ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

<<Ασφάλεια εφαρμογών CMS>>

Της φοιτήτριας
Αλεξίου Μαρία Ραφαηλία
Αρ. Μητρώου: 134154

Τίτλος Δ.Ε: Ασφάλεια εφαρμογών CMS

Κωδικός: Δ.Ε 22229

Ονοματεπώνυμο φοιτητή: ΑΛΕΞΙΟΥ ΜΑΡΙΑ ΡΑΦΑΗΛΙΑ.

Ονοματεπώνυμο εισηγητή: Παναγιώτης Τζέκης

Ημερομηνία ανάληψης Δ.Ε: 13 Μαΐου 2022

Ημερομηνία περάτωσης Δ.Ε. …

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραιτήτως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

Περιεχόμενα

ΠΡΟΛΟΓΟΣ

Τα συστήματα διαχείρισης περιεχομένου Ιστού (WCMS) είναι συστήματα που χρησιμοποιούνται στη δημιουργία, δημοσίευση, προσαρμογή και σχεδιασμό υπηρεσιών ιστοτόπων από διαχειριστές Ιστού για την παροχή διαδικτυακών εφαρμογών με επίκεντρο τον χρήστη και Υπηρεσίες. Τέτοιες εφαρμογές περιλαμβάνουν το Joomla, το Drupal και το WordPress, που έχουν βρει τη χρήση τους διάφορα ιδρύματα, συμπεριλαμβανομένων πανεπιστημίων και κολλεγίων, μη κυβερνητικών και κυβερνητικών ιδρυμάτων. Ενώ αυτά τα WCMS παρέχουν στους χρήστες εύκολη πρόσβαση σε υπηρεσίες Ιστού, είναι ευάλωτα στην ασφάλεια, τις παραβιάσεις και απειλές. Ο στόχος αυτής της εργασίας ειναι να προσδιορίσει το ευρέως χρησιμοποιούμενο WCMS και το επίπεδο παραβιάσεων ασφαλείας σε αυτές τις εφαρμογές από διαχειριστές ιστού. Θα παρουσιάσουμε λοιπόν μερικά από τα πιο γνωστά συστήματα καθώς και πιθανές ευπάθειες τους. Θα αναλύσουμε τρόπους αντιμετώπισης και θα παραθέσουμε παραδείγματα επιθέσεων.

ΠΕΡΙΛΗΨΗ

Τα συστήματα διαχείρισης περιεχομένου γίνονται όλο και πιο δημοφιλή εξαιτίας της ταχύτητας και της εύκολης υλοποίησης ιστοσελίδων χωρίς προγραμματιστικές γνώσεις που παρέχουν. Η ραγδαία αυτή αύξηση είναι ο λόγος που οι ιστότοποι που αναπτύχθηκαν με ενα cms αποτελούν στόχο για τους εγκληματίες του κυβερνοχώρου. Τα CMS χωρίζονται σε 2 κατηγορίες: δωρεάν ανοιχτού κώδικα και επί πληρωμή. Οι κατάλληλες ενέργειες διαχείρισης μπορούν να βελτιώσουν σημαντικά την αποτελεσματικότητα ενός διαδραστικού συστήματος από τη σκοπιά του χρήστη. Η πιο κοινή τύποι εγκλήματος στον κυβερνοχώρο είναι η έκθεση ευαίσθητων δεδομένων. Σήμερα, οι διαδικτυακές εφαρμογές χρησιμοποιούνται σε διάφορους τύπους βιομηχανιων όπως οι τράπεζες, το εμπόριο, η εκπαίδευση, η ιατρική, το μάρκετινγκ κ.λπ. Μία από τις απλούστερες μεθόδους για να αποκτήσουν οι χάκερ πρόσβαση σε σημαντικά δεδομένα παραδείγματος χάριν μεγάλων εταιρειών είναι η εύρεση μιας ευπάθειας σε μια εφαρμογή ώστε τελικά να καταφέρουν να παραβιάσουν το σύστημα. Τα επιπρόσθετα και τα θέματα που χρησιμοποιούν τα συστήματα αυτά μπορούν να προκαλέσουν μεγάλα προβλήματα ασφάλειας.

Οι ευπάθειες των συστημάτων αυτών απαραίτητο είναι να γίνονται γνωστές αφού έχουν αντιμετωπιστεί καθώς κακόβουλοι χρήστες μπορούν να τις εκμεταλλευτούν και τελικά να καταφέρουν να εισβάλουν στο σύστημα και να δημιουργήσουν πρόβλημα σε πολλές IP.

ABSTRACT

Content management systems are becoming increasingly popular because of the speed and ease of implementing websites without programming knowledge that they provide. This rapid growth is why sites developed with a cms are a target for cybercriminals. CMS are divided into 2 categories: free open source and paid. Appropriate management actions can greatly improve the effectiveness of an interactive system from the user's perspective. The most common types of cybercrime is the exposure of sensitive data. Today, online applications are used in various types of industries such as banking, commerce, education, medicine, marketing, etc. One of the simplest methods for hackers to gain access to important data of for example large companies is to find a vulnerability in an application so that they can eventually breach the system. Add-ons and themes that use these systems can cause major security issues.

The vulnerabilities of these systems must be known after they have been addressed as malicious users can take advantage of them and eventually manage to break into the system and create a problem for many IPs.

1. Εισαγωγή

Στα τέλη του 2019, εμφανίστηκε για πρώτη φορά ο κορονaϊός, ο κορονaϊός της Ουχάν (γνωστός επίσημα ως SARS-CoV-2), ο οποίος και προκάλεσε την πανδημία αυτή. Η μετάδοση του ιού αυτού γίνεται μέσω των σταγονιδίων μεταξύ των ανθρώπων. Αποτέλεσμα της πανδημίας αυτής δεν αποτελεί μόνο ο θάνατος χιλιάδων ανθρώπων ανά τον κόσμο αλλά και τη έναρξη μιας καινούργιας εποχής στον τεχνολογικό τομέα. Η τηλεκπαίδευση, οι ηλεκτρονικές πληρωμές, η ηλεκτρονική αγορά καθώς και η ικανοποίηση αναγκών του δημοσίου αποτελούν μερικά παραδείγματα.

Στις μέρες μας λοιπόν ένας ιστότοπος αποτελεί ουσιαστικό μέρος μιας επιχείρησης για την ενίσχυση της. Αδιαμφισβήτητα η ανάπτυξη ενός δικτυακού τόπου σήμαινε τη χρήση γλωσσών προγραμματισμού. Ωστόσο, η παράδοση έχει αλλάξει μετά την εφεύρεση των Συστημάτων Διαχείρισης Περιεχομένου (CMS). Με την χρήση αυτών η δημιουργία, επεξεργασία και ενημέρωση περιεχομένου ενός ιστότοπου πραγματοποιείτε εύκολα χωρίς να υπάρχουν γνώσεις προγραμματισμού.

Με την πάροδο των ετών και με την εξέλιξη της τεχνολογίας λοιπόν, οι δυνατότητες των web browsers και των σελίδων που μπορούσαν να υποστηρίξουν αναπτύχθηκαν ραγδαία. Οι απλές HTML σελίδες μετατράπηκαν σε δυναμικές ιστοσελίδες δίνοντας στους χρήστες την δυνατότητα να αλληλεπιδρούν τακτικά έχοντας πρόσβαση σε συγκεκριμένες πληροφορίες από τη βάση δεδομένων.

Η αύξηση των χρηστών και του διαδικτύου όμως έφερε στην επιφάνεια και ζητήματα ασφάλειας που επίσης αυξάνονται ταχύτατα. Όταν μια εταιρεία τελικά επιλέγει ένα CMS για την υποστήριξη διαδικτυακών συναλλαγών, δεν σκέφτεται τις πιθανές παραβιάσεις που μπορεί να προκληθούν από χάκερ, όπως παραδείγματος χάριν μπορεί να είναι η κλοπή πιστωτικής κάρτας ή και προσωπικά στοιχεία ταυτοποίησης. Η πιθανότητα αυτή μπορεί να προκαλέσει σημαντικά και κοστοβόρα προβλήματα στις επιχειρήσεις.

Τα CMS συστήματα σε μεγάλο βαθμό χρησιμοποιούν έτοιμα θέματα και επιπρόσθετα τα οποία μπορούν να προκαλέσουν σημαντικές ευπάθειες στο σύστημα προκαλώντας μεγάλες ανησυχίες.

Στο παρελθόν οι κακόβουλοι χρήστες έβαζαν μόνο έναν συγκεκριμένο στόχο, όπως για παράδειγμα ένα ακαδημαϊκό ίδρυμα, τράπεζα ή ηλεκτρονικό εμπόριο, όπου έβρισκαν μια ευπάθεια στο σύστημα και στη συνέχεια εκμεταλλευόμενοι την ευπάθεια αυτή έκλεβαν δεδομένα. Η διαδικασία αυτή ήταν χρονοβόρα και δύσκολη. Σήμερα όμως τους παρουσιάζονται μεγάλες ευκαιρίες απο τα CMS. Αντί λοιπόν να προσδιορίσουν έναν στόχο και να χάσουν πολύτιμο στόχο, αναζητούν ανάμεσα στις κοινές ευπάθειες ασφάλειες των συστημάτων αυτών για να πετύχουν την τελική κλοπή των δεδομένων και υπάρχουν κυριολεκτικά χιλιάδες από αυτούς. Μόλις λοιπόν ενας κακόβουλος χρήστης εντοπίσει μια αδυναμία μπορεί γρήγορα να εκμεταλλευτεί πολλαπλά CMS σε πολλές εταιρείες.

