαρροής ευαίσθητων πληροφοριών. Από το 2016 έχει τεθεί σε ισχύ ο κανονισμός του Γενικού Κανονισμού για την Προστασία Δεδομένων (ΓΚΠΔ) ή πιο γνωστό ως GDPR (General Data Protection Regulation) ο οποίος αναβάθμισε σημαντικά το προηγούμενο νομικό πλαίσιο και ενδυνάμωσε τα θεμελιώδη δικαιώματα και ελευθερίες των φυσικών προσώπων.

Ενότητα 2.5 - Βασικές έννοιες και ορισμοί στην Ασφάλεια Πληροφοριών

Ο **Ιδιοκτήτης** (Owner) , είτε φυσικό πρόσωπο είτε επιχείρηση, διαθέτει τα προσωπικά του **Περιουσιακά Στοιχεία** (Asset). Ως περιουσιακά στοιχεία θεωρούνται φυσικά αντικείμενα (σκληροί δίσκοι, Η/Υ) , ψηφιακές πληροφορίες και δεδομένα ακόμα και το προσωπικό του δίκτυο. Τα περιουσιακά του στοιχεία κατέχουν μια **Αξία** (Value). Ο **Κυβερνοεγκληματίας** (Hacker) έχει ως σκοπό να αποκτήσει πρόσβαση στα περιουσιακά στοιχεία του ιδιοκτήτη. **Θύμα** (Victim) κατονομάζεται ο ιδιοκτήτης που επηρεάζεται από τις ενέργειες του κυβερνοεγκληματία. Άρα κάθε περιουσιακό στοιχείο είναι εκτεθειμένο σε ενέργειες προερχόμενες από κυβερνοεγκληματίες ανά πασά στιγμή. Οποιαδήποτε πιθανή ενέργεια που στοχεύει στην παραβίαση ασφάλειας , στην ανάγνωση , κλοπή ή παραποίησης ενός περιουσιακού στοιχείου ονομάζεται **Απειλή** (Threat). Μια απειλή χαρακτηρίζεται ως:

- **Εκούσια απειλή**: Αναφέρονται σε απειλές που συνέβησαν οικειοθελώς και εσκεμμένα από χρήστες.
- **Ακούσια απειλή:** Αναφέρονται σε απειλές που προέκυψαν από απεριοκεψία , λάθη , σφάλματα από ελλιπείς γνώσεις και χωρίς την επιθυμία του χρήστη να προκαλέσει ζημία.
- **Φυσική απειλή:** Αναφέρεται σε απειλές από φυσικά φαινόμενα, πχ πυρκαγιές, πλημμύρες.

- **Αποτυχία Υλικού:** Αναφέρεται σε απειλές από την φθορά και καταστροφή τεχνολογικών εξαρτημάτων, πχ "κάψιμο" μητρικής – σκληρού δίσκου – τροφοδοτικού κλπ.

Στην περίπτωση που κάποιος κυβερνοεγκληματίας – κακόβουλος χρήστης κατορθώσει να αποκτήσει πρόσβαση σε ένα η περισσότερα συστήματα , πόρους, δεδομένα ή επιχειρήσει να διακινδυνεύσει την ακεραιότητα , εμπιστευτικότητα και διαθεσιμότητα του συστήματος τότε το γεγονός χαρακτηρίζεται ως **Επίθεση** (Attack).
Για την επιτυχία μιας επίθεσης ο κυβερνοεγκληματίας πρέπει να παρατηρήσει να ανιχνεύσει και να εκμεταλλευτεί ένα ή περισσότερα **κενά ασφαλείας** (Security Flaws). Τα κενά ασφαλείας χαρακτηρίζονται και ως **Ευπάθειες** (Vulnerabilities).
Συσκευές και ενέργειες οι οποίες αποσκοπούν στην ανίχνευση ή αντιμετώπιση και εναντίωση ενάντια σε οποιαδήποτε μορφής απειλή , επίθεση και ευπάθειας χαρακτηρίζεται ως **Αντίμετρα** (Countermeasures). Τα αντίμετρα μπορεί να διαδραματίσουν σπουδαιότερο ρόλο ακόμα και από την προστασία διότι προσπαθούν να ελαχιστοποιήσουν την έκθεση και την ζημία ακόμα και μετά την επίθεση.

Υποενότητα 2.5.1 – Τύποι Hacker

Στην Ενότητα 2.5 αναφέρεται ο κυβερνοεγκληματίας ως **Hacker**. Ωστόσο αυτή η διατύπωση δεν είναι απόλυτα ορθή καθώς οι υποκατηγορίες ενός hacker είναι αρκετές και συνάμα διαχωρίζει τα άτομα που εγκληματούν και τα άτομα τα οποία προασπίζουν και προστατεύουν πληροφοριακά συστήματα από κυβερνοεπιθέσεις. Ως hacker ορίζεται το άτομο με γνώσεις πληροφορικής κυρίως σε θέματα ασφάλειας. Ο διαχωρισμός των hackers γίνεται με κύρια βάση τις ηθικές αρχές, τους τρόπους που χρησιμοποιεί τις τεχνολογικές του γνώσεις και τον σκοπό του εκάστοτε πληροφορικού.
Συνοπτικά οι σημαντικότερες είναι:

- **Black Hat Hackers:** Αναφέρεται στους κακόβουλους χρήστες – κυβερνοεγκληματίες οι οποίοι έχουν σκοπό την κλοπή ή την καταστροφή ζωτικών δεδομένων. Χαρακτηρίζονται και ως Crackers σύνθεση των λέξεων Crack (σπάω) και Hacker.

- **White Hat Hackers:** Αναφέρεται στους ηθικούς και καλοήθης πληροφορικούς οι οποίοι είναι εξουσιοδοτημένοι, πιστοποιημένοι και έμπιστοι hackers. Εργάζονται για τις κυβερνήσεις ή για επιχειρήσεις και πραγματοποιούν ελέγχους με σκοπό την ανίχνευση ευπαθειών και τρυπών (loopholes) τα οποία θα μπορούσαν να εκμεταλλευτούν από κακοήθης και να διαπεράσουν την προστασία των συστημάτων.
- **Gray Hat Hackers:** Αναφέρεται στην κατηγορία hacker οι οποίοι δεν είναι νομικά εξουσιοδοτημένοι και ισορροπούν ανάμεσα στους white και black hat hackers. Έχουν καλές και κακές προθέσεις και ανάλογα με τις πράξεις τους χαρακτηρίζονται.
- **Blue Hat Hackers:** Είναι παρόμοια κατηγορία με τους White Hat Hackers με την μόνη διαφορά ότι αποτελούν υπάλληλοι μιας εταιρίας και εργάζονται πάνω στα συστήματα της αλλά και τα προϊόντα της προτού διανεμηθούν στο ευρύ κοινό.
- **Red Hat Hackers:** Είναι η κατηγορία Hacker οι οποίοι στοχεύουν και εξαπολύουν επιθέσεις ενάντια σε Black Hat Hackers ώστε να αναστατώσουν τις επιθέσεις τους ή για να ανταποδώσουν. Είναι μια ιδιαίτερη κατηγορία διότι ενώ οι πράξεις τους είναι θετικές, οι πρακτικές και τα μέσα που χρησιμοποιούν είναι ανάλογα των Black Hat.

Καταλήγουμε στο συμπέρασμα ότι ο όρος Hacker δεν είναι αυτόματα αρνητικός ή θετικός αλλά διαμορφώνεται από τις πράξεις και προθέσεις του ατόμου.

Ενότητα 2.6 - Ευπάθειες

Το National Institute of Standards and Technology (NIST) ορίζει την Ευπάθεια ως εξής:

'Αδυναμία σε ένα πληροφοριακό σύστημα, διαδικασιών ασφάλειας συστήματος, εσωτερικών ρυθμίσεων ή υλοποίηση που θα μπορούσε να εκμεταλλευτεί ή να ενεργοποιηθεί από μία απειλητική πηγή'[5]

Αυτή η αδυναμία είναι ικανή να εκθέσει ολόκληρο το πληροφοριακό σύστημα. Τι προκαλεί ευπάθειες σε ένα πληροφοριακό σύστημα;

- **Ευπάθεια Λογισμικού**: Η προβληματική κατασκευή του λογισμικού, σφάλματα στην προγραμματιστική γλώσσα του λογισμικού ,ρυθμιστικά σφάλματα και

απροσεξίες προκαλούν αδυναμίες. Κάθε λογισμικό αναβαθμίζεται με σκοπό το μπάλωμα(patch) ευπαθειών που βρήκαν οι προγραμματιστές που το δημιουργήσαν. Συχνά κυβερνοεγκληματίες ανακαλύπτουν ευπάθειες και τις εκμεταλλεύονται πριν προλάβουν οι προγραμματιστές να δημιουργήσουν το κατάλληλο patch. Αυτό το είδος ευπαθειών ονομάζεται **Zero Day**.Η μη τακτική ενημέρωση των λειτουργικών προγραμμάτων αφήνει εκτεθειμένες παλαιές ευπάθειες που πιθανό να έγιναν γνωστές αυξάνοντας το ενδεχόμενο για zero day επίθεση.

- **Ευπάθεια Δικτύου**: Συμπεριλαμβάνονται ευπάθειες στο λογισμικό, το υλικό και στις επεξεργασίες που διέπουν τη ροή του φόρτου εργασίας σε ένα δίκτυο. Είναι ιδιαίτερα δύσκολες στην ανίχνευση τους καθώς η ευπάθεια μπορεί να βρίσκεται σε όλο το εύρος του μοντέλου OSI.
- **Φυσικές ευπάθειες**: Αφορούν κυρίως μεγάλες εταιρίες τύπου cloud infrastructure και data centers όπου δεν γίνεται σωστή καταγραφή γεγονότων, πχ αλλαγή αποθηκευτικών συσκευών, υπάρχει περιορισμός εισόδου σε ορισμένους χώρους. Επίσης μπορεί να κλαπεί εξοπλισμός ή ο προσωπικός εξοπλισμός ενός υπαλλήλου να προκαλέσει ζημιά.
- **Ρυθμιστικές και επεξεργαστικές ευπάθειες**: Ακόμα και αν το λογισμικό και τα μηχανικά εξαρτήματα λειτουργούν χωρίς καμία ευπάθεια οι εσφαλμένες διαμορφώσεις είναι ικανές να εκθέσουν το σύστημα. Συχνό λάθος είναι η χρήση προϊόντων με προεπιλεγμένα διαπιστευτήρια διαχείρισης τα οποία να είναι γνωστά σε κυβερνοεγκληματίες.
- **Εσωτερικές απειλές**: Αποτελούν την κρισιμότερη ευπάθεια για την ασφάλεια ενός πληροφοριακού συστήματος. Εκτιμάται ότι πάνω από 90% των ατυχημάτων κυβερνοασφάλειας οφείλονται στον ανθρώπινο παράγοντα. Η σοβαρότητα στην συγκεκριμένη ευπάθεια είναι ότι η ευπάθεια είναι ο άνθρωπος ο οποίος έχει πρόσβαση σε κρίσιμα τεχνολογικά συστήματα και ευαίσθητες πληροφορίες. Ακόμα και χωρίς εξουσιοδότηση συχνά λαμβάνει τις πληροφορίες από άλλα μέλη της εταιρίας με αποτέλεσμα να καταστεί αδύναμο κάθε μέτρο προστασίας. Είναι αρκετά καταγεγραμμένα γεγονότα όπου δυσαρεστημένοι υπάλληλοι προκάλεσαν ζημία στις εταιρίες τους για προσωπικούς λόγους ή σε ακραίες περιπτώσεις υπάλληλοι εκβιάστηκαν από εξωτερικούς να παραδώσουν πληροφορίες. Παρόλα αυτά λάθη μπορούν να προκύψουν από απροσεξία, χωρίς επίγνωση του λάθους.

Το Common Vulnerabilities and Exposures (CVE) σύστημα παρέχει μία εξαιρετικά μεγάλη βάση δεδομένων για γνωστά ζητήματα ασφάλειας πληροφοριών. Το CVE παρέχει ένα αξιόπιστο τρόπο ώστε ενδιαφερόμενοι, επιχειρήσεις και ακαδημαϊκοί να ανταλλάξουν γνώσεις και πληροφορίες για θέματα κυβερνοασφάλειας. Χρηματοδοτείται από το US National Cyber Security Division και παραδόθηκε στο κοινό το 1999.
Ο αριθμός των γνωστών ευπαθειών CVEs παγκοσμίως το 2023 έφτασε τις 29.065 μετά από μέτρηση της Statista.[6]
Είναι άξιο σημείωσης να αναφερθεί ότι αποκλείεται να υπάρξει το τέλειο σύστημα το οποίο δεν θα έχει καμία ευπάθεια και κανένα τρωτό σημείο, αυτό το γεγονός είναι που ωθεί την παγκόσμια κοινότητα που ασχολείται με την ΑΠ και την κυβερνοασφάλεια να βελτιώνεται ακόμα περισσότερο και με μεγαλύτερη ένταση για την ελαχιστοποίηση των γνωστών ευπαθειών, την βέλτιστη κατασκευή συστημάτων και δικτύων και την πιο αποτελεσματική ενημέρωση των ανθρώπων που δεν είναι εξοικειωμένοι με το αντικείμενο.

Ενότητα 2.7 - Ιδιωτικότητα

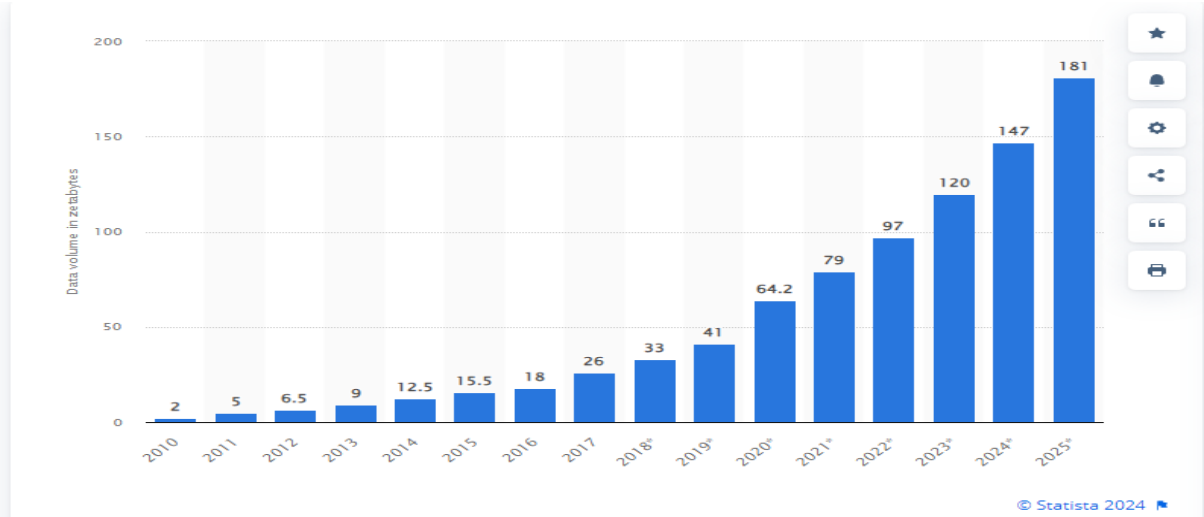
Ο όρος **Ιδιωτικότητα** (Privacy) είναι πολύ αόριστος να διατυπωθεί εκ ολοκλήρου ηθικά και νομικά. Το 1890 οι δικηγόροι Samuel D. Warren και Louis Brandeis με άρθρο τους στο Harvard Law Review υποστήριξαν ότι,

“ *το δικαίωμα στην ιδιωτικότητα είναι*
το δικαίωμα να μείνεις μόνος. ”[7]

Η έννοια της ιδιωτικότητας στον τομέα του κυβερνοχώρου είναι ακόμα πιο περίπλοκη και ανοιχτή ωστόσο μπορεί να οριστεί ως το δικαίωμα των ανθρώπων να καθορίζουν για τους εαυτούς τους πότε, με ποιον τρόπο και ποια προσωπικά τους δεδομένα επιτρέπεται να συλλεχθούν, αποθηκευτούν, επεξεργαστούν, πουληθούν ή διαδοθούν από άλλους. Με απλά λόγια ο κάθε άνθρωπος έχει το δικαίωμα του ελέγχου των προσωπικών του δεδομένων.

Τι είναι όμως τα προσωπικά δεδομένα;
Σύμφωνα με την Ευρωπαϊκή Επιτροπή,
"*Προσωπικά δεδομένα είναι οποιεσδήποτε πληροφορίες που αχετίζονται με έναν γνωστό ή αναγνωρίσιμο άτομο. Διάφορα κομμάτια πληροφοριών τα οποία αν συγκεντρωθούν έχουν την δυνατότητα να ταυτοποιήσουν ένα συγκεκριμένο άτομο, αποτελούν και αυτά προσωπικά δεδομένα*."[8]

Η ιδιωτικότητα στον κυβερνοχώρο λαμβάνει άλλη μορφή ανάλογα την περίσταση.
Η ιδιωτικότητα της συμπεριφοράς και των πράξεων στον κυβερνοχώρο αναφέρεται στο δικαίωμα του ατόμου να είναι ελεύθερο να εκφραστεί και να στηρίξει οτιδήποτε νιώθει και πιστεύει ότι τον εκφράζει. Η ιδιωτικότητα της επικοινωνίας αναφέρεται ότι το άτομο μπορεί να επικοινωνήσει με οποιοδήποτε χωρίς το περιεχόμενο της επικοινωνίας να διακόπτεται. Επίσης το περιεχόμενο πρέπει να είναι προστατευμένο. Η ιδιωτικότητα της τοποθεσίας αναφέρεται ότι το άτομο δεν επιτρέπεται να παρακολουθείτε και να καταγράφεται που βρίσκεται και ποια σημεία επισκέπτεται. Τέλος η ιδιωτικότητα των σχέσεων αναφέρεται στο δικαίωμα του ατόμου να είναι φίλος και να αλληλοεπιδράει με όποιον επιθυμεί χωρίς η σχέση αυτή να χρησιμοποιηθεί εναντίον του.
Στον κυβερνοχώρο η ψηφιακή παρουσία ενός ανθρώπου, επώνυμη ή ανώνυμη, αφήνει ψηφιακά αποτυπώματα, τα γνωστά ως δεδομένα. Ο κάθε άνθρωπος δημιουργεί και λαμβάνει καθημερινά μεγάλο όγκο δεδομένων με δικές του αποφάσεις ωστόσο πολλές εφαρμογές και συστήματα που έχει στην κατοχή του κάνουν μετάδοση και λήψη δεδομένων με την συγκατάθεση του ίδιου ή και όχι.
Ο όγκος των δεδομένων/πληροφοριών που δημιουργήθηκαν, κρατήθηκαν, αντιγράφηκαν και καταναλώθηκαν παγκοσμίως από το 2010 έως το 2020 με προβλέψεις για την περίοδο 2021-2025 από ανάλυση της Statista.



Εικόνα 2.1

(https://www.statista.com/statistics/871513/worldwide-data-created/)

Η ανάλυση καταγράφει τον όγκο των δεδομένων σε zettabytes!

Με αυτή την απίστευτη μεταφορά δεδομένων στο διαδίκτυο ήταν επιτακτική ανάγκη να θεσμοθετηθούν νόμοι για την προστασία των προσωπικών δεδομένων των πολιτών. Όπως αναφέρθηκε και στην Ενότητα 2.4 η Ευρωπαϊκή Ένωση έθεσε σε εφαρμογή το GDPR. Το GDPR εστίαζε σε δύο κατηγορίες στην προστασία των δεδομένων και στην ιδιωτικότητα των δεδομένων. Η προστασία δεδομένων επιβάλλει σε κυβερνήσεις, οργανισμούς και επιχειρήσεις να λαμβάνουν τα απαραίτητα μέτρα ώστε τα ευαίσθητα προσωπικά δεδομένα που τους εμπιστεύτηκαν οι χρήστες να μην κλαπούν ή διαρρεύσουν ελεύθερα. Επίσης υποχρεώνονται να ενημερώνουν τον χρήστη ποια δεδομένα θα συλλέξουν και πως θα τα χρησιμοποιήσουν αφήνοντας στην κρίση του ατόμου να δώσει την συγκατάθεση του ή να αρνηθεί. Το GDPR βασίζεται στις έξι αρχές για το χειρισμό των ευαίσθητων προσωπικών δεδομένων.

1. Τα προσωπικά δεδομένα θα πρέπει να υποβληθούν σε επεξεργασία με νόμιμο, δίκαιο και διαφανή τρόπο.

2. Τα προσωπικά δεδομένα θα πρέπει να συλλέγονται για συγκεκριμένους, ακριβής και νόμιμους σκοπούς.

3. Τα προσωπικά δεδομένα θα πρέπει να είναι επαρκή, σχετικά και να περιορίζονται στα απολύτως απαραίτητα.

4. Τα προσωπικά δεδομένα θα πρέπει να είναι ακριβής, και όπου κρίνεται απαραίτητο, να ενημερώνονται.

5. Τα προσωπικά δεδομένα θα πρέπει να διατηρούνται για όσο είναι αναγκαία.

6. Τα προσωπικά δεδομένα θα πρέπει να επεξεργάζονται με κατάλληλο τρόπο για την διατήρηση της ασφάλειας.

Το GDPR δίνει το νομικό δικαίωμα στους ευρωπαίους πολίτες, αν καταλάβουν ότι μία επιχείρηση δεν τηρεί τις βασικές αρχές του GDPR, να αιτηθούν την ολική διαγραφή των προσωπικών του δεδομένων από τα μέσα αποθήκευσης της επιχείρησης. Ισχύει και για δεδομένα που παρείχε όσο ήταν ακόμα ανήλικος.

Αναφέρεται ως ‘το δικαίωμα στη λήθη’.[9]

ΚΕΦΑΛΑΙΟ 3 – Σχεδιασμός Ιστοσελίδας

Ενότητα 3.1 - Πρόγραμμα

Στο συγκεκριμένο κεφάλαιο θα αναλύσουμε τον σχεδιασμό και την λειτουργικότητα μίας ιστοσελίδας. Η παρακάτω ιστοσελίδα είναι κατηγορίας μέσου κοινωνικής δικτύωσης (social media) στην οποία παρουσιάζονται άρθρα με σχέση τον τομέα της πληροφορικής και δίνεται η δυνατότητα επικοινωνίας και αλληλεπίδρασης μεταξύ των χρηστών της.