1.2 Ασφαλής ιστότοπος

Ανδιαφισβητητα το τοπίο απειλής για την ασφάλεια αλλάζει συνεχώς, παρόλα αυτά οι ίδιες οι επιχειρήσεις μπορούν να προστατεύσουν τον εαυτό τους με μερικά απλά βήματα.

Αρχικά η συνειδητοποίηση αποτελεί το κλειδί για την προστασία της επιχείρησης. Με μια μικρή έρευνα στα τρωτά σημεία της πλατφόρμας ο επιχειρηματίας μπορεί να είναι σε θέση και να τα αντιμετωπίσει με τη συνεργασία ειδικών. Έπειτα η συνεχής παρακολούθηση μπορεί να αποδειχθεί σωτήρια. Έχοντας ειδοποιήσεις σε πραγματικό χρόνο οι οποίες παρακολουθούν μια βασική γραμμή συμπεριφοράς, μπορεί να πραγματοποιείται έλεγχος όπου μια μικρή αλλαγή μπορεί να εντοπιστεί αμέσως. Καθώς ο κώδικας που συντάχθηκε από κάποιον τρίτο δεν μπορεί να είναι ελεγχόμενος χρειάζεται συνεχής παρακολούθηση. Τελικά ο χρυσός κανόνας για όλες τις επιχειρήσεις είναι να υπάρχει έλεγχος από τις ίδιες και να μην εμπιστεύονται τις αυτόματες ενημερώσεις καθώς αυτες δεν προσφέρουν καμία ασφάλεια στο σύστημα.

2. Συστήματα Διαχείρισης Περιεχομένου (CMS)

Τα συστήματα διαχείρισης περιεχομένου (CMS) παρέχουν ένα προσαρμόσιμο περιβάλλον πολλών χρηστών (με διαφορετικά επίπεδα άδειας πρόσβασης) για να διαχειριστούν περιεχόμενο, δεδομένα ή πληροφορίες ενός προγράμματος ή μιας εφαρμογής internet. Με τον όρο διαχείριση περιεχομένου εννοούμε την δημιουργία, τροποποίηση, δημοσίευση, συνεργασία, αρχειοθέτηση και αναφορά του. Ενα απο τα κύρια χαρακτηριστικά των CMS τελικά είναι η δυνατότητα που προσφέρει στους χρήστες να προσθέτουν εύκολα διάφορες λειτουργίες ανάλογα τις ανάγκες τους.

Τα πρώτα CMS εμφανίστηκαν στα τέλη της δεκαετίας του 1990, αλλα έγιναν δημοφιλή στα μέσα του 2000. Έχουν αναπτυχθεί πάνω από 200 CMS σε διάφορες γλώσσες προγραμματισμού όπως η PHP, JAVA, Perl, .NET και άλλα χωρισμένα ανοιχτού κώδικα και ιδιόκτητα CMS, το καθένα με τις δικές υποστηριζόμενες βάσεις δεδομένων όπως είναι η MySQL, SQLite, Oracle, PostgreSQL). [1]

2.1 Βασικά χαρακτηριστικά ενός CMS

1. Αυτοματοποιημένα πρότυπα

   Δημιουργία τυποποιημένων οπτικών προτύπων που μπορούν να εφαρμοστούν αυτόματα στο νέο και υπάρχον περιεχόμενο, δημιουργώντας μια κεντρική θέση για να αλλάξουν την εμφάνιση σε μια ολόκληρη ομάδα περιεχομένου σε ένα site.

2. Εύκολα επεξεργάσιμο περιεχόμενο

   Μόλις το περιεχόμενό διαχωριστεί από την οπτική παρουσίαση της ιστοσελίδας, γίνεται συνήθως ευκολότερο και γρηγορότερο να τροποποιηθεί και να χειριστεί. Τα περισσότερα CMS περιλαμβάνουν WYSIWYG (What You See Is What You Get) εργαλεία επεξεργασίας επιτρέποντας στα μη τεχνικά άτομα να δημιουργήσουν και να επεξεργαστούν το περιεχόμενο.

3. Εξελικτικά σύνολα χαρακτηριστικών γνωρισμάτων

   Τα περισσότερα CMS έχουν plug-ins ή ενότητες (modules) που μπορούν να εγκατασταθούν εύκολα για να επεκτείνουν τη λειτουργία μιας υπάρχουσας ιστοσελίδας.

4. Βελτιώσεις προτύπων

Τα CMS λαμβάνουν συνήθως updates που περιλαμβάνουν νέα σύνολα χαρακτηριστικών γνωρισμάτων (feature sets) και κρατούν το σύστημα σύμφωνα τα τρέχοντα πρότυπα του ιστοχώρου.

5. Διαχείριση ροής εργασίας

Η ροή εργασίας (workflow) είναι η διαδικασία δημιουργίας κύκλων από διαδοχικούς και παράλληλους στόχους που πρέπει να ολοκληρωθούν στο CMS. Για παράδειγμα, ένας content creator υποβάλλει μια ιστορία αλλά δεν τη δημοσιεύει στον ιστοχώρο έως ότου την καθαρίσει ο editor και την εγκρίνει ο editor-in-chief.

Διαχείριση εγγράφων

Τα CMS μπορούν να παρέχουν έναν τρόπο διαχείρισης για τον κύκλο ζωής ενός εγγράφου από τον αρχικό χρόνο δημιουργίας του.[2]

2.2 Οι απαιτήσεις του συστήματος

Διακομιστής εφαρμογών - Application server

Ένας διακομιστής εφαρμογών είναι ένας διακομιστής που φιλοξενεί εφαρμογές παρέχοντας τόσο ευκολίες για τη δημιουργία εφαρμογών όσο και περιβάλλον για την εκτέλεση τους. Αποτελεί λοιπόν μια μηχανή η οποία παραδίσει τις εφαρμογές στους υπολογιστές των πελατών ή στις συσκευές διαδικτυακά χρησιμοποιώντας HTTP. Επίσης εκτός από τη δημιουργία ιστοσελίδων, εφαρμόζουν υπηρεσίες όπως η ομαδοποίηση, η αποτυχία και η εξισορρόπηση φορτίου, ώστε οι προγραμματιστές να μπορούν να επικεντρωθούν στην εφαρμογή της επιχειρηματικής λογικής. Υπάρχουν πολλοί διακομιστές εφαρμογών. Η τελική επιλογή μπορεί να καθορίσει το συνολικό κόστος, την απόδοση, την αξιοπιστία καθώς και την επεκτασιμότητα μιας εφαρμογής. [3]

Βάση δεδομένων - DataBase

Βάση δεδομένων ονομάζεται μια οργανωμένη συλλογή δεδομένων που αποθηκεύονται και έχουν πρόσβαση ηλεκτρονικά. Τα αποθηκευμένα αρχεία μπορούν να ανακτηθούν γρήγορα προκειμένου να χρησιμοποιηθούν για μελλοντικές αποφάσεις. Ο σχεδιασμός τους καλύπτουν τεχνικές και πρακτικά ζητήματα όπως είναι η ασφαλής αποθήκευση δεδομένων, η πραγματοποίηση ερωτημάτων καθώς και τη μοντελοποίηση δεδομένων. Οι μικρές βάσεις δεδομένων έχουν την δυνατότητα να αποθηκευτούν σε ένα σύστημα αρχείων, ενώ οι μεγάλες βάσεις δεδομένων φιλοξενούνται είτε σε συμπλέγματα υπολογιστών είτε αποθήκευση cloud . [4]

Γλώσσα προγραμματισμού - Programming Language

Μια γλώσσα προγραμματισμού υπολογιστή είναι μια γλώσσα που χρησιμοποιείται για τη σύνταξη προγραμμάτων υπολογιστή, η οποία περιλαμβάνει έναν υπολογιστή που εκτελεί κάποιο είδος υπολογισμού ή αλγόριθμου και πιθανώς ελέγχει εξωτερικές συσκευές όπως εκτυπωτές, μονάδες δίσκου, ρομπότ, και ούτω καθεξής. Οι γλώσσες προγραμματισμού, όπως οι φυσικές γλώσσες, καθορίζονται από τους συντακτικούς και σημασιολογικούς κανόνες που περιγράφουν τη δομή και τη σημασία τους αντίστοιχα. [5]

Διακομιστής Ιστού - Web Server

Ο διακομιστής ιστού είναι ένα λογισμικό το οποίο δέχεται αιτήματα HTTP ή με την ασφαλή παραλλαγή HTTPS. Ένας χρήστης λοιπόν ξεκινάει την επικοινωνία μέσω ενός browser στέλνοντας ένα αίτημα για μια ιστοσελίδα και ο διακομιστής απαντάει με το περιεχόμενου αυτής της ιστοσελίδας ή με την εμφάνιση σφάλματος. Στη συνέχεια του κεφαλαίου αυτού θα δούμε πιο αναλυτικά την υποενότητα αυτή καθώς αποτελεί ένα από τα πιο σημαντικά σημεία ενός ιστοτόπου[6]

2.3 Κατηγορίες CMS

Υπάρχουν έξι κύριες κατηγορίες συστημάτων διαχείρισης περιεχομένου: [7]

- Επιχειρησιακά CMS
- Συστήματα διαχείρισης web περιεχομένων CMS (WCMS)
- Συστήματα διαχείρισης εγγράφων - Document management system (DMS)
- Συστήματα διαχείρισης περιεχομένου συστατικών
- Συστήματα διαχείρισης περιεχομένου φορητών συσκευών

2.3.1 Ανάλυση των κατηγοριών

Επιχειρησιακά CMS

Το περιεχόμενο, τα έγγραφα, τα στοιχεία και τα παραστατικά που σχετίζονται με τις οργανωτικές διαδικασίες μιας επιχείρησης αποτελούν τα κύρια στοιχεία του συστήματος αυτού με σκοπό την σωστή και γρήγορη διαχείριση των πληροφοριών με όλες τις πληροφορίες ανεξαρτήτως τοποθεσίας. [7]

Συστήματα διαχείρισης web περιεχομένων CMS

Η διαχείριση περιεχομένου ιστοσελίδων (WCM) είναι ένα σύστημα CMS σχεδιασμένο για να διευκολύνει την κατασκευή ιστοσελίδων, την επεξεργασία περιεχομένου καθώς και την αλληλεπίδραση του χρήστη με το σύστημα χωρίς όμως αυτός να είναι απαραίτητο να γνωρίζει κάποια γλώσσα προγραμματισμού. Τα τελευταία χρόνια νέες εξελίξεις έχουν φέρει τα cms και σε επιπλέον τομείς της αγοράς, όπως την ανάπτυξη του marketing καθώς δίνεται η δυνατότητα παρακολούθησης ιστοτόπων και αποστολή email.

Πολλά web-based συστήματα διαχείρισης περιεχομένου υφίστανται, τόσο Ανοιχτού Κώδικα όσο και κλειστού έχοντας όμως το λογισμικό ανοιχτού κώδικα να κυριαρχεί στα εμπορικά πακέτα. Στη συνέχεια θα αναλύσουμε λεπτομερώς τις διαφορές, τα πλεονεκτήματα αλλα και τα μειονεκτήματα ενός ανοιχτού και κλειστού λογισμικού.

Συστήματα διαχείρισης εγγράφων

Το σύστημα διαχείρισης εγγράφων (DMS) χρησιμοποιείται κυρίως για την αποθήκευση και παρακολούθηση ηλεκτρονικών εγγράφων. Επίσης, παρέχει την δυνατότητα της παρακολούθησης διαφόρων χρηστών. Βασικό χαρακτηριστικό τους λοιπόν αποτελεί η δυνατότητα λήψης σύλληψης φορμών, να αποθηκεύουν εικόνες και η ανταπόκριση του συστήματος να είναι άμεση.

Με το πέρασμα των χρόνων οι δυνατότητες των συστημάτων αυτών εξελίχθηκαν επιτρέποντας τη χρήση όλων των αποθηκευμένων αρχείων σε ένα δίκτυο, καλύπτοντας με αυτόν τον τρόπο συνεργατικά εργαλεία, δυνατότητες ελέγχου, ασφάλεια και ηλεκτρονικά έγγραφα.

Συστήματα διαχείρισης περιεχομένου συστατικών

Στο σύστημα αυτό η αποθήκευση και διαχείριση των περιεχομένων πραγματοποιείται σε επιπεδο συστατικού και οχι εγγράφου προκειμένου να επιτευχθεί μεγαλύτερη επαναχρησιμοποίηση των περιεχομένων. Τα συστατικά μπορεί να είναι μια περιγραφή προϊόντων, μια εικόνα ή ένας πίνακας καθώς το μέγεθος του μπορεί να ποικίλει. Η αποθήκευση του συστατικού γίνεται μόνο μια φορά στο σύστημα δίνοντας συνέπεια όταν χρησιμοποιούνται.

Συστήματα διαχείρισης περιεχομένου φορητών συσκευών

Αυτός ο τύπος συστήματος επιτρέπει την αποθήκευση και την παροχή περιεχομένου και υπηρεσιών σε φορητές συσκευές όπως είναι τα κινητά τηλέφωνα. Η χρήση αυτού του τύπου έχει μερικούς περιορισμούς όπως είναι η μειωμένη χωρητικότητα των συσκευών, μειωμένος χώρος αποθήκευσης, μικρό μέγεθος οθόνης, αδύνατους επεξεργαστές και περιορισμένο ασύρματο εύρος ζώνης. Παρόλα αυτά η ζήτηση των συστημάτων αυτών αυξήθηκε ραγδαία λόγω της αύξησης της πολυπλοκότητας στη χρήση τους.

2.4 Διάφοροι τύποι CMS

Στη συνέχεια του κεφαλαίου αυτού θα καταγράψουμε τους διάφορους τύπους των συστημάτων διαχείρισης περιεχομένου.

- Electronic news / magazine publishing: Εργαλεία κατάλληλα για τη δημοσίευση ειδήσεων. Γρήγορα με εύκολο χειρισμό περιεχομένου. Παραδείγματα αποτελούν τα: Errom, expressroom και SlasjDot
- E-business / E- Commerce: Δυνατότητα δημιουργίας online καταστημάτων με ταυτόχρονη αλληλεπίδραση χρηστών
- Document management systems: Συστήματα διαχείρισης περιεχομένου με την ταυτόχρονη συνεργασία πληροφοριών
- Web content management frameworks

2.5 Στατικές σελίδες

Τα περιεχόμενα μιας στατικής ιστοσελίδας μεταφέρονται με την ίδια μορφή σε όλους του χρήστες με την μορφή που είναι αποθηκευμένα στο σύστημα αρχείων του εξυπηρετητή

ιστοσελίδων (web server). Η αποθήκευση των στατικών ιστοσελίδων γίνεται σε HTML μορφή και η μεταφορά τους πραγματοποιείται με τη χρήση του πρωτοκόλλου HTTP.

Πλεονέκτημα μια στατικής ιστοσελίδας αποτελεί η ευκολία στην δημιουργία της, δεν χρειάζονται προγραμματιστικές ικανότητες. Επιπλέον δεν αποτελεί απαραίτητη η ύπαρξη ειδικού λογισμικού στον εξυπηρετητή ιστοσελίδων για την ολοκλήρωση της δημοσίευσης των σελίδων. Επίσης η ιστοσελίδα είναι απευθείας διαθέσιμη στον φυλλομετρητή χωρίς την ανάγκη μεσολαβητή με κατάλληλο λογισμικό.

Μειονέκτημα όμως αποτελεί η διαδραστικότητα με τον χρήστη η οποία είναι δύσκολη καθώς για την ανανέωση μια σελίδας χρειάζεται να ανοίξουμε το κατάλληλο πρόγραμμα, να κάνουμε τις αλλαγές να τις αποθηκευσουμε και στη συνέχεια να τις ανεβάσουμε ξανά στο διαδίκτυο. Τελικά η διαχείριση μεγάλου αριθμού στατικών ιστοσελίδων δεν είναι εύκολη χωρίς αυτόματα εργαλεία καθώς είναι χρονοβόρα και μερικές φορές δύσκολη για τους απλούς χρήστες.

2.6 Δυναμικές σελίδες

Το κύριο χαρακτηριστικό των δυναμικών ιστοσελίδων αποτελεί η απευθείας δημιουργία της (δυναμική) την στιγμή που ο χρήστης στέλνει το αίτημα στον εξυπηρετητή ιστοσελίδων αποτελώντας μια απο τις πιο σημαντικές δυνατότητες της νέας γενιάς του παγκόσμιου ιστού όπου οι πληροφορίας μιας σελίδας διαμοιράζονται σε πολλές ιστοσελίδες.

2.7 Κατηγορίες CMS ανοιχτού και κλειστού κώδικα

2.7.1 Λογισμικό ανοικτού κώδικα

Λογισμικό ανοιχτού κώδικα είναι λογισμικό του οποίου ο πηγαίος κώδικας διατίθεται ελεύθερα για λήψη.

Ο κώδικας μπορεί να ελεγχθεί και να τροποποιηθεί ώστε να ταιριάζει στις συγκεκριμένες απαιτήσεις του χρήστη. Καθώς ο πηγαίος κώδικας είναι ανοιχτός, οι χρήστες μπορούν να συνεισφέρουν στη διόρθωση σφαλμάτων και τρωτών σημείων στον κώδικα και μερικές φορές μπορούν να παρέχουν βοήθεια με το προϊόν σε άλλους χρήστες. Επίσης, οποιοσδήποτε αριθμός ατόμων θα μπορούσε ενδεχομένως να συνεισφέρει στην ανάπτυξη λογισμικού και η ανάπτυξη είναι συχνά μια ομαδική προσπάθεια.