Υποενότητα 3.1.1 – Χρήστης

Ο χρήστης έχει τη δυνατότητα να διαβάσει τα άρθρα που έχουν δημοσιευτεί στην ιστοσελίδα και να παρατηρήσει τα προφίλ άλλων χρηστών. Στην περίπτωση που αποφασίσει να δημιουργήσει το προσωπικό του προφίλ καλείται να καταθέσει συγκεκριμένα δεδομένα και είναι σε θέση πλέον να δημοσιεύει δικά του άρθρα, να αξιολογεί και να σχολιάζει άρθρα των υπολοίπων χρηστών καθώς και να έρχεται σε επικοινωνία με χρήστες.

Υποενότητα 3.1.2 – Άρθρα

Τα άρθρα συντάσσονται από δηλωμένους χρήστες της ιστοσελίδας και κατηγοριοποιούνται ανάλογα με τον τομέα στον οποίο αναφέρονται. Οι χρήστες έχουν την δυνατότητα να αξιολογήσουν το άρθρο επιλέγονταν εάν τους αρέσει ή όχι και επιπλέον να σχολιάσουν τις προσωπικές τους σκέψεις και ιδέες.

Υποενότητα 3.1.3 – Δωμάτια Επικοινωνίας

Τα δωμάτια επικοινωνίας (Chatrooms) δημιουργούνται από τους χρήστες οι οποίοι επιθυμούν την συνομιλία μέσω μηνυμάτων σε προσωπικό ή ομαδικό επίπεδο με άλλους χρήστες. Είναι εμφανή σε όλους ωστόσο η πρόσβαση γίνεται αυστηρά μόνο σε εγγεγραμμένα μέλη.

Ενότητα 3.2 - Τεχνολογίες που χρησιμοποιήθηκαν

Για την δημιουργία και ανάπτυξη την ιστοσελίδας χρησιμοποιήθηκαν οι παρακάτω τεχνολογίες.

Υποενότητα 3.2.1 – Python

Η γλώσσα προγραμματισμού Python μία διερμηνευόμενη, αντικειμενοστραφής γλώσσα προγραμματισμού υψηλού επιπέδου με ενσωματωμένες δομές δεδομένων με ιδιότητες δυναμικού προγραμματισμού. Κυκλοφόρησε το 1991 και ταχύτατα προτιμήθηκε από τους προγραμματιστές εξαιτίας της απλότητας της γλώσσας, στην αναγνωσιμότητα, την εκμάθηση και την χρήση της, καθώς και στις δυνατότητες που παρείχε. Είναι ανοιχτού λογισμικό (open source) και την διαχείριση της την έχει αναλάβει το Python Software Foundation. Στην εποχή μας υπάρχουν χιλιάδες βιβλιοθήκες που διευκολύνουν ιδιαίτερα τυπικές εργασίες και προσδίδουν υπεραρκετές δυνατότητες στους προγραμματιστές. Ένα ηχηρό μειονέκτημα της Python είναι ότι σαν διερμηνευόμενη γλώσσα υπολείπεται σε χρόνο εκτέλεσης έναντι μεταγλωττιζόμενων γλωσσών προγραμματισμού, πχ C, C++. Η Python χρησιμοποιείται στην ανάπτυξη διαδικτυακών εφαρμογών, προγραμμάτων αυτοματισμού, διαχείρισης δεδομένων, τεχνητής νοημοσύνης, παιχνιδιών και προγραμμάτων ασφάλειας.

Είναι εύκολα αντιληπτοί οι λόγοι για τους οποίους η Python είναι τόσο δημοφιλής και απαιτητική καθώς με τόσες εφαρμογές και με τεράστια κοινότητα να την υποστηρίζει είναι βέβαιο ότι θα συνεχίσει η ανάπτυξη, συντήρηση και βελτίωση πληθώρας εφαρμογών στο μέλλον.

Για περισσότερες πληροφορίες και για την λήψη της Python επιστεφτείτε την ιστοσελίδα www.python.org [10].

Υποενότητα 3.2.2 – Python Εικονικό Περιβάλλον

Το Python εικονικό περιβάλλον (venv) είναι ένας κατάλογος με ιδιαίτερη δομή. Είναι περιβάλλοντα απομονωμένα και ανεξάρτητα όπου μπορούν να εγκατασταθούν πακέτα Python. Χρησιμοποιούνται για να περιέχουν συγκεκριμένο διερμηνέα Python, βιβλιοθήκες λογισμικού και αρχεία που θα φανούν χρήσιμα στην υποστήριξη και ανάπτυξη ενός έργου. Η εγκατάσταση πακέτου σε venv πραγματοποιείται με την εντολή,

```
> python -m pip install (όνομα_πακέτου)
```

Υποενότητα 3.2.3 – Django Web Framework

Το web application framework είναι ένα δομημένο λογισμικό το οποίο δημιουργήθηκε με σκοπό την απλοποίηση της ανάπτυξης διαδικτυακών εφαρμογών. Παρέχουν μία ομάδα στοιχείων για την βοήθεια στην δημιουργία ιστοσελίδων γρηγορότερα και με μεγαλύτερη άνεση. Τα web frameworks παρέχουν ένα πρότυπο τρόπο για το χτίσιμο της ιστοσελίδας και έπειτα ο προγραμματιστής μπορεί να αναπτύξει οτιδήποτε επιθυμεί.

"*Django είναι ένα υψηλού επιπέδου python web framework το οποίο ενθαρρύνει την ταχύ ανάπτυξη και το καθαρό, ρεαλιστικό σχεδιασμό. Χτισμένο από έμπειρους προγραμματιστές,*

φροντίζει για μεγάλο μέρος της ταλαιπωρίας της ανάπτυξης εφαρμογών διαδικτύου, ώστε να μπορείς να εστιάσεις στο γράψιμο της εφαρμογής χωρίς να χρειαστεί να επανεφευρεις τον τροχό. Είναι ελεύθερο και λογισμικό ανοιχτοί κώδικα." [11]

Το Django εστιάζει στην αρχή DRY (Don't Repeat Yourself / Μην επαναλαμβάνεσαι) η οποία ενθαρρύνει την αποφυγή διπλοτυπιών στον κώδικα. Βασίζεται στην λογική την επαναχρησιμοποίησης των πραγμάτων που ορίστηκαν ήδη μια φορά βελτιώνοντας την διατήρηση του προγράμματος.

Το Django είναι εξοπλισμένο με μια πολύ ισχυρή τεχνολογία αποκαλούμενη ως ORM (Object Relational Mapper) η οποία ως λειτουργία έχει την διευκόλυνση και εξομάλυνση της ανταλλαγής δεδομένων μεταξύ ΒΔ και τα μοντέλα που ανήκουν στην εφαρμογή, δίνοντας την δυνατότητα να γίνεται η αλληλεπίδραση με την ΒΔ σχεδόν με τον παρόμοιο τρόπο που γίνεται με την SQL.

Η ευελιξία του framework είναι σημαντική καθώς είναι ικανό για την ανάπτυξη σχεδόν κάθε είδους διαδικτυακής εφαρμογής, είναι συμβατό με αρκετές βάσεις δεδομένων και συνεργάσιμο με client frameworks (πχ ReactJS).

Η ταχύτητα στην ανάπτυξη των εφαρμογών με τη χρήση οφείλεται στη πληθώρα βιβλιοθηκών που επιλύουν κοινότυπες και κλασσικές λειτουργίες και συνάμα η χρησιμοποίηση των ίδιων βιβλιοθηκών από πολλές εφαρμογές προσφέρει σταθερότητα, ασφάλεια και γρήγορη κατανόηση και αντίληψη από τους προγραμματιστές. Επίσης η κοινότητα του Django αναπτύσσει εφαρμογές οι οποίες μπορούν αυτούσιες να χρησιμοποιηθούν σε οποιοδήποτε Django πρόγραμμα . Η επαναχρησιμοποίηση κώδικα κάνει τα προγράμματα πιο συντηρήσιμα.

Το συγκεκριμένο framework είναι πολύ καλά τεκμηριωμένο λόγω της μεγάλης κοινότητας και επιλέγεται κυρίως για την ταχύτητα που προσδίδει στην ανάπτυξη , την επεκτασιμότητα

και δυνατότητα βελτιστοποίησης που προσφέρει και στην σοβαρότητα που δίνεται στην ασφάλεια ώστε να αποφεύγονται αρκετά σφάλματα που θα μπορούσαν να εκθέσουν την ασφάλεια της εφαρμογής.

Η ιστοσελίδα www.djangoproject.com παρέχει πληροφορίες για το framework [11].

Υποενότητα 3.2.4 – PostgreSQL

Το PostgreSQL είναι ένα σύστημα διαχείρισης βάσεων δεδομένων. Βασίζεται στη SQL (Structured Query Language) και είναι αξιόπιστο, αρκετά αποδοτικό και με μεγάλες δυνατότητες επεκτασιμότητας σύστημα βάσης δεδομένων. Είναι και αυτό λογισμικό ανοιχτού κώδικα. Επιλέγεται για τον τεράστιο όγκο δεδομένων που είναι ικανό να διαχειριστεί και να αποθηκεύσει καθώς και την ταυτόχρονη εξυπηρέτηση πολλαπλών χρηστών. Το PostgreSQL συμμορφώνεται με ιδιότητες του ACID (atomicity(ατομικότητα) ,consistency(συνέπεια), isolation(απομόνωση), durability(ανθεκτικότητα)) για να είναι αξιόπιστες οι συναλλαγές των βάσεων δεδομένων

"Για να διασφαλιστεί η αξιοπιστία, ο πλεονασμός δεδομένων, η υψηλή διαθεσιμότητα και η ανάκτηση από καταστροφές, η PostgreSQL προσφέρει λειτουργίες όπως καταγραφή εγγραφής εκ των προτέρων (WAL), αναπαραγωγή master-slave, ενεργή αναμονή και ανάκτηση σημείου σε χρόνο (PITR) και πολλά άλλα . Όλα αυτά επιτρέπουν την ανάπτυξη συμπλέγματος βάσεων δεδομένων πολλών κόμβων που μπορεί να αποθηκεύσει και να διαχειριστεί μεγάλους όγκους (terabyte) δεδομένων και εξειδικευμένα συστήματα που διαχειρίζονται petabyte." [12]

Το PostgreSQL λαμβάνει πολύ σοβαρά τα θέματα ασφαλείας και παρέχει αρκετές μεθόδους και τακτικές για να διασφαλίσει την ακεραιότητα και την ιδιωτικότητα των βάσεων δεδομένων. Όσο αναφορά την πιστοποίηση παρέχει SSPI μια τεχνολογία των Windows για

ασφαλή πιστοποίηση με μια μοναδική εγγραφή και LDAP που είναι μια μέθοδος επιβεβαίωσης κωδικού.

Ορισμένα από τα χαρακτηριστικά της είναι:

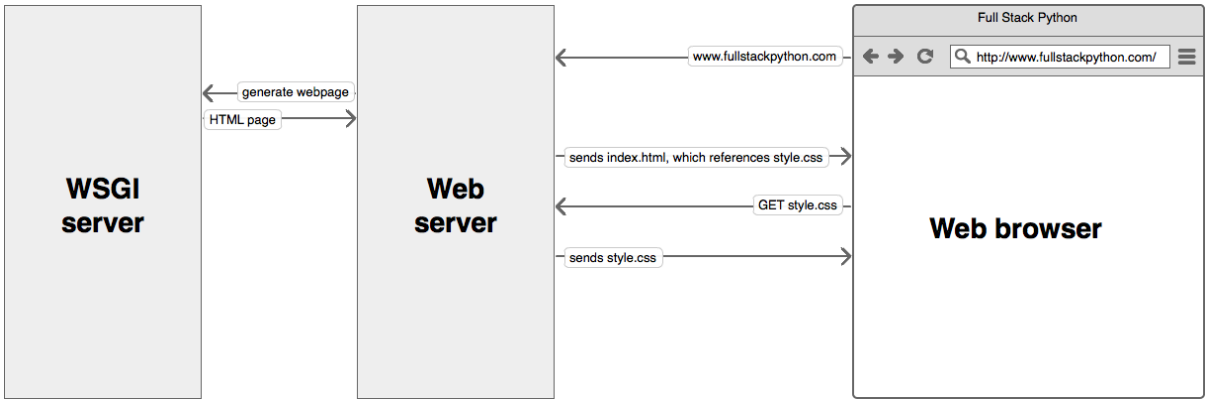
- Χαρακτηρίζεται ως ORDBMS (Object-Relational Database Management System)
- Περιβάλλον ανεκτικό σε σφάλματα.
- Συμβατό με ANSI SQL.
- Υποστήριξη δικτύου πελάτη – διακομιστή.
- Καταγραφή της εγγραφής.
- Υποστηρίζει γεωγραφικά στοιχεία για την ανάπτυξη εφαρμογών που χρησιμοποιούν γεωγραφικά δεδομένα.
- Εύκολη συντήρηση.
- Εύκολη στην εκμάθηση της από νέους προγραμματιστές.

Υιοθετείται όλο και περισσότερο από επιχειρήσεις λόγω του χαμηλού κόστους μίας και σαν open source δεν υπάρχουν κόστη αδειοδότησης, των προηγμένων χαρακτηριστικών και της φιλικής αρχιτεκτονικής για την ανάπτυξη cloud εφαρμογών.

Στη σελίδα www.postgresql.org υπάρχουν όλες οι απαραίτητες πληροφορίες για το PostgreSQL[12].

Υποενότητα 3.2.5 – WSGI

Το Web Server Gateway Interface (WSGI) είναι ο μεσολαβητής που βοηθάει στην επικοινωνία web servers και python web applications και στην επεξεργασία των αιτημάτων . Είναι απαραίτητο για Django εφαρμογές.



Εικόνα 3.1 WSGI
(https://www.fullstackpython.com/wsgi-servers.html)

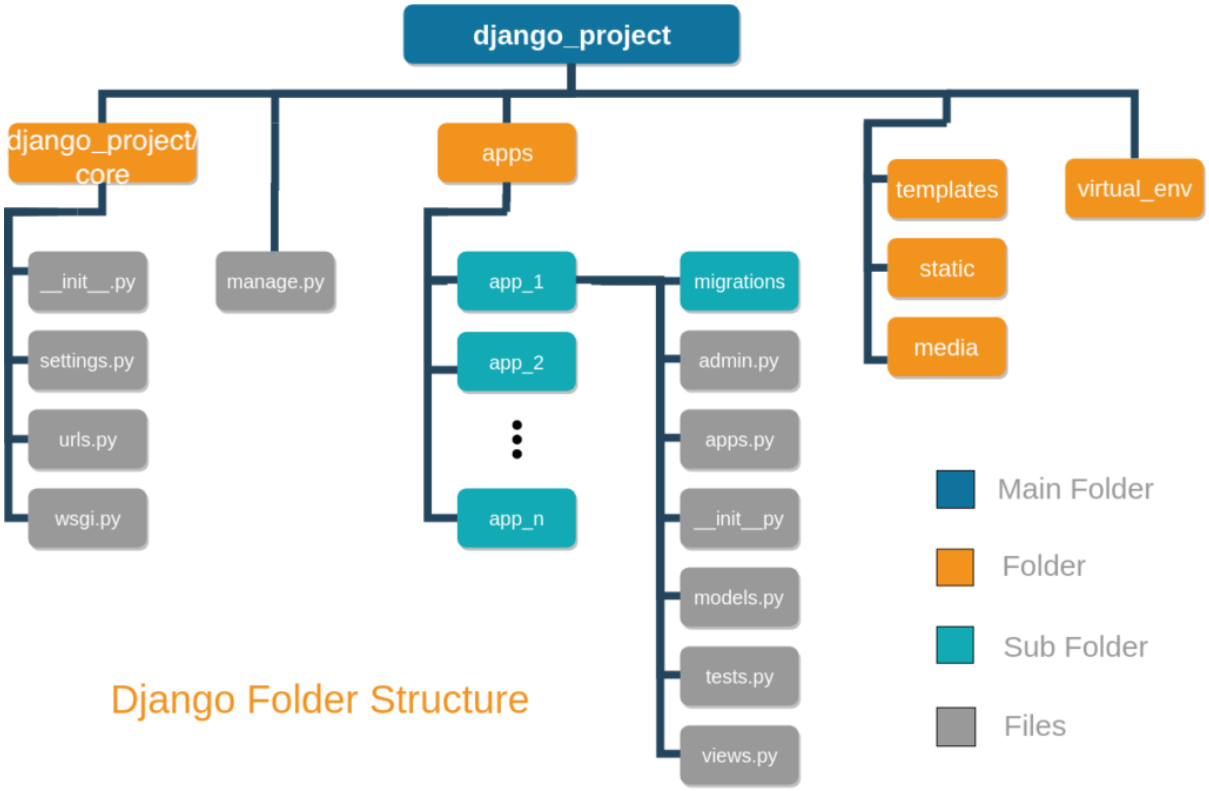
Το WSGI αποδεικνύεται χρήσιμο λόγω της ευελιξία όταν οι προγραμματιστές αποφασίσουν να αλλάξουν web stack στοιχεία χωρίς να είναι απαραίτητη η μορφοποίηση της εφαρμογής.

Με βάση τις πληροφορίες από την επίσημη ιστοσελίδα του Django υποστηρίζει τα GUNOCORN και uWSGI ως WSGI servers.

Ενότητα 3.3 - Αρχιτεκτονική Django

Όπως αναφέρθηκε και στην υποενότητα 3.2.2 το Django παρέχει εργαλεία για την γρήγορη και απλή ανάπτυξη διαδικτυακών εφαρμογών. Το βασικότερο στοιχείο που προσφέρει είναι η αρχιτεκτονική με την οποία δομείται η κάθε εφαρμογή. Αυτό συμβαίνει με σκοπό να γίνει αρμονική διευθέτηση των διαφορετικών σημείων της εκάστοτε εφαρμογής για να αποφευχθεί η σύγχυση των προγραμματιστών. Είναι σημαντικό να εξηγηθεί η σημασία και τα επιμέρους μέρη της αρχιτεκτονικής για να γίνει αντιληπτή η υλοποίηση μίας ιστοσελίδας με τη χρήση Django.

Υποενότητα 3.3.1 – Γενική Δομή Εφαρμογής Django



Εικόνα 3.2 Αρχιτεκτονική του Django
(https://rai-shahnawaz.medium.com/creating-a-django-project-the-right-way-14d230358d72)

Η εικόνα 3.2 καταδεικνύει τα βασικά μέρη της δομής. Αρχικά,

- **django_project** – Ο ριζικός κατάλογος (root directory) του προγράμματος
- **django_project/core** – Η βασική Django εφαρμογή
- **apps** – Οι εφαρμογές που απαρτίζουν τη Django εφαρμογή
- **templates/static/media** – Οι συγκεκριμένοι φάκελοι περιέχουν στατικό περιεχόμενο που χρησιμοποιεί η εφαρμογή.

- **virtual_env** – Είναι αποθηκευμένα τα αρχεία του εικονικού περιβάλλοντος.

- **manage.py** – Το συγκεκριμένο πρόγραμμα είναι χρήσιμο για διάφορες Django εντολές διαχείρισης καθώς και για εργασίες αποσφαλμάτωσης (debugging) και έναρξής ενός web server. Οι πιο κύριες εντολές που διαχειρίζεται το πρόγραμμα είναι

runserver – Εντολή εκκίνησης του διακομιστή του προγράμματος.

makemigration – Αν εισαχθούν νέα δεδομένα στην βάση δεδομένων επιβάλλεται με την συγκεκριμένη εντολή να ενημερώσουμε το πρόγραμμα.

migrate – Αν γίνουν αλλαγές στη βάση δεδομένων, ενημερώνει το πρόγραμμα.

Προτείνεται να μην γίνει καμία αλλαγή στο manage.py αρχείο!

Την στιγμή που θα δημιουργηθεί ένα Django πρόγραμμα, αυτομάτως θα κατασκευαστεί ο ριζικός κατάλογος (root directory) του προγράμματος, με το όνομα που επιλέχθηκε από τον χρήστη, ο οποίος θα περιέχει τα ζωτικά αρχεία για τις βασικές λειτουργίες του.

Η δημιουργία Django προγράμματος γίνεται με την είσοδο την παρακάτω εντολής:

> django-admin startproject (όνομα_προγράμματος)

η συγκεκριμένη εντολή παράγει την δομή:

όνομα_προγράμματος /

manage.py

όνομα_προγράμματος /

__init__.py

settings.py

urls.py

asgi.py

wsgi.py

- **__init__.py**
  Είναι ένα κενό αρχείο το οποίο παραμένει κενό και η λειτουργεία του είναι να ενημερώνει πως ο συγκεκριμένος κατάλογος είναι πακέτο Python.

- **settings.py**
  Αποτελεί το σημαντικότερο αρχείο μίας και σε αυτό εμπεριέχονται όλες τις ρυθμίσεις που διαμορφώνουν την εφαρμογή. Επιπλέον περιλαμβάνει και πληροφορίες για ενδιάμεσα λογισμικά (middleware), πρότυπα (templates), αρχεία και τις βάσεις δεδομένων. Το συγκεκριμένο αρχείο καθορίζει την λειτουργικότητα της εφαρμογής.

- **urls.py**
  Το συγκεκριμένο αρχείο κατευθύνει τον χρήστη στον κατάλληλο πόρο όταν δώσει κάποιο URL.

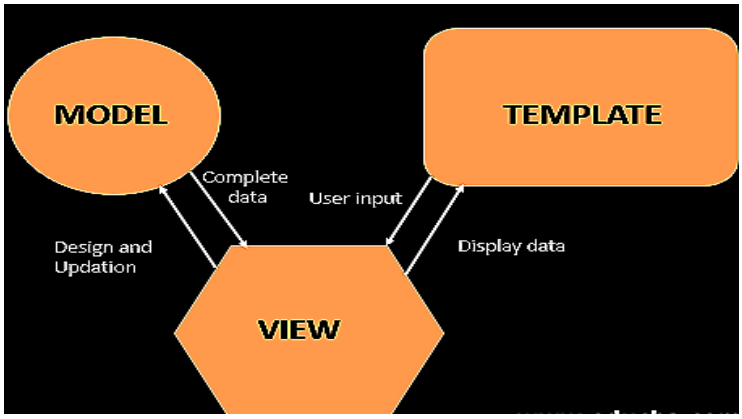
- **wsgi.py και asgi.py**
  Για το wsgi αναφερθήκαμε στην υποενότητα 3.2.4. Το asgi (Asynchronous Server Gateway Interface)είναι παρόμοιο με την διαφορά ότι χειρίζεται ασύγχρονα HTTP αιτήματα.