Το λογισμικό ανοιχτού κώδικα είναι συνήθως δωρεάν, αλλά δεν είναι όλο το ελεύθερο λογισμικό ανοιχτού κώδικα (το λογισμικό κλειστού κώδικα μπορεί επίσης να είναι δωρεάν).

Η ποιότητα του τελικού προϊόντος μπορεί να ποικίλλει: κάποιο λογισμικό ανοιχτού κώδικα είναι εξαιρετικά υψηλής ποιότητας και ενημερώνεται τακτικά, ενώ κάποιο περιέχει πολλά σφάλματα και γρήγορα γίνεται ξεπερασμένο.[8]

2.7.2 Λογισμικό κλειστού κώδικα

Το λογισμικό κλειστού κώδικα είναι λογισμικό για το οποίο ο πηγαίος κώδικας δεν είναι ελεύθερα διαθέσιμος. Αναπτύσσεται και παρέχεται στον χρήστη ως ένα πλήρως μεταγλωττισμένο, εκτελέσιμο σύνολο αρχείων. Ο προγραμματιστής συχνά παρέχει υποστήριξη στους χρήστες μετά την αγορά και διασφαλίζει ότι το λογισμικό λειτουργεί όπως αναμένεται. Καθώς ο χρήστης δεν διαθέτει τον πηγαίο κώδικα, δεν μπορεί να κάνει αλλαγές στο λογισμικό.

Το λογισμικό κλειστού κώδικα πωλείται συνήθως στους τελικούς χρήστες, αν και μερικές φορές είναι διαθέσιμο δωρεάν. Είναι σημαντικό ότι κατά την αγορά λογισμικού, ο χρήστης δεν αγοράζει το ίδιο το λογισμικό, αλλά αγοράζει άδεια χρήσης του λογισμικού.

2.7.3 Διαφορές μεταξύ λογισμικού ανοιχτού κώδικα και λογισμικού κλειστού κώδικα

Στη συνέχεια του κεφαλαίου αυτού θα καταγράψουμε αναλυτικά τις πέντε βασικές διαφορές (Κόστος, Ασφάλεια, Υποστήριξη, Διαθεσιμότητα και χρηστικότητα) του λογισμικού ανοιχτού κώδικα και του λογισμικού κλειστού κώδικα [9]

1. **Ασφάλεια**

   Το ζήτημα της ασφάλειας είναι πολύ αμφιλεγόμενο καθώς κάθε λογισμικό έχει δύο όψεις του νομίσματος. Ο κώδικας του λογισμικού ανοιχτού κώδικα μπορεί να προβληθεί, να μοιραστεί και να τροποποιηθεί από την κοινότητα, πράγμα που σημαίνει ότι ο καθένας μπορεί να διορθώσει, να αναβαθμίσει και να δοκιμάσει τον κατεστραμμένο κώδικα. Τα σφάλματα διορθώνονται γρήγορα και ο κώδικας ελέγχεται διεξοδικά μετά από κάθε έκδοση. Ωστόσο, λόγω διαθεσιμότητας, ο πηγαίος κώδικας είναι ανοιχτός για εξάσκηση στους χάκερ.

   Αντίθετα, το λογισμικό κλειστού κώδικα μπορεί να επιδιορθωθεί μόνο από έναν προμηθευτή. Εάν κάτι πάει στραβά με το λογισμικό, στέλνετε ένα αίτημα και περιμένετε την απάντηση από την ομάδα υποστήριξης. Η επίλυση του προβλήματος μπορεί να διαρκέσει πολύ περισσότερο σε σύγκριση με το OSC.

   Όσον αφορά την επιλογή του πιο ασφαλούς λογισμικού, η απάντηση είναι ότι καθένα από αυτά έχει τα πλεονεκτήματα και τα μειονεκτήματά του. Έτσι, είναι συχνά μια πρόκληση για τις επιχειρήσεις που εργάζονται σε έναν συγκεκριμένο κλάδο.

2. **Κόστος**

   Το λογισμικό ανοιχτού κώδικα αναφέρεται συχνά ως δωρεάν λογισμικό. Μπορεί, ωστόσο, να έχει κόστος για πρόσθετα, όπως βοήθεια, πρόσθετες υπηρεσίες ή πρόσθετη λειτουργικότητα. Έτσι, μπορείτε να πληρώσετε για μια υπηρεσία με OSS.

   Το λογισμικό κλειστού κώδικα είναι συνήθως ένα λογισμικό επί πληρωμή. Το κόστος μπορεί να ποικίλλει ανάλογα με την πολυπλοκότητα του λογισμικού. Ενώ η τιμή μπορεί να είναι υψηλότερη, αυτό που παίρνετε είναι καλύτερο προϊόν, πλήρης υποστήριξη, λειτουργικότητα και καινοτομία. Ωστόσο, οι περισσότερες εταιρείες παρέχουν δωρεάν δοκιμές για να πείσουν τον αγοραστή ότι το λογισμικό τους είναι το σωστό.

3. **Ποιότητα Υποστήριξης**

   Συγκρίνοντας την υποστήριξη λογισμικού ανοιχτού κώδικα και κλειστού κώδικα, είναι προφανές ότι το CSS κυριαρχεί σε αυτήν την περίπτωση. Το κόστος για αυτό περιλαμβάνει μια επιλογή επικοινωνίας με την υποστήριξη και λήψης σε μία εργάσιμη ημέρα στις περισσότερες περιπτώσεις. Η απάντηση είναι καλά οργανωμένη και τεκμηριωμένη.

Για λογισμικό ανοιχτού κώδικα, τέτοια επιλογή δεν παρέχεται. Οι μόνες επιλογές υποστήριξης είναι τα φόρουμ, τα χρήσιμα άρθρα και ένας μισθωμένος ειδικός. Ωστόσο, δεν αποτελεί έκπληξη το γεγονός ότι χρησιμοποιώντας τέτοιου είδους υπηρεσία δεν υπάρχει υψηλό επίπεδο ανταπόκρισης.

4. Διαθεσιμότητα πηγαίου κώδικα

Το λογισμικό ανοιχτού κώδικα παρέχει τη δυνατότητα αλλαγής του πηγαίου κώδικα χωρίς περιορισμούς. Οι μεμονωμένοι χρήστες μπορούν να αναπτύξουν αυτό που θέλουν και να επωφεληθούν από την καινοτομία που αναπτύχθηκε από άλλους εντός της κοινότητας χρηστών. Καθώς ο πηγαίος κώδικας είναι εύκολα προσβάσιμος, δίνει τη δυνατότητα στους προγραμματιστές λογισμικού να βελτιώσουν τα ήδη υπάρχοντα προγράμματα.

Το λογισμικό κλειστού κώδικα είναι πιο περιορισμένο από το λογισμικό ανοιχτού κώδικα, επειδή ο πηγαίος κώδικας δεν μπορεί να αλλάξει ή να προβληθεί. Ωστόσο, αυτός ο περιορισμός είναι που μπορεί να συμβάλει στην ασφάλεια και την αξιοπιστία του.

5. Ευχρηστία

Η χρηστικότητα είναι ένα επίπονο θέμα του λογισμικού ανοιχτού κώδικα. Οι οδηγοί χρήσης είναι γραμμένοι για προγραμματιστές και όχι για απλούς χρήστες. Επίσης, αυτά τα εγχειρίδια δεν συμμορφώνονται με τα πρότυπα και τη δομή.

Για το λογισμικό κλειστού κώδικα, η χρηστικότητα είναι ένα από τα πλεονεκτήματα. Η τεκμηρίωση είναι συνήθως καλογραμμένη και περιέχει λεπτομερείς οδηγίες.

Συμπεράσματα

Η αγορά είναι γεμάτη από εφαρμογές ανοιχτού κώδικα και κλειστού κώδικα. Η βασική διαφορά έγκειται στην τιμή. Τα συστήματα ανοιχτού κώδικα είναι δωρεάν, ενώ για προγράμματα κλειστού κώδικα υπάρχει τις περισσότερες φορές κόστος. Με την πληρωμή, δίνεται υποστήριξη πελατών και εμπιστοσύνη. Επειδή τα προγράμματα ανοιχτού κώδικα είναι δωρεάν, δεν έχουν τέτοια επιλογή. Ωστόσο, η κοινότητά τους σε διάφορα φόρουμ είναι πολύ ενεργή και πάντα έτοιμη να βοηθήσει.

Τα οφέλη των λύσεων ανοιχτού κώδικα είναι κυρίως η ευελιξία και η επεκτασιμότητα. Υπάρχει πλήρης έλεγχος σε κάθε πτυχή του σχεδιασμού του ιστότοπού σας, χάρη στον ανοιχτό κώδικα. Όταν η επιχείρησή επεκτείνεται και οι μηνιαίες πωλήσεις αυξάνονται, υπάρχει η δυνατότητα ενημέρωσης εύκολα και χωρίς επιπλέον κόστος να υποστηρίξει περισσότερο αυξημένο όγκο πωλήσεων.