Υποενότητα 3.3.2 – Αρχιτεκτονική MVT

Το Django έχει την αρχιτεκτονική Μοντέλο – Εικόνα – Πρότυπο γνωστή ως MVT (Model – View – Template). Το MVT είναι ένα υπόδειγμα σχεδίασης λογισμικού το οποίο απαρτίζεται από τρία μέρη τα οποία αναλαμβάνουν συγκεκριμένες λειτουργίες.

Το **Μοντέλο** διαχειρίζεται τα δεδομένα και τη δομή της βάσης δεδομένων. Η **View** εμπεριέχει συναρτήσεις που λαμβάνουν HTTP αιτήματα κάνουν την απαραίτητη επεξεργασία και γυρνάνε HTTP απαντήσεις. Ουσιαστικά χειρίζεται την λειτουργικότητα της εφαρμογής. Το **Πρότυπο** χειρίζεται τη δομή του τι προβάλλεται στην οθόνη του χρήστη.

Η view για την επεξεργασία ορισμένων αιτημάτων μπορεί να χρειαστεί να λάβει δεδομένα από το μοντέλο. Μπορεί ωστόσο να χρειαστεί να προσθέσει, ενημερώσει ή να διαγράψει δεδομένα από την ΒΔ. Επιπλέον η view προσφέρει στο πρότυπο δεδομένα για να προβάλλει και λαμβάνει δεδομένα από τον χρήστη. Παρατηρείται ότι η view αλληλοεπιδράει και με το μοντέλο και με το πρότυπο αναλαμβάνοντας το ρόλο του ενδιάμεσου.



Εικόνα 3.3 MVT αρχιτεκτονική

(https://www.geeksforgeeks.org/django-project-mvt-structure/)

Η MVT δομή απλοποιεί σε μεγάλο βαθμό τον σχεδιασμό μίας εφαρμογής διότι κατηγοριοποιεί τις ευθύνες και τις λειτουργικότητες με αποτέλεσμα να παράγεται μία οργανωμένη και εύκολα συντηρήσιμη εφαρμογή.

Υποενότητα 3.3.3 – Δομή Εφαρμογής Django

Το Django πρόγραμμα είναι πιθανό να είναι μεγάλο σε έκταση και αρκετά περίπλοκο. Για να διαμορφώσουμε ένα διαχειρίσιμο, καλά δομημένο, ευανάγνωστο, εύκολο στη συντήρηση και στην επεκτασιμότητα πρόγραμμα προτιμάται το πρόγραμμα να απαρτίζεται από διάφορες εφαρμογές(apps). Η εφαρμογή είναι μια ανεξάρτητη ενότητα που πραγματοποιεί συγκεκριμένη λειτουργεία. Είναι μικρότερη σε μέγεθος και μπορεί να ενσωματωθεί σε πολλά και διαφόρων ειδών προγράμματα. Αυτή η δυνατότητα έχει ως αποτέλεσμα την αύξηση της αποδοτικότητας, την εξοικονόμησης χρόνου και προσπάθειας διότι οι προγραμματιστές βρίσκουν προκατασκευασμένες εφαρμογές που πραγματοποιούν τις απαιτούμενες λειτουργίες που επιθυμούν να εντάξουν στο πρόγραμμα τους. Η κανονικοποίηση εφαρμογών ενισχύει την συνέπεια του κώδικα και την σύνταξη της εφαρμογής διευκολύνοντας την συνεργασία της κοινότητας του django δημιουργώντας μία πληθώρα έτοιμων, αποτελεσματικών και δοκιμασμένων εφαρμογών.

Με το πρόγραμμα χωρισμένο σε κομμάτια ενισχύεται δραματικά η αποσφαλμάτωση και η αντιμετώπιση προβλημάτων καθώς οι προγραμματιστές είναι ικανοί να βρουν σε ποιο σημείο του κώδικα δημιουργείται πρόβλημα και να το επιλύσουν.

Η δημιουργία Django εφαρμογής γίνεται με την είσοδο την παρακάτω εντολής:

> python manage.py startapp (όνομα_εφαρμογής)

η συγκεκριμένη εντολή παράγει την δομή:

όνομα_εφαρμογής/

__init__.py

admin.py

apps.py

migrations/

__init__.py

models.py

tests.py

views.py

- **__init__.py:** Ορίζει ότι η εφαρμογή είναι πακέτο Python.

- **admin.py:** Σε αυτό το αρχείο καταχωρούνται τα μοντέλα στη διεπαφή του διαχειριστή.
- **apps.py:** Περιέχει ρυθμίσεις της εφαρμογής.

- **migrations:** Αυτός ο φάκελος αποθηκεύει τις μετεγκαταστάσεις και αλλαγές στη ΒΔ.

- **models.py:** Περιέχει τα μοντέλα της εφαρμογής.

- **tests.py:** Περιέχει εκτελέσιμες διαδικασίες σε περίπτωση δοκιμής της εφαρμογής.

- **views.py:** Περιέχει τη λογική που καθορίζει με ποιον τρόπο λειτουργεί η εφαρμογή, όπως την επεξεργασία δεδομένων και την ανταπόκριση στα αιτήματα των χρηστών.

Είναι απαραίτητο να δημιουργηθεί από τον προγραμματιστή ένα επιπλέον αρχείο για την εφαρμογή με τίτλο **urls.py**. Σε αυτό το αρχείο θα καταχωρηθούν τα URL μονοπάτια της εφαρμογής.

Για να ενημερωθεί το πρόγραμμα για την νέα εφαρμογή επιβάλλεται να συμπληρώσουμε το όνομα της εφαρμογής στη λίστα 'INSTALLED APPS' που βρίσκεται στο αρχείο settings.py.

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',

    'όνομα_εφαρμογής',
]
```

Είναι βασικό η εφαρμογή να εισάγεται στο τέλος της λίστας διότι η λίστα διαβάζεται σειριακά και ίσως η εφαρμογή να χρειάζεται λειτουργίες από τις εφαρμογές πυρήνα του Django.

Έπειτα για να γίνει η σύνδεση του URL του προγράμματος με τα URL της εφαρμογής στο αρχείο urls.py του ριζικού καταλόγου πρέπει να προσθέσουμε το άρθρωμα (module) '**include**' και στη λίστα **urlpatterns** να εισαχθεί το μονοπάτι για το url αρχείο της εφαρμογής. Η μορφή του urls.py αρχείου ύστερα από την εισαγωγή της εφαρμογής θα είναι:

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path(' όνομα_εφαρμογής/', include("όνομα_εφαρμογής.urls")),
]
```

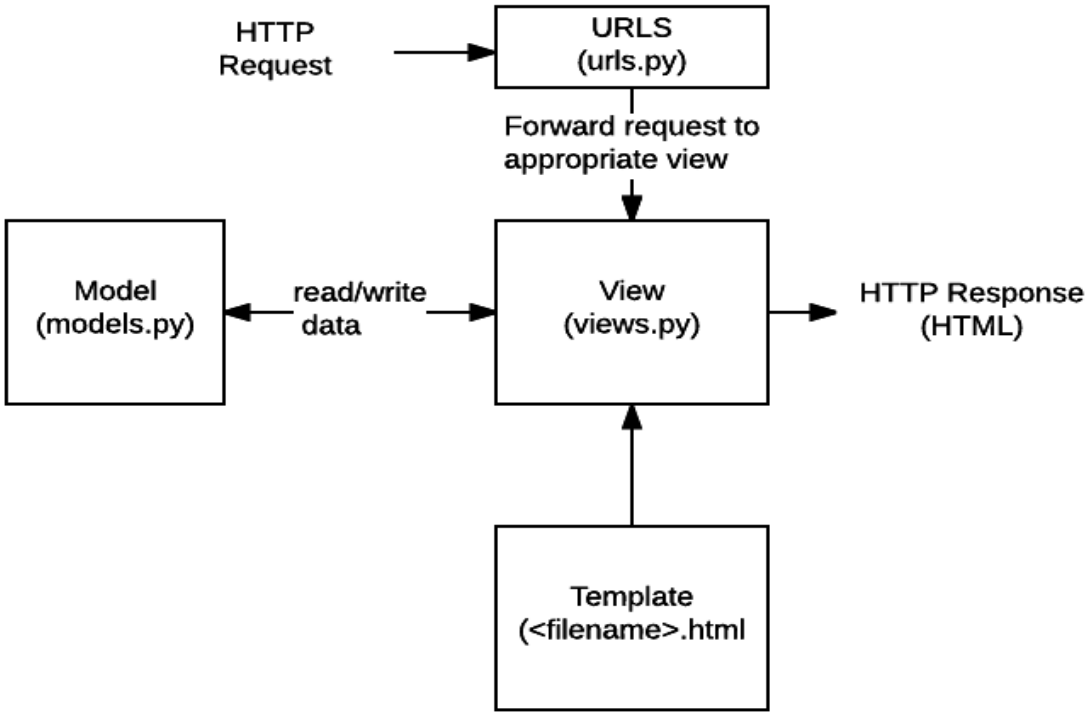
Με την ανάπτυξη της εφαρμογής πιθανό να είναι αναγκαία η δημιουργία νέων αρχείων που θα εξυπηρετούν μοναδικές διαδικασίες για την λειτουργικότητα της εφαρμογής.

Ο χρήστης στέλνει ένα HTTP/HTTPS αίτημα μέσω του browser στον web server. Ο web server θα παραδώσει το αίτημα στο WSGI και σαν μεσολαβητής προωθεί το αίτημα στο πρόγραμμα.

Το αίτημα έχει τη μορφή URL για αυτό αρχικά θα ελεγχθεί το αρχείο urls.py για να αποφασιστεί ποιο view θα χρησιμοποιεί από το αρχείο views.py.

Για την επεξεργασία του αιτήματος μπορεί να χρειαστεί να διαβαστούν/γραφτούν/διαγραφούν δεδομένα από την ΒΔ για αυτό ελέγχεται το αρχείο models.py. Υπάρχει και η περίπτωση να ικανοποιηθεί το αίτημα με μία HTTP απάντηση ορισμένη από το κατάλληλο view.

Το view στέλνει τα δεδομένα σε συγκεκριμένο πρότυπο που είναι αποθηκευμένο στον φάκελο templates. Το HTML πρότυπο με τα δεδομένα επιστρέφει ολοκληρωμένο πίσω στον browser του χρήστη.



Εικόνα 3.4 Τρόπος λειτουργίας του Django

(https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction)

Η λειτουργία του Django framework μπορεί να είναι πολύ πιο εξεζητημένη λόγω των τρομερών δυνατοτήτων που προσφέρει ωστόσο έγινε αναφορά στα βασικά σημεία διαχείρισης αιτήματος – ανταπόκρισης του Django.