Το λογισμικό κλειστού κώδικα είναι πιο εύκολο για αρχάριους ή για όσους δεν ξέρουν να κωδικοποιούν. Επίσης, οι ιστότοποι κλειστού κώδικα είναι ευκολότερο και πιο γρήγορο να ρυθμιστούν.

Οι κορυφαίες εφαρμογές ανοιχτού κώδικα είναι το Wordpress το Joomla και τρίτο έρχεται το Drupal, καθώς το Shopify είναι δημοφιλείς πλατφόρμες κλειστού κώδικα.

Κάθε τύπος πλατφόρμας έχει τη δική του φιλοσοφία, μεθοδολογία, πλεονεκτήματα και μειονεκτήματα. Δεν υπάρχει μονοσήμαντη επιλογή, καθώς εξαρτάται από τις επιχειρηματικές ανάγκες.

2.8 Η Τελική επιλογή CMS

Αδιαμφισβήτητα το κάθε σύστημα διαχείρισης περιεχομένου αποτελείται απο διαφορετικές λειτουργίες και χαρακτηριστικά. Πώς όμως ενας χρήστης μπορεί να επιλέξει το πιο κατάλληλο για αυτόν προκειμένου να καλύπτει όλες τις ανάγκες του; Ακολουθούν μερικές από τις βασικές λειτουργίες που πρέπει ένας χρήστης να σκεφτει πριν την επιλογή αυτή.

- Το είδος της διαδικτυακής επιχείρησης που θέλουμε να δημιουργήσουμε (eshop, blog)
- Ποιος θα επεξεργάζεται το περιεχόμενο και σε τι βαθμό μελλοντικά
- Δυνατότητες αναζήτησης
- Προσαρμογή, ευελιξία που θα χρειαστούμε στο μέλλον
- Αλληλεπίδραση με τους χρήστες (φόρμες επικοινωνίας, δημιουργία λογαριασμού, εκπτώσεις)
- Δικαιώματα χρηστών
- Υποστήριξη multi-site
- Δυνατότητα πολυγλωσσίας

Εφόσον λοιπόν γίνει η καταγραφή των αναγκών και ενημέρωση των δυνατοτήτων του κάθε cms ο χρήστης θα είναι σε θέση να επιλέξει το πιο κατάλληλο σύστημα και να προχωρήσει έπειτα στην επιλογή παρόχου φιλοξενίας. Στη συνέχεια αναλύουμε τα επόμενα βήματα για τη σωστή ολοκλήρωση της διαδικασίας επιλογής πριν την τελική εκκίνηση της κατασκευής.

2.9 Φιλοξενία ενός ιστοτόπου στο διαδίκτυο

2.9.1 Domain name

Το domain name αποτελεί την ταυτότητα ενός ιστοτόπου. Τα ονόματα τομέα σχηματίζονται ακολουθόντας τους κανόνες του συστήματος DNS. Κάθε διεύθυνση κατοχυρώνεται και είναι μοναδική.

Όταν λοιπόν θέλουμε να αναζητήσουμε μια ιστοσελίδα μέσα από έναν browser θα πρέπει να γνωρίζουμε το ακριβές όνομα της διεύθυνσης που θέλουμε να ψάξουμε και να την πληκτρολογήσουμε στη γραμμή διευθύνσεων. Με την διαδικασία αυτή αναθέτουμε στον web browser να ψάξει σε ποιο web server είναι αποθηκευμένη η συγκεκριμένη διεύθυνση. [10]

2.9.2 Web Server

Ο web server είναι ο χώρος αποθήκευσης ενός domain. Προκειμένου ένα site να είναι διαθέσιμο στους χρήστες υπεύθυνος είναι ο web server. Υπάρχουν δύο κύριοι web server. Ο Apache που συνήθως χρησιμοποιεί λειτουργικό σύστημα Linux και ο IIS που χρησιμοποιεί λειτουργικό σύστημα Windows. Προκειμένου το Joomla να μπορέσει να λειτουργήσει σε κάποιον web server προϋπόθεση αποτελεί η υποστήριξη της php.[10]

2.9.3 Web Hosting

Μια επιχείρηση προκειμένου να ανοίξει και να πουλήσει τα προϊόντα της χρειάζεται ένα φυσικό κατάστημα. Οι ίδιοι κανόνες ισχύουν και στον ψηφιακό κόσμο κατά τη δημιουργία ενός ιστότοπου.

Ένας διαδικτυακός ιστότοπος αποτελείται από μια σειρά από αρχεία, εικόνες και HTML κώδικα τα οποία συνθέτουν τον ιστότοπο. Προκείμενου τα αρχεία αυτά να είναι διαθέσιμα στους χρήστες χρειάζονται ένα χώρο φιλοξενίας. Ένας πάροχος φιλοξενίας λοιπόν παρέχει μια θέση σε έναν διακομιστή ιστού για την αποθήκευση όλων των αρχείων και είναι υπεύθυνος για την παράδοση των αρχείων του ιστότοπού αμέσως μόλις ένα πρόγραμμα περιήγησης υποβάλει αίτημα πληκτρολογώντας το όνομα τομέα. Με την πληρωμή μιας υπηρεσίες φιλοξενίας, "ενοικιάζεται" ενας αποθηκευτικός χώρο στο διαδίκτυο όπως ακριβώς συμβαίνει σε ένα φυσικό κατάστημα για την επιχείρησή.

Καθώς υπάρχουν διαφορετικοί πάροχοι φιλοξενίας για την σωστή επιλογή πρέπει να καταγραφούν πρώτα οι ανάγκες που έχει ο ιστότοπος που πρόκειται να δημιουργηθεί. Όπως για παράδειγμα τι είδος θα είναι η ιστοσελίδα (eCommerce, blog, portfolio), αν θα χρειάζεται η δημιουργία email, τι ασφάλεια προσφέρουν. [10]

2.10 Τύποι φιλοξενίας Ιστού

1. Κοινόχρηστη φιλοξενία (Shared Hosting)

- Η κοινόχρηστη φιλοξενία είναι ο πιο κοινός τύπος φιλοξενίας Ιστού και είναι κατάλληλος για τους περισσότερους ιδιοκτήτες διαδικτυακών επιχειρήσεων. Με την κοινή φιλοξενία, πολλοί πελάτες μοιράζονται χώρο αποθήκευσης σε έναν ισχυρό διακομιστή. Υπάρχουν πολλά πλεονεκτήματα στην κοινή φιλοξενία, όπως: [11]
- Προσιτές τιμές: Είναι πολύ φθηνότερο να μοιράζεται ο χώρος σε έναν διακομιστή παρά ολόκληρο το μηχάνημα.
- Ευκολία χρήσης: Ο διακομιστής είναι προρυθμισμένος, καλά οργανωμένος, εύκολος στη χρήση και η εταιρεία φιλοξενίας κάνει όλες τις ενημερώσεις συντήρησης και ασφάλειας.

2. Αφιερωμένη φιλοξενία (Dedicated Hosting)

Αντί ο χώρος να μοιράζεται, η χρήση του διακομιστή είναι αποκλειστική για τον ιδιοκτήτη. Τα πλεονεκτήματα της αποκλειστικής φιλοξενίας περιλαμβάνουν:

- Προσαρμογή. Το λογισμικό και το υλικό προσαρμόζεται ώστε να ανταποκρίνεται στις ατομικές ανάγκες
- Απεριόριστοι πόροι. Εφόσον δεν γίνεται διαμοιρασμός του διακομιστή με κανέναν, όλος ο χώρος αποθήκευσης είναι του ιδιοκτήτη.
- Πλήρης έλεγχος.[11]

3. VPS Hosting

Η φιλοξενία εικονικού ιδιωτικού διακομιστή (VPS) περιλαμβάνει όλες τις δυνατότητες ενός αποκλειστικού διακομιστή, αλλά στην τιμή ενός κοινόχρηστου διακομιστή. Ακολουθούν μερικοί από τους κορυφαίους λόγους για τους οποίους η φιλοξενία VPS μπορεί να ταιριάζει πολύ:[11]

- Λειτουργικότητα. Η φιλοξενία VPS είναι χτισμένη σε ένα cPanel και υποστηρίζει την εύκολη πλοήγηση με πολλά διαισθητικά εργαλεία.
- Λειτουργίες με ένα κλικ. Υπάρχει πρόσβαση σε εγκαταστάσεις με ένα κλικ των WordPress, Magento και Drupal.
- Εύκολη πλοήγηση στον ιστότοπο.

4. Cloud Hosting

Η φιλοξενία στο cloud θεωρείται συχνά η πιο αξιόπιστη από όλες τις υπηρεσίες. Αντί να βασίζεται στον χώρο του δίσκου ενός μεμονωμένου διακομιστή, αντλεί την ισχύ του από πολλούς πόρους, διασφαλίζοντας ότι δεν θα υπάρχει ποτέ χρόνος διακοπής λειτουργίας. Άλλα οφέλη περιλαμβάνουν:[11]

- Επεκτασιμότητα . Δυνατότητα προσθήκης στον χώρο στο cloud ανά πάσα στιγμή.
- Αμέτρητο εύρος ζώνης. Δεν υπάρχει κίνδυνος ο ιστότοπος να πέσει λόγω αποτυχίας διακομιστή.