Ενότητα 3.5 - Υλοποίηση Προγράμματος

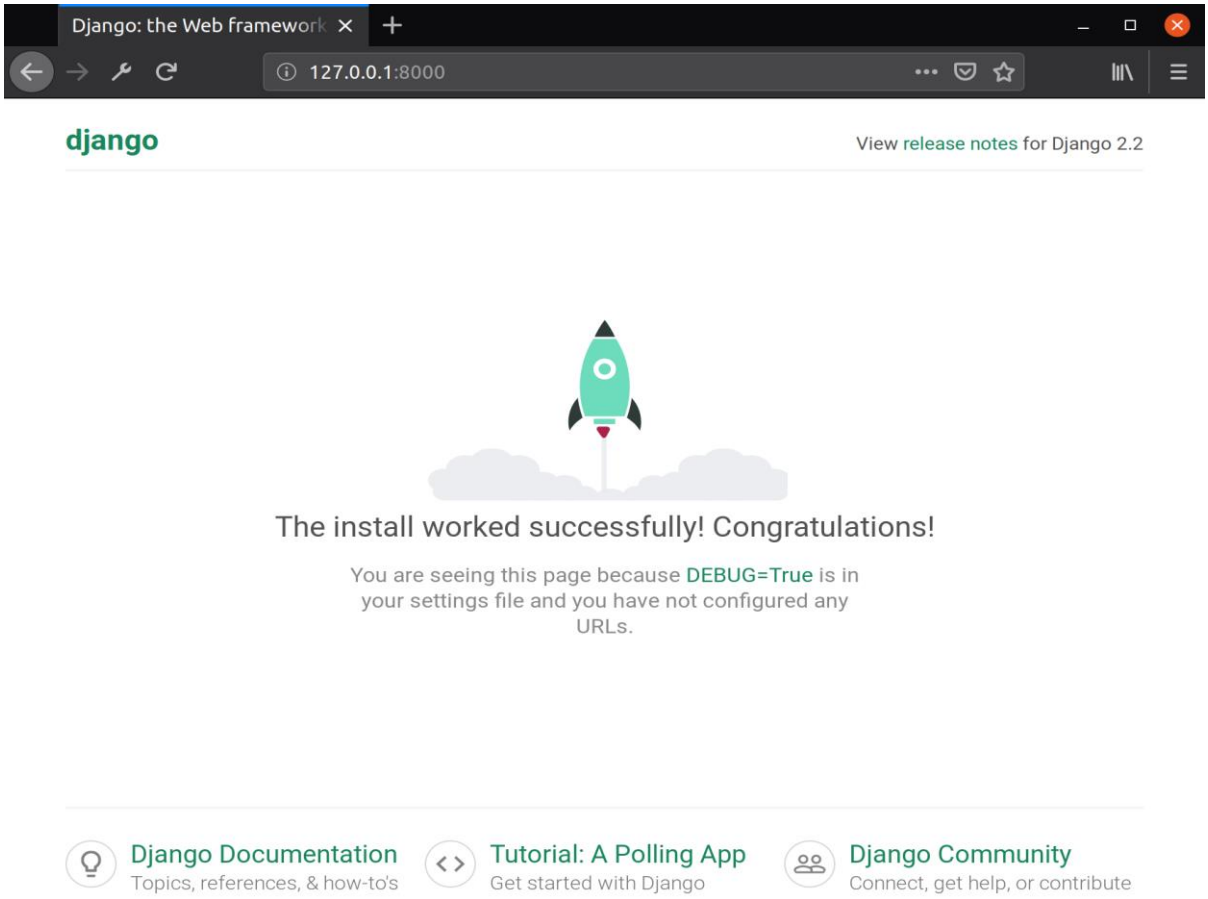
Όπως αναφέρθηκε και στην Ενότητα 3.1 μελετάμε την σχεδίαση ενός ιστότοπου κοινωνικής δικτύωσης.
Χρησιμοποιείται ένα venv (υποενότητα 3.1.2) με τα παρακάτω πακέτα.

asgiref
certifi
channels
charset-normalizer
Django
django-channels
idna
oauthlib
pillow
pip
psycopg2
requests
requests-oauthlib
six
sqlparse
tzdata
urllib3

Όπως ορίζει η αρχιτεκτονική του Django δημιουργούμε το πρόγραμμα με όνομα website και αυτομάτως παράγεται η κατάλληλη δομή. Για να σιγουρευτούμε ότι το πρόγραμμα δουλεύει θα εισάγουμε στην κονσόλα την εντολή

```
> python manage.py runserver
```

Εισάγοντας στο browser το URL 127.0.1:8000 θα μας προβληθεί η αρχική σελίδα του Django.



Υποενότητα 3.5.1 – Δομή Προγράμματος

Το Django γνωρίζουμε ότι δημιουργεί προκαθορισμένες ρυθμίσεις οι οποίες πρέπει να τροποποιηθούν ώστε να καλύψουν τις ανάγκες μας και να αποκτήσουν τη λειτουργικότητα που επιθυμούμε.

Το Django χρησιμοποιεί ως προκαθορισμένη ΒΔ την SQLite την οποία θα αλλάξουμε στη PostgreSQL. Για να το επιτύχουμε αυτό θα διαγράψουμε τα αρχεία της προηγούμενης ΒΔ και έπειτα θα τροποποιήσουμε το αρχείο **settings.py** στο σημείο που αναγράφει Database. Η νέα μορφή θα είναι:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql_psycopg2',
        'NAME': 'django-p',
        'USER': 'postgres',
        'PASSWORD': '123',
        'HOST': 'localhost',
        'PORT': '5432',
    }
}
```

Είναι σημαντικό να αναφερθεί ότι στο venv πρέπει να εγκατασταθεί πρώτα το πακέτο psycopg2 για να λειτουργήσει. Οι τιμές των μεταβλητών ορίζονται από τον προγραμματιστή.

Αφού έχει αναπτύχθηκε ο κορμός του προγράμματος θα δημιουργήσουμε την εφαρμογή μας με όνομα base. Ύστερα θα ενημερώσουμε το πρόγραμμα εισάγοντας την εφαρμογή στην λίστα.

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',

    'channels',
    'base.apps.BaseConfig',

]
```

Παρατηρείται ότι πριν το base έχουμε εισάγει την εφαρμογή **channels** (κανάλια).
Η εφαρμογή channels επεκτείνει τις δυνατότητες του Django επιτρέποντας το να διαχειριστεί και πρωτόκολλα διαφορετικά του HTTP όπως WebSockets, chat protocols και άλλα τα οποία απαιτούν μεγάλης διάρκειας συνδεσιμότητα. Πλέον το πρόγραμμα μας μπορεί να χρησιμοποιήσει και asgi. Θα χρειαστούμε τα channels για την επικοινωνία των χρηστών

που θα προσφέρει η ιστοσελίδα. Για την χρήση του πακέτου channels πρέπει να γίνει εγκατάσταση του στο venv.

Στο settings.py θα ενημερώσουμε το πρόγραμμα για την χρήση wsgi και asgi.

```
WSGI_APPLICATION = 'website.wsgi.application'
ASGI_APPLICATION = 'website.asgi.application'

CHANNEL_LAYERS = {
    'default': {
        'BACKEND': 'channels.layers.InMemoryChannelLayer'
    }
}
```

Επίσης το πρόγραμμα θα δημιουργήσουμε ένα νέο αρχείο με όνομα asgi.py με περιεχόμενο

```
asgi.py > ...
import os

from django.core.asgi import get_asgi_application
from channels.auth import AuthMiddlewareStack
from channels.routing import ProtocolTypeRouter, URLRouter

import base.routing

os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'website.settings')

application = ProtocolTypeRouter({
    "http": get_asgi_application(),
    "websocket": AuthMiddlewareStack(
        URLRouter(
            base.routing.websocket_urlpatterns
        )
    )
})
```

Ακολουθεί η τροποποίηση των static αρχείων. Στον φάκελο static θα δημιουργήσουμε άλλους τρεις φακέλους με τα ονόματα images, js και style.

Όπως υποδεικνύουν και τα ονόματα τους στους συγκεκριμένους φακέλους θα αποθηκεύονται αρχεία εικόνων, αρχεία JavaScript και αρχεία CSS. Ως JavaScript αρχείο έχω ένα πρόγραμμα που προβάλλει την ημερομηνία και την ώρα. Στο φάκελο CSS υπάρχει το αρχείο style.css που περιέχει στοιχεία σχεδίασης των ιστοσελίδων. Τα δεδομένα αυτά θα είναι διαθέσιμα σε όλο το πρόγραμμα. Για αυτό θα αλλάξουμε το settings.py στο σημείο του static files για να γνωρίζει το πρόγραμμα που να βρει τα αρχεία.

```
STATIC_URL = 'static/'
MEDIA_URL = '/images/'

STATICFILES_DIRS = [
    BASE_DIR / 'static',
]

MEDIA_ROOT = BASE_DIR / 'static/images'
```

Ακολουθεί η μετατροπή του προτύπου (template).

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [
            BASE_DIR / 'templates'
        ],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]
```

Επιπρόσθετα στο φάκελο templates του προγράμματος θα δημιουργήσουμε ένα HTML αρχείο με την βασική δομή και χαρακτηριστικά κάθε ιστοσελίδας του προγράμματος. Επειδή κάθε ιστοσελίδα θα έχει και μία μπάρα πλοήγησης θα γραφτεί ο κώδικας της σε ένα άλλο HTML αρχείο το οποίο θα εμπεριέχετε στο αρχείο της βασικής δομής.

Στο αρχείο urls.py θα προβούμε σε αλλαγές ώστε να ορίσουμε την κύρια αρχική σελίδα της ιστοσελίδας, να καταγράψουμε το μονοπάτι για τα url αρχεία της εφαρμογής base και να υποδείξουμε που βρίσκονται τα static αρχεία του προγράμματος.

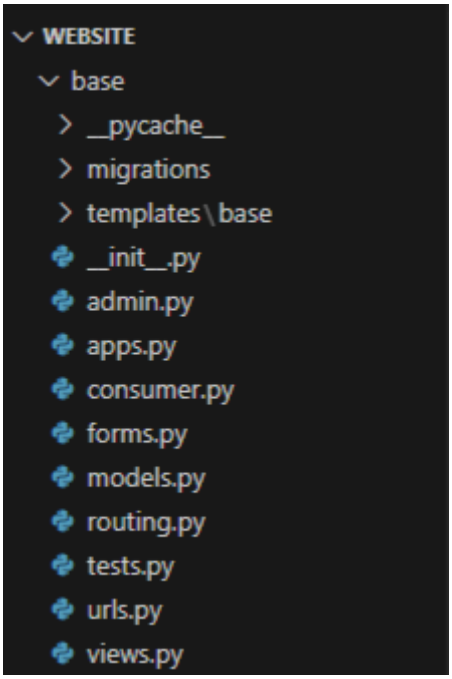
```python
e > urls.py > ...
from django.contrib import admin
from django.urls import path , include
from django.conf.urls.static import static
from django.views.generic.base import TemplateView
from django.conf import settings

urlpatterns = [
    path('admin/', admin.site.urls),
    path('',include('base.urls')),
    path('', TemplateView.as_view(template_name="home.html"), name="home"),
] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

Με αυτές τις αλλαγές σχεδιάσαμε τις ρυθμίσεις του προγράμματος για να μπορέσει να υλοποιηθεί η ιστοσελίδα.