3. Η εξέλιξη των CMS

Μέχρι πριν από λίγα χρόνια η ιδέα κατασκευής ιστοσελίδας και διαχείρισής της έμοιαζε πολύ δύσκολη και απαιτητική. Τα τελευταία χρόνια όμως αυτό άλλαξε δίνοντας στους επιχειρηματίες ή και απλούς χρήστες την δυνατότητα με εύκολο και γρήγορο τρόπο να διαχειρίζονται ή ακόμα και να κατασκευάζουν οι ίδιοι την διαδικτυακή τους ταυτότητα. Την επανάσταση αυτή την έφεραν τα συστήματα διαχείρισης περιεχομένου επιτρέποντας στους χρήστες να διαχειρίζονται το διαδικτυακό τους περιεχόμενο, όπως κείμενα, βίντεο πίνακες, φόρμες επικοινωνίας και πολλά άλλα.

Τα πρώτα συστήματα διαχείρισης περιεχομένου δημιουργήθηκαν μέσα στους οργανισμούς που τα χρειαζόταν από το αρμόδιο τμήμα. Πρώτη η εταιρία CNET το 1995[12], πήρε την απόφαση επέκτασης του εσωτερικού συστήματος διαχείρισης περιεχομένου, το οποίο χρησιμοποιούνταν για να δημοσιεύεται ηλεκτρονικό υλικό με κύριος στόχο την εμπορική εκμετάλλευση των CMS. Οι ιδρυτές Halsey Minor και Jonathan Rosenberg λοιπόν δημιούργησαν το δικό τους σύστημα διαχείρισης περιεχομένου Ιστού και εισήγαγε μια σειρά από βασικές δυνατότητες του σήμερα, όπως η επαναχρησιμοποίηση περιεχομένου και η εξατομίκευση.

Με το πέρασμα των χρόνων ακολουθησαν πολλες αποπειρες ανάπτυξης περιεχομένου και τελικά σήμερα πιθανολογείται ότι υπάρχουν εκατοντάδες εφαρμογές cms[13]. Με την συνεχόμενη εξέλιξη και ανάπτυξη των συστημάτων αυτών οι επιχειρήσεις υποχρεούνται να ενημερώνονται διαρκώς διατηρώντας το σύστημα τους ενημερωμένου και ασφαλή. Στις μέρες μας τα συστήματα CMS έχουν αλλάξει πολύ από εκείνα που υπήρχαν τα προηγούμενα χρόνια. Έχουν γίνει περισσότερο φιλικά προς το χρήστη, μειώνοντας ακόμα περισσότερο τις απαιτήσεις σε γνώση από την πλευρά του σχεδιαστή. Παράλληλα εμφανίζονται να προσφέρουν μικρούς χρόνους υλοποίησης μιας ιστοσελίδας συγκριτικά με τη χρήση κώδικα. Με την πάροδο των χρόνων τα συστήματα δυναμικής διαχείρισης περιεχομένου γινόντουσαν όλο και πιο δημοφιλή, για να φτάσουμε σήμερα να γνωρίζουμε ότι περίπου το 25% των διαθέσιμων

ιστοσελίδων παγκοσμίως, έχει δημιουργηθεί με αυτή την τεχνική. Μερικά απο τα πιο δημοφιλή [14] cms σήμερα είναι το Wordpress, το joomla και το Drupal για τα οποία θα μιλήσουμε στη συνέχεια του κεφαλαίου αυτού.

3.1 Wordpress

Εικόνα 1: Λογότυπο wordpress

Το Wordpress είναι το πιο δημοφιλές σύστημα διαχείρισης περιεχομένου (CMS) κατακτόντας πάνω από 30 τοις εκατό των σελίδων. Το WordPress ξεκίνησε το 2003 όταν ο Matt Mullenweg και ο Mike O Little δημιούργησαν την πρώτη έκδοση του Wordpress. Το WordPress είναι open source έργο που κυκλοφόρησε υπό την GPL (GNU General Public License) v2.0. Ο Mullenweg ξεκίνησε το WordPress Foundation το 2010 (εμπνευσμένο από το Ίδρυμα Ελεύθερου Λογισμικού και το Ίδρυμα Mozilla) για να παρέχει υποστήριξη της βιωσιμότητας του WordPress και προώθηση του έργου.

Από την αρχή, η αποστολή του WordPress ήταν ο εκδημοκρατισμός της δημοσίευσης, διασφαλίζοντας ότι οποιαδήποτε μη τεχνικό άτομο μπορεί να δημιουργήσει την δική του ιστοσελίδα, ενώ ταυτόχρονα να μπορεί να κατασκευάσει ένα προϊόν που να μπορεί να κλιμακωθεί μέχρι και σε εταιρικούς πελάτες έχοντας ακόμα και περίπλοκες ανάγκες (για παράδειγμα, ηλεκτρονικό εμπόριο, πολύγλωσσο ή κινητό). Η πρόσφατη προσθήκη του WordPress REST (αντιπροσωπευτικό State Transfer) API είναι ένα ακόμα βήμα ανάπτυξης. Χάρη στο API, ο χρήστης μπορεί πλέον να χρησιμοποιήσετε το WordPress ως ένα ακέφαλο CMS για να δημιουργήσει τις δικές του εφαρμογές Ιστού, ενώ επωφελούνται από όλες τις βασικές λειτουργίες του back-end (για παράδειγμα, για συνεργασία και διαχείριση περιεχομένου και χρηστών). [15]

Βασικά Χαρακτηριστικά

- Προσαρμοσμένες Ταξινομίες
- Προσαρμοσμένοι τύποι αναρτήσεων (άρθρα)
- Θέματα WordPress
- Εργαλεία επικοινωνίας μεταξύ ιστολογίων
- Προστασία ανεπιθύμητων
- Πλήρης εγγραφή χρήστη

- Αναρτήσεις που προστατεύονται με κωδικό πρόσβασης
- Εύκολη εισαγωγή
- Διασύνδεση XML-RPC
- Ροή εργασιών
- Έξυπνη μορφοποίηση κειμένου
- Διαχείριση μενού

3.1.1 Κώδικας στο WordPress

Το WordPress περιλαμβάνει περισσότερες από 800 χιλιάδες γραμμές κώδικα (KLOC)[16] που αποτελούνται κυρίως από κώδικα PHP που εξυπηρετεί αιτήματα HTTP κάνοντας ερώτημα σε μια βάση δεδομένων MySQL. Ωστόσο, η χρήση JavaScript παρόλο που το 2018 αυξήθηκε ραγδαία, ειδικά στο front-end του WordPress, τα τελευταία χρόνια φαίνεται να παρουσιάζει ξανα πτώση. Το React χρησιμεύει ως το βασικό πλαίσιο για το νέο WordPress που βασίζεται σε JavaScript εξελίξεις.

Σήμερα, η PHP εξακολουθεί να αντιπροσωπεύει πάνω από το ήμισυ του συνόλου των γραμμών κώδικα (LOC), με επιπλέον 30 τοις εκατό της JavaScript. CSS, HTML και XML αποτελούν τα υπόλοιπα.

Γλώσσα	Γραμμές κώδικα	Γραμμές σχολίων	Αναλογία σχολίων	Κενές γραμμές	Σύνολο Γραμμών	Συνολικό ποσοστό
PHP	313.654	140.204	30,9%	63.463	517.321	44,1%
HTML	241.596	1.093	0,5%	32.294	274.983	23,4%
JavaScript	170.124	34.147	16,7%	22.878	227.149	19,4%
CSS	116.622	8.399	6,7%	24.069	149.090	12,7%
XML	3.060	237	7,2%	814	4.111	0,4%
Πύθων	39	7	15,2%	13	59	0,0%
Μετασχηματισμός XSL	32	33	50,8%	11	76	0,0%
Modelica	18	0	0,0%	0	18	0,0%
SQL	2	5	71,4%	1	8	0,0%
Σύνολα	845.147	184.125		143.543	1.172.815	

Εικόνα 2: Γλωσσική κατανομή

Ο κώδικας του WordPress είναι οργανωμένος σε μερικές δεκάδες βασικά στοιχεία με βάση τη λειτουργικότητά τους (και όχι το μέγεθος ή τη γλώσσα). Καθε συστατικό κωδικοποιείται χρησιμοποιώντας ένα μείγμα διαδικαστικής και αντικειμενοστραφούς τεχνικής προγραμματισμού. [17]

Για ιστορικούς λόγους, το WordPress δεν χρησιμοποιεί σημασιολογική έκδοση. Ετσι, τα δύο πρώτα ψηφία της έκδοσης του ταυτοποιούν μια σημαντική κυκλοφορία, με το τρίτο ψηφίο να προσδιορίζει μικρές αλλαγές στην έκδοση (κυρίως για ενημερώσεις κώδικα ασφαλείας και διορθώσεις σφαλμάτων). Κάθε κυκλοφορία έχει έναν κύριο προγραμματιστή και μπορεί πλέον να έχει και έναν υποψήφιο πελάτη σχεδιαστή. (Αυτή η αναγνώριση του βασικού ρόλου των σχεδιαστών στην ανάπτυξη πλέον εμφανίζεται σε πολλές εταιρείες τεχνολογίας βελτιώνοντας δραστικά την αναλογία προγραμματιστή προς σχεδιαστή[18]. Οι προγραμματιστές

εναλλάσσονται με κάθε κυκλοφορία, το οποίο βοηθά στη συμμετοχή περισσότερων ανθρώπων στις βασικές θέσεις του έργου. Οι προγραμματιστές αποφασίζουν για όλες τις τεχνικές πτυχές της κυκλοφορίας, αλλά εξαρτώνται από την κοινότητα του WordPress για την προώθηση του νέου κώδικα. Η εξέλιξη του κώδικα περιλαμβάνει επίσης την προσθήκη δοκιμαστικής μονάδας (Το WordPress χρησιμοποιεί PHPUnit και QUnit για αυτόματη δοκιμή του PHP και JavaScript, αντίστοιχα).