Υποενότητα 3.5.2 – Δομή Εφαρμογής

Η εφαρμογή μας ονομάζεται base και έχει την παρακάτω δομή:



Ας αναλύσουμε τα αρχεία από τα οποία αποτελείται η εφαρμογή

- **Templates**

  Στον φάκελο templates\base έχουν αποθηκευτεί όλα τα HTML αρχεία που σχετίζονται με την εφαρμογή. Όλα τα αρχεία έχουν στοιχεία από τα αρχεία HTML του κύριου προγράμματος (Website).

- **__init__.py , apps.py , tests.py**

  Έχει γίνει αναφορά στην υποενότητα 3.3.3 για την χρησιμότητα και την λειτουργία τους

- **models.py**

  Δημιουργία μοντέλων που περιέχουν τα βασικά πεδία και τις συμπεριφορές των δεδομένων που αποθηκεύονται.

  Το μοντέλο **User** απαρτίζεται από συνολικά οκτώ πεδία (username,email,bio,avatar,created,friends,USERNAME_FIELD,REQUIRED_FIELDS) και από δύο συναρτήσεις που εξυπηρετούν συγκεκριμένη λειτουργία για το μοντέλο.

  Το μοντέλο **Role** απαρτίζεται από συνολικά ένα πεδία (name) και μία συνάρτηση.

  Το μοντέλο **Article** απαρτίζεται από συνολικά οκτώ πεδία (user,title,type,description,updated,created,likes,dislikes), από μία κλάση η οποία καθορίζει ότι πρώτα θα προβάλλονται τα νέα άρθρα και δύο συγκεκριμένες συναρτήσεις.

  Το μοντέλο **Comment** απαρτίζεται από συνολικά πέντε πεδία (user,article,body,updated,created), την ίδια κλάση με το Article και δύο συναρτήσεις.

Το μοντέλο **ChatRoom** απαρτίζεται από συνολικά έξι πεδία (host,name,slug,members,updated,created), την ίδια κλάση με το Article και μία συνάρτηση.

Το μοντέλο **Message** απαρτίζεται από συνολικά τέσσερα πεδία (chatroom,user,content,created), μία κλάση και μια συνάρτηση που ορίζει ότι θα είναι εμφανή οι πρώτοι 30 χαρακτήρες του μηνύματος.

Ορισμένα μοντέλα σχηματίζουν σχέσεις μεταξύ τους που σημαίνει ότι τα πεδία ενός μοντέλου περιέχουν τιμές που αναφέρονται σε άλλο μοντέλο. Υπάρχουν δυο ειδών σχέσεις.

- **Πολλά σε Ένα (Many-to-One)**

  Για να οριστεί το είδος σχέσης ενός πεδίου με κάποιο άλλο μοντέλο χρησιμοποιείται το models.ForeignKey. Για παράδειγμα το πεδίο user του μοντέλου Article είναι models.ForeignKey με το μοντέλο User. Αυτό σημαίνει ότι πολλά άρθρα μπορούν να δημοσιευτούν από έναν και μοναδικό χρήστη.

- **Πολλά σε Πολλά (Many-to-Many)**

  Για να οριστεί το είδος σχέσης ενός πεδίου με κάποιο άλλο μοντέλο χρησιμοποιείται το models.ManyToManyField. Για παράδειγμα το πεδίο members του μοντέλου ChatRoom συσχετίζεται με το μοντέλο User. Άρα σε πολλά δωμάτια επικοινωνίας μπορούν να είναι μέλη πολλοί χρήστες

- **admin.py**

Είναι καταχωρημένα τα μοντέλα User, Role, Article, Comment, Chatroom, Me στη διεπαφή του διαχειριστή. Είναι ορατά στον διαχειριστή μέσω της σελίδας διεπαφής που παρέχει το django.

- **urls.py**

  Είναι αποθηκευμένα μονοπάτια για ενέργειες ή πόρους για την συγκεκριμένη εφαρμογή.

- **views.py**

  Στο αρχείο όπου γίνεται η λήψη των αιτημάτων από τα urls μονοπάτια και ύστερα από επεξεργασία επιστρέφει την κατάλληλη απάντηση ή την αδυναμία να εξυπηρετήσει το αίτημα.

- **forms.py**

  Οι forms είναι χρήσιμες στην λήψη δεδομένων απευθείας από τον χρήστη με συγκεκριμένο τρόπο και να πραγματοποιεί ενέργειες στη ΒΔ. Μία form της εφαρμογής είναι η LoginForm η οποία χρησιμοποιείται για την σύνδεση (login) των χρηστών. Τα δεδομένα που καλείται να εισάγει ο χρήστης είναι το όνομα χρήστη και τον κωδικό πρόσβασης που του αναλογεί. Η LoginForm παρουσιάζεται στον χρήστη όταν αιτείται να συνδεθεί στην ιστοσελίδα. Όπως καθορίζει ο τρόπος λειτουργίας του django το αίτημα μεταβιβάζεται στη κατάλληλη view που ελέγχει αν τα δεδομένα που έδωσε ο χρήστης είναι αληθή. Στην περίπτωση που δεν αντιστοιχούν στα δεδομένα της ΒΔ η view επιστρέφει μήνυμα σφάλματος. Στην περίπτωση που είναι τα σωστά δεδομένα επιστρέφει την αρχική σελίδα με τον χρήστη συνδεδεμένο.

Το forms.py αρχείο δεν δημιουργείται αυτόματα κατά δημιουργία της εφαρμογής. Είναι στην ευχέρεια του χρήστη σε περίπτωση που επιθυμεί να χρησιμοποιήσει αυτό το χαρακτηριστικό της django.

- **consumer.py**

  Στην υποενότητα 3.5.1 αναφέρθηκε η χρήση και η λειτουργία του πακέτου channels. Αφού έγιναν οι κατάλληλες διεργασίες στο κύριο πρόγραμμα πρέπει να τροποποιηθεί και η εφαρμογή. Ο consumer (καταναλωτής) αποτελεί βασική οντότητα για το πακέτο. Περιέχει τον κώδικα ASGI εφαρμογών που ικανοποιούν συγκεκριμένες διεργασίες. Σε αντίθεση με τις views που ενεργοποιούνται όποτε λάβουν κάποιο αίτημα, οι consumers είναι ενεργοποιημένοι για μεγάλο χρονικό διαστήματα αναμένοντας αιτήματα σύνδεσης.

  Ένα django πρόγραμμα μπορεί να διαθέτει πολλούς consumers με τον καθένα να έχει το ατομικό του σύνολο οδηγιών και πληροφοριών για τις εισερχόμενες συνδέσεις.

- **routing.py**

  Το αρχείο routing.py χρησιμοποιείται για τον συνδυασμό των consumers. Είναι παρόμοιο με τα django URLs.

ΚΕΦΑΛΑΙΟ 4 – Μέτρα Ασφάλειας

Ενότητα 4.1 - Ασφάλεια στα Django προγράμματα

Στις μέρες μας αναπτύσσονται διαδικτυακές εφαρμογές με ραγδαίο ρυθμό. Οι προγραμματιστές που τις κατασκευάζουν επιλέγουν το Django Framework για την παράδοση εφαρμογών που θα εξυπηρετούν αποτελεσματικά τις ανάγκες των χρηστών και ικανοποιούν τις απαιτήσεις του πελάτη. Στην υποενότητα 3.2.3 αναφέρθηκε ότι ένας σημαντικότατος παράγοντας που ωθεί τους προγραμματιστές να εμπιστεύονται το django για τα έργα που αναλαμβάνουν είναι η ασφάλεια που παρέχει στο πρόγραμμα και στα δεδομένα των χρηστών.

Για να γίνει αντιληπτό πόσο διαδεδομένο είναι το συγκεκριμένο framework ακολουθεί μία λίστα με τις πιο γνωστές και με μεγάλη επισκεψιμότητα ιστοτόπους που έχουν υλοποιηθεί εξ ολοκλήρου ή μερικά τους κομμάτια με τη χρήση django.

- **Mozilla Firefox**
- **YouTube**
- **Instagram**
- **Dropbox**

Όλοι οι παραπάνω ιστότοποι διαχειρίζονται όγκους δεδομένων τόσο για την λειτουργία τους όσο και τα προσωπικά δεδομένα των ατόμων που επιλέγουν να χρησιμοποιήσουν την

εφαρμογή τους. Είναι ευκόλως αυτονόητο ότι επιβάλλεται να δοθεί έμφαση στην προστασία των δεδομένων από πιθανές κλοπές και αλλοιώσεις.

Το Django παρέχει προκαθορισμένους μεθόδους και εργαλεία ασφάλειας στο πρόγραμμα που ξεκινάει η ανάπτυξη του.

Αρχικά προκαθορίζει στο αρχείο settings.py ένα SECRET_KEY το οποίο προσδίδει κρυπτογραφημένη υπογραφή. Χρειάζεται επίσης και για την κρυπτογράφηση των cookies. Είναι ζήτημα καίριας σημασίας το SECRET_KEY να παραμείνω προστατευμένο από κυβερνοεγκληματίες.

Επίσης στο ίδιο αρχείο παρατηρείται ότι στις λίστες INSTALLED_APPS και MIDDLEWARE περιλαμβάνονται django πακέτα τα οποία στοχεύουν την σωστή αυθεντικοποίηση των στοιχείων, την κρυπτογράφηση δεδομένων και άλλων λειτουργιών.

Το προκαθορισμένο σύστημα αποθήκευσης κωδικών χρησιμοποιεί την κρυπτογράφηση PBKDF2 (Password-Based Key Derivation Function 2) που είναι αλγόριθμος κρυπτογράφησης που παράγει κρυφά κλειδιά από μία μυστική τιμή, κωδικό ή φράση με την χρήση μίας ψευδοτυχαίας συνάρτησης. Η τεχνική αυτή παράγει τυχαία και μεγάλα σε αριθμό και ποικιλία χαρακτήρων αλφαριθμητικά ως κωδικούς μειώνοντας δραστικά ωμής βίας επιθέσεις. Αυτού του είδους επιθέσεις προσπαθούν να βρουν τον κωδικό δοκιμάζοντας πολλούς κωδικούς.

Το σύστημα πιστοποίησης ενημερώνει και προτρέπει τους χρήστης να δημιουργούν μεγάλους και δυνατούς κωδικούς. Κατά τη διάρκεια της εισαγωγής των διαπιστευτηρίων αν γίνει λάθος το πρόγραμμα ενημερώνει για το λάθος χωρίς να δίνει παραπάνω πληροφορίες.

Ενότητα 4.2 - Τυπικές ευπάθειες στο πρόγραμμα

Το django προσφέρει πολλές δυνατότητες προστασίας και ασφάλειας δεδομένων ωστόσο δεν είναι και τελείως ασφαλές. Είναι αδύνατο να αναπτυχθεί ένα πρόγραμμα ή ένα πληροφοριακό σύστημα το οποίο να είναι ανεπηρέαστο σε επιθέσεις.