Πέρα από τους κύριους προγραμματιστές, ένας αριθμός βασικών committers[19] (τα άτομα στο οποίο επιτρέπεται να τροποποιήσει τον πηγαίο κώδικα ενός έργου λογισμικού, που θα χρησιμοποιηθεί στις επίσημες εκδόσεις του έργου) έχουν πρόσβαση εγγραφής στο SVN(Apache Subversion είναι ένα σύστημα ελέγχου έκδοσης και αναθεώρησης λογισμικού που διανέμεται ως ανοιχτού κώδικα υπό την άδεια Apache) . και επομένως μπορούν να δεσμεύσουν τις ενημερώσεις κώδικα που υποβάλλονται από αυτούς τους συνεισφέροντες. Μερικές φορές το WordPress παραχωρεί προσωρινή πρόσβαση δέσμευσης σε άτομα που εργάζονται σε συγκεκριμένα στοιχεία. Ορισμένα άτομα έχουν μόνιμη πρόσβαση δέσμευσης. αποτελούν τη βασική ομάδα του WordPress.4 Η προώθηση από εξωτερικό συντελεστή σε μέλος της βασικής ομάδας βασίζεται κυρίως στην αξία. Αυτή η αξιοκρατία λαμβάνει υπόψη όχι μόνο τις τεχνικές δεξιότητες αλλά και τη στάση, τον επαγγελματισμό και το σεβασμό για τις βασικές φιλοσοφίες του έργου. Στην κορυφή της αλυσίδας, ο Mullenweg επιβλέπει τα πάντα υπό τον (ανεπίσημο) ρόλο του Καλοπροαιρετου Δικτάτορα για τη Ζωή.

3.1.2 Πλεονεκτήματα Wordpress

Στο κεφάλαιο αυτό ακολουθεί μια λίστα με μερικές από τις δυνατότητες που παρέχονται με το WordPress. Ωστόσο, υπάρχουν κυριολεκτικά χιλιάδες πρόσθετα που επεκτείνουν αυτό που κάνει το WordPress, επομένως η πραγματική λειτουργικότητα είναι σχεδόν απεριόριστη. Οι προγραμματιστές είναι ελεύθεροι να κάνουν ό,τι θέλουν με τον κώδικα του WordPress, να τον επεκτείνουν ή να τον τροποποιήσουν με οποιονδήποτε τρόπο ή να τον χρησιμοποιήσουν για εμπορικά έργα χωρίς τέλη αδειοδότησης. Αυτή είναι η ομορφιά του ελεύθερου λογισμικού. Ακολουθούν μερικά απο τα πλεονεκτήματα του Wordpress τόσο για τους χρήστες όσο και για τους προγραμματιστές.

- Ευελιξία: Δυνατότητα δημιουργίας διαφορετικών τύπων ιστοτόπου (blog ή ιστότοπο, ένα photoblog, έναν επιχειρηματικό ιστότοπο, ένα επαγγελματικό χαρτοφυλάκιο, έναν κυβερνητικό ιστότοπο, έναν ιστότοπο περιοδικών ή ειδήσεων κ.τλ) .
- Διαχείριση χρηστών: Δεν απαιτούν όλοι την ίδια πρόσβαση στον ιστότοπό. Η δυνατότητα διαχείρισης των ρόλων των χρηστών δίνει πολλές δυνατότητες.
- Search Engine Optimized To: To Wordpress δίνει τη δυνατότητα στον χρήστη να κάνει τον ιστότοπο του φιλικό προς τις μηχανές αναζήτησης.
- Πολυγλωσσικότητα: Μπορεί να μην παρέχονται πολλές γλώσσες για αυτόματη μετάφραση των βασικών στοιχείων wordpress αλλα προσφέρονται πολλά πρόσθετα που μπορούν να κάνουν τη διαδικασία αυτή αρκετά εύκολη, παραδείγματος χάριν το plug-in Loco Translate

- Σύστημα: Τα API του WordPress δίνουν τη δυνατότητα να δημιουργηθούν προσθήκες και να επεκτείνουν το WordPress.
- Σύστημα θεμάτων: Το WordPress παρέχει την επεκτασιμότητα για να δημιουργηθούν θέματα όσο απλά ή σύνθετα θέλετε.

3.1.2 Κερδίζοντας χρήματα με το WordPress

Το WordPress είναι μια επιχείρηση εκατομμυρίων και όλα δείχνουν αυτό να συνεχίσει να μεγαλώνει. Βασικός παράγοντας σε αυτή την ανάπτυξη βρίσκεται η τεράστια WordPress κοινότητα χρηστών.

Πέρα από την προσφορά όλων των ειδικών συμβουλευτικών υπηρεσιών (εγκατάσταση, διαμόρφωση, συντονισμός, μετεγκατάσταση κλπ.) για την εξυπηρέτηση της κοινότητας του WordPress, πολλοί άνθρωποι κατασκευάζουν και πουλούν πρόσθετα και θέματα. Τα πρόσθετα επεκτείνουν τον πυρήνα λειτουργικότητας WordPress. Το WordPress προσφέρει πολλά προκαθορισμένα πρότυπα τα οποία με προσθήκες και themes μπορούν να "κολλήσουν" για να παρέχουν τη λειτουργικότητά τους χωρίς την ανάγκη για τροποποίηση των βασικών αρχείων του WordPress.[20] Περίπου 2.000 πρότυπα είναι διαθέσιμα όπου κάθε ένα αντιστοιχεί σε ένα κοινό σημείο του WordPress (όπως η αποθήκευση, ανάρτηση, έγκριση ενός σχολίου ή η δημιουργία χρήστη). Το επίσημο αποθετήριο περιέχει περίπου 47.000 πρόσθετα, τα οποία έχουν ληφθεί περισσότερα από 600 εκατομμύρια φορές. Πρόσθετα (και θέματα) μπορεί να είναι δωρεάν, να πληρώνονται (μερικές φορές ως μέρος μιας συνδρομητικής υπηρεσίας) ή να ακολουθούν ένα μοντέλο freemium. Η ποιότητα των προσθηκών ποικίλλει πάρα πολύ και συνήθως, φταίνε όταν ο ιστότοπος WordPress παραβιάζεται. Όπως όλα τα σύνολα κανόνων, έτσι και οι οδηγίες προσπαθούν να είναι ακριβείς.

Αν και οι δημιουργοί των plug-ins και τα θέματα είναι κυρίως ανεξάρτητοι προγραμματιστές ή μικρές εταιρείες, οι μεγαλύτερες εταιρείες προσφέρουν υπηρεσίες φιλοξενίας WordPress με εναν πιο προβλέψιμο και επαναλαμβανόμενο μοντέλο εσόδων. Παρόλο που οι χρήστες θα μπορούσαν να εγκαταστήσουν το WordPress σε οποιοδηποτε πάροχο φιλοξενίας διαδικτύου, ορισμένοι παρέχουν πιο αποκλειστική υποστήριξη Ιστότοπων WordPress, που προσφέρουν, για παράδειγμα, ιστότοπους εγκατάστασης ή ενσωματωμένους συστήματα κρυφής μνήμης. [21]

3.1.3 Εξέλιξη Wordpress

Αναμφισβήτητα το WordPress έχει προχωρήσει πολύ από μια ταπεινή πλατφόρμα blogging στο ευέλικτο CMS που είναι σήμερα, αλλά θα πρέπει να συνεχίσει να εξελίσσεται αν θέλει να παραμείνει στην κορυφή. Η αγορά CMS είναι ελκυστική, με νέους ανταγωνιστές να εμφανίζονται κάθε χρόνο σε όλους τους τομείς του φάσματος CMS. Πολλοί από αυτούς προσπαθούν να γίνουν το καλύτερο CMS για συγκεκριμένα προφίλ πελατών ή τομείς, σε αντίθεση με τον στόχο του WordPress να είναι η λύση που ταιριάζει σε όλους.