Στην ενότητα 2.6 αναφέρθηκε ότι το μεγαλύτερο ποσοστό που εκθέτει την ασφάλεια ενός προγράμματος είναι ο ανθρώπινος παράγοντας. Ο διαχειριστής ενός προγράμματος οφείλει να είναι εξαιρετικά προσεκτικός με τους κωδικούς που χρησιμοποιεί. Οφείλει να χρησιμοποιεί δυνατούς κωδικούς, μη προβλέψιμους, να μην τους μοιράζεται με κανέναν άλλον. Εννοείται το SECRET_KEY και ο κωδικός πρόσβασης στη ΒΔ πρέπει να παραμείνουν κρυφοί με κάθε δυνατό τρόπο. Επίσης σαν διαχειριστής συστήματος είναι αναγκαίο να είναι απόλυτα ενημερωμένος σε θέματα ασφάλειας πληροφοριών και να εφιστά αμέριστη προσοχή να μην πέσει θύμα phishing(υποενότητα 2.3.1), που μπορεί να οδηγήσει σε κλοπή του κωδικού του ή σε Trojan Horse επίθεση που θα εκθέσει τον πηγαίο κώδικα και την ΒΔ του προγράμματος.

Επιπλέον ευπάθειες προκαλούνται και από εσφαλμένες ρυθμίσεις και παραλείψεις του προγραμματιστή ή διαχειριστή στο πρόγραμμα.

```
DEBUG = True

ALLOWED_HOSTS = []
```

Η μέθοδος DEBUG αν είναι ενεργοποιημένη, σε περίπτωση που εγείρει μια εξαίρεση το πρόγραμμα θα παρουσιάσει λεπτομερώς τις αιτίες που οδήγησαν στο σφάλμα διαρρέοντας σημαντικές πληροφορίες για τις ρυθμίσεις του προγράμματος. Πληροφορίες που σε άτομο που γνωρίζει θα είναι αρκετά χρήσιμες για να κατανοήσει το πρόγραμμα.

Η λίστα ALLOWED_HOSTS αν το πρόγραμμα είναι στο διαδίκτυο θα ήταν ALLOWED_HOSTS = [*] το οποίο καταδεικνύει ότι το πρόγραμμα μπορεί να εξυπηρετήσει κάθε οικοδεσπότη (host). Αυτό είναι απίστευτα επίφοβο επειδή θα επιτρέπει εισερχόμενες επικοινωνίες από εξωτερικούς χρήστες εκθέτοντας το πρόγραμμα σε πιθανές επιθέσεις.

Το django στις προκαθορισμένες ρυθμίσεις του χρησιμοποιεί ως πρωτόκολλο επικοινωνίας το HTTP (ενότητα 1.3). Είναι γνωστό ότι σε συστήματα που μεταδίδονται προσωπικά δεδομένα το πρωτόκολλο HTTP πρέπει να αποφεύγεται καθώς προβάλλει ανοιχτά τα δεδομένα.

Το πρόγραμμα είναι ευάλωτο σε επιθέσεις SQL injection (υποενότητα 2.3.1) εξαιτίας της ΒΔ που διαθέτει.

Είναι αρκετά πιθανό και επιθέσεις τύπου XSS (υποενότητα 2.3.1) να κατορθώσουν να εισχωρήσουν κακόβουλα scripts στη ΒΔ για τους ανυποψίαστους χρήστες της ιστοσελίδας.

Τέλος το URL της διεπαφής του διαχειριστή της ιστοσελίδας είναι το προκαθορισμένο, πληροφορία γνωστή σε κυβερνοεγκληματίες οι οποίοι ίσως να προσπαθήσουν να μαντέψουν τα διαπιστευτήρια αποκτώντας πρόσβαση διαχειριστή και δυνατότητα να δουν και να παραποιήσουν δεδομένα της ΒΔ.

Ενότητα 4.3 - Αντιμετώπιση των τυπικών ευπαθειών του προγράμματος

Στην προηγούμενη ενότητα διατυπώθηκαν εμφανή και πιθανές ευπάθειες που θα ήταν ικανές να προσβάλλουν την ασφάλεια του προγράμματος. Σε αυτή την ενότητα θα παρουσιαστούν οι τρόποι αντιμετώπισης των ευπαθειών για την προστασία των δεδομένων των χρηστών και του προγράμματος.

Αρχικά το SECRET_KEY πρέπει να αλλαχθεί και να πάρει τη θέση του ένα ισχυρό κλειδί δημιουργημένο από μία συνάρτηση παραγωγής τυχαίων κλειδιών όπως η **get_random_key()** συνάρτηση του django. Επίσης για να διατηρηθεί μυστικό θα ήταν καλή κίνηση να φορτώνεται από μια μεταβλητή περιβάλλοντος.

Στο αρχείο settings.py θα γίνει εισαγωγή της βιβλιοθήκης os και αλλαγή της τιμής του SECRET_KEY.

```
import os

SECRET_KEY = os.environ["SECRET_KEY"]
```

Επιπλέον θα αλλάξουν και οι μεταβλητές:

```
DEBUG = False

ALLOWED_HOSTS = ['website.com',]
```

Με αυτό τον τρόπο η ευπάθεια του DEBUG εξαλείφεται και η μοναδική διεύθυνση από την οποία μπορεί η ιστοσελίδα να προσπελαστεί είναι το Domain Name.

Το πρόγραμμα αποθηκεύει στη ΒΔ πολλές ευαίσθητες πληροφορίες όπως κωδικούς πρόσβασης, ηλεκτρονικές διευθύνσεις και μηνύματα για είναι απολύτως απαραίτητη η χρήση πρωτοκόλλου HTTPS για την ασφαλή μεταφορά των δεδομένων. Η αλλαγή των πρωτοκόλλων επιτυγχάνεται με την εισαγωγή του αρθρώματος **django.middleware.security.SecurityMiddleware** στη λίστα του MIDDLEWARE και την εισαγωγή της μεταβλητής

SECURE_SSL_REDIRECT = True στο αρχείο settings.py.

Η προστασία από SQL injections πρέπει να ληφθεί πολύ σοβαρά. Ευτυχώς το επίπεδο ORM (υποενότητα 3.2.3) δρα ως μερική προστασία έναντι τέτοιων επιθέσεων. Το ORM μετατρέπει τα δεδομένα από την ΒΔ σε αντικείμενα (objects) κατά την μεταφορά τους. Επίσης τα querysets του django είναι προστατευμένα εξαιτίας της παραμετροποίησης των δηλώσεων

(queries). Τα queries είναι ερωτήματα ή αιτήσεις πληροφοριών προς την ΒΔ. Τα querysets αντιπροσωπεύουν μια συλλογή αντικειμένων από την ΒΔ. Με αυτό τον τρόπο η ενσωμάτωση SQL εντολών δεν μπορεί εύκολα να συμβεί.

Δεδομένα που υποβάλλονται από τους χρήστες δεν πρέπει να εισάγονται απευθείας στις ιστοσελίδες διότι μπορεί να προέρχονται από κακόβουλους χρήστες οι οποίοι προσπαθούν μια XSS επίθεση (υποενότητα 2.3.1). Για την αποτροπή τέτοιων επιθέσεων προτείνεται είτε να ελέγχεται κάθε αναξιόπιστη μεταβλητή από το φίλτρο **escape** είτε να γίνει χρήση του django automatic HTML escaping. Τα πρότυπα (templates) του django προβάλλουν αρκετή αντίσταση έναντι της πλειοψηφίας των XSS επιθέσεων. Μεγάλη προσοχή στην αποθήκευση HTML στη ΒΔ ειδικά αν αυτή η HTML προβάλλεται.

Ως τελική συμβουλή για την προστασία του προγράμματος δηλώνεται ο μετασχηματισμός του URL που οδηγεί στη διεπαφή του διαχειριστή. Για την υλοποίηση αυτού του σκοπού στο αρχείο urls.py του κυρίως προγράμματος θα αλλαχθεί το μονοπάτι στη λίστα urlpatterns από admin/ σε κάτι πιο περίπλοκο, πχ

this-is-admin-page/.

Οι παραπάνω ήταν τρόποι για την χαρακτηριστική βελτίωση της προστασίας δεδομένων και ασφάλειας του προγράμματος. Είναι βέβαιο πως δεν θα θωρακίσουν σε απόλυτο βαθμό το πρόγραμμα αλλά είναι σίγουρη η αντιμετώπιση της πλειοψηφίας των επιθέσεων.

ΚΕΦΑΛΑΙΟ 5 - Συμπεράσματα

Το Διαδίκτυο θα συνεχίσει να επεκτείνεται για πολλές δεκαετίες ακόμα φιλοξενώντας όλο και περισσότερους ανθρώπους και πληροφοριακά συστήματα. Κάθε άτομο ξεχωριστά θα συνεχίσει να παραδίδει μεγάλο όγκο ευαίσθητων προσωπικών δεδομένων σε κυβερνήσεις και επιχειρήσεις με την πίστη ότι αυτά θα επεξεργαστούν κατάλληλα και θα παραμείνουν ασφαλή. Η ιδιωτικότητα του ατόμου στον κυβερνοχώρο θα βρίσκεται σε μόνιμο κίνδυνο από επιθέσεις οι οποίες θα στοχεύουν διαδικτυακές εφαρμογές για την διαρροή ή μετατροπή των δεδομένων του. Αποτελεί μείζον ζήτημα η προστασία των προσωπικών δεδομένων. Για να επιτευχθεί όσο το δυνατόν περισσότερο αυτός ο στόχος οι ήδη υπάρχοντες και μελλοντικές ιστοσελίδες οφείλουν να δώσουν έμφαση στον κατάλληλο σχεδιασμό που σαν κύριο γνώμονα θα έχει την ασφάλεια του προγράμματος και την αντιμετώπιση αυτού σε επιθέσεις. Οι γνωστές ευπάθειες θα πρέπει να σταματήσουν να προκαλούν προβληματισμούς στους προγραμματιστές και νέοι τρόποι προστασίας να δημιουργηθούν. Αυτές οι ενέργειες θα διαμορφώσουν έναν ασφαλή ψηφιακό κόσμο όπου κάθε άνθρωπος θα είναι βέβαιος ότι αυτός επιλέγει που και ποια δεδομένα παραχωρεί χωρίς καμία αμφισβήτηση για την ασφάλεια τους. Το δικαίωμα στην ιδιωτικότητα θα έχει σταθερές βάσεις και δεν θα προσβάλλεται από κανέναν.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Ανάκτηση από https://csrc.nist.gov/glossary/term/cyberspace

[2] Ανάκτηση από https://www.weforum.org/agenda/2024/01/cybersecurity-cybercrime-system-safety/

[3] Ανάκτηση από https://csrc.nist.gov/glossary/term/cyber_attack

[4] Ανάκτηση από https://csrc.nist.gov/glossary/term/cyber_security

[5] Ανάκτηση από https://csrc.nist.gov/glossary/term/vulnerability

[6] Ανάκτηση από https://www.statista.com/chart/28878/expected-cost-of-cybercrime-until-2027/

[7] Ανάκτηση από https://en.wikipedia.org/wiki/The_Right_to_Privacy_(article)

[8] Ανάκτηση από https://commission.europa.eu/law/law-topic/data-protection/reform/what-personal-data_en

[9] Ανάκτηση από https://commission.europa.eu/law/law-topic/data-protection/reform/rights-citizens/my-rights/can-i-ask-company-delete-my-personal-data_en

[10] Ανάκτηση από https://www.python.org/

[11] Ανάκτηση από https://www.djangoproject.com/

[12] Ανάκτηση από https://el.linux-console.net/?p=538#gsc.tab=0

[13] Ανάκτηση από https://www.postgresql.org/