Για την αντιμετώπιση αυτής της απειλής το Wordpress ανέπτυξε ένα Συμβούλιο Ανάπτυξης. Η πρωτοβουλία αυτή αφορά την αίτηση σε όλες τις εταιρείες που αναπτύσσονται στο WordPress να αφιερώσουν το 5 τοις εκατό των ανθρώπων τους για να συνεισφέρουν πίσω στον πυρήνα του WordPress — είτε αυτό ανάπτυξη, τεκμηρίωση, ασφάλεια, φόρουμ υποστήριξης, κριτικές θεμάτων, εκπαίδευση, δοκιμή, μετάφραση ή οτιδήποτε βοηθάει στην εξέλιξη του

WordPress. Από τεχνικής πλευράς, οι επόμενες εκδόσεις του WordPress αποστέλλονται με Gutenber, μια σημαντική αρχιτεκτονική μετατόπιση για το WordPress και η μεγαλύτερη προσπάθεια ανάπτυξης χαρακτηριστικών στο ιστορικό του WordPress. Ο Gutenberg στοχεύει να απλοποιήσει όλες τις προηγούμενες έννοιες του WordPress (για παράδειγμα, το μενού, τα γραφικά στοιχεία και οι σύντομοι κωδικοί) σε μια κομψή ιδέα: το μπλοκ.[22] Σύμφωνα με τον Mullenweg, ο Gutenberg θα είναι το μέλλον της γραφής, επεξεργασίας και προσαρμογής του WordPress τα επόμενα χρόνια. Όπως με κάθε κύρια αλλαγή, ο Gutenberg έχει φέρει μεγάλες αναταραχές στην κοινότητα του wordpress καθώς θα αναγκάζει πολλούς προγραμματιστές να κάνουμε μεγάλες αλλαγές στον κώδικα τους.

Ο στόχος του WordPress λοιπόν είναι να κυριαρχήσει στην αγορά CMS, από μεγάλες επιχειρήσεις μέχρι και σε μεμονωμένους bloggers με μικρή τεχνική γνώση. Ωστόσο, κάποιες κοινότητες χρηστών WordPress θα μπορούσαν να νιώσουν ότι το έργο εξελίσσεται σε μια κατεύθυνση που δεν εκπροσωπεί τις απόψεις τους και να αποφασίσουν να διαχωρίσουν το έργο και να δημιουργήσουν μια εξειδικευμένη έκδοση έτσι ώστε να ταιριάζει καλύτερα στις ανάγκες τους. Αυτός τελικά είναι ένας κίνδυνος που όλα τα ανοιχτού κώδικα έργα αντιμετωπίζουν και πρέπει να σκεφτούν.

Με τον ένα ή τον άλλο τρόπο, η ερευνητική κοινότητα έχει να συμβάλλει σημαντικά για το μέλλον του WordPress. Είναι εκπληκτικό ότι τόσο λίγα ερευνητικά άρθρα επικεντρώνονται στο WordPress, σε σύγκριση, για παράδειγμα, έγγραφα που αναλύουν Linux από κάθε δυνατή οπτική γωνία. Ο πλούτος και η σημασία της βάσης κώδικα του WordPress και του οικοσυστήματος θέτουν πολλές ενδιαφέρουσες προκλήσεις για την ερευνητική κοινότητα, ειδικά για ερευνητές που εργάζονται σε αποθετήρια λογισμικού εξόρυξης.

3.2 Joomla

Εικόνα 3: Λογότυπο Joomla!

Το Joomla! αποτελεί ένα από τα πιο δημοφιλή συστήματα διαχείρισης περιεχομένου (CMS) ανοιχτού κώδικα. Η πρώτη του κυκλοφορία ξεκίνησε το 2005 φτάνοντας σήμερα μετά από πολλές αναβαθμίσεις την έκδοση 4 αυξάνοντας την ταχύτητα και την ασφάλεια, φέρνοντας την βέλτιστη αναζήτηση στην εσωτερική πλοήγηση του ιστοτόπου και δημιουργώντας νέα template για την αποστολή emails.

Η προεπιλεγμένη εγκατάσταση του Joomla! είναι δηλαδή ήδη μια πολυλειτουργική διαχείριση περιεχομένου, αλλά εάν οι προεπιλεγμένες βασικές λειτουργίες δεν μπορούν να ανταποκριθούν στις απαιτήσεις του πελάτη, μπορεί εύκολα να το επεκτείνει με επεκτάσεις. Υπάρχουν πέντε τύποι επεκτάσεων διαθέσιμοι για το Joomla!: Στοιχεία, Ενότητες, Πρόσθετα, Πρότυπα και Γλώσσες. Κάθε μία από αυτές τις επεκτάσεις[23] ασχολείται με μοναδική λειτουργικότητα.

Ένα εξάρτημα είναι το μεγαλύτερο και πιο περίπλοκο από όλες τις επεκτάσεις του Joomla, επίσης μπορούν συχνά να αναφέρονται ως και μικροεφαρμογές.

Ένα εξάρτημα λειτουργεί σε δύο διαφορετικά μέρη, το μέρος του διαχειριστή και το τμήμα του ιστότοπου. Κάθε φορά που υπάρχει μια σελίδα Joomla φορτωμένη, ένα στοιχείο καλείται να αποδώσει το κύριο σώμα της σελίδας. Για παράδειγμα, το com_login είναι το στοιχείο που χειρίζεται διαδικασία σύνδεσης του χρήστη ο οποίος μπορεί να συνδεθεί στο σύστημα στο frontend του ιστότοπου που λειτουργεί με Joomla!, εάν είναι ήδη μέλος στο σύστημα. Κατα συνέπεια, τα εξαρτήματα αυτά έχουν εξαιρετική σημασία για ολόκληρο το σύστημα και σίγουρα μπορεί να χαρακτηριστεί ως το κύριο μέρος του συστήματός. [24]

Βασικά Χαρακτηριστικά

- Διαχείριση χρηστών
- Media Manager
- Διαχείριση Banner
- Επικοινωνήστε με τη Διαχείριση
- Δημοσκοπήσεις
- Αναζήτηση
- Διαχείριση συνδέσμων Ιστού
- Διαχείριση περιεχομένου
- Συνδικάτο και Διαχείριση ειδήσεων
- Διαχείριση προτύπων
- Ενσωματωμένο σύστημα βοήθειας
- Ισχυρή επεκτασιμότητα

3.2.1 Πρότυπα Κωδικοποίησης στο Joomla!

Το Joomla! περιλαμβάνει περισσότερες από 390 χιλιάδες γραμμές κώδικα (KLOC) που αποτελούνται κυρίως από κώδικα PHP που εξυπηρετεί αιτήματα HTTP κάνοντας ερώτημα σε μια βάση δεδομένων MySQL φτάνοντας έως και το 77% του συνόλου των γραμμών κώδικα.[25]

Η μορφή των αρχείων στο joomla πρέπει να είναι αποθηκευμένα ως κείμενο ASCII έχοντας όμως μερικά δοκιμαστικά αρχεία τα οποία δεν είναι ASCII. Οι χαρακτήρες είναι κωδικοποιημένοι σε UTF-8 και το Unix να είναι μορφοποιημένο σύμφωνα με τους ακόλουθους κανόνες. Οι γραμμές πρέπει να τελειώνουν μόνο με τροφοδοσία γραμμής (LF). Οι τροφοδοσίες γραμμής αντιπροσωπεύονται ως τακτική 10, οκταδική 012 και εξάγωνο 0A.

Η ορθογραφία των λέξεων και των όρων που χρησιμοποιούνται στα σχόλια κώδικα και στην ονομασία κλάσεων, συναρτήσεων, μεταβλητών και σταθερών θα πρέπει γενικά να είναι σύμφωνα με τους βρετανικούς αγγλικούς κανόνες (en_GB). Επιτρέπονται ορισμένες εξαιρέσεις, για παράδειγμα όταν χρησιμοποιούνται κοινά ονόματα προγραμματισμού που ευθυγραμμίζονται με το API της PHP ή άλλες καθιερωμένες συμβάσεις, όπως για το color πού είναι κοινή πρακτική η διατήρηση της αγγλικής ορθογραφίας των ΗΠΑ.

Οι καρτέλες χρησιμοποιούνται για την εσοχή κώδικα (όχι κενά όπως απαιτείται από το πρότυπο PEAR). Τα προγράμματα επεξεργασίας πηγαίου κώδικα ή τα Ενσωματωμένα Περιβάλλοντα

Ανάπτυξης (IDE) όπως το Eclipse πρέπει να έχουν τα στηλοθέτες για την εσοχή μέτρησης τεσσάρων (4) διαστημάτων σε μήκος.

Όσον αφορά το μήκος γραμμής δεν υπάρχει μέγιστο όριο για τα μήκη στα αρχεία, ωστόσο, συνιστάται μια πλασματική τιμή περίπου 150 χαρακτήρων για να επιτευχθεί καλό επίπεδο αναγνωσιμότητας χωρίς οριζόντια κύλιση. Επιτρέπονται μεγαλύτερες γραμμές εάν η φύση του κώδικα για μεμονωμένες γραμμές το απαιτεί και οι αλλαγές γραμμής θα είχαν αρνητική επίδραση στην τελική έξοδο (όπως για μικτά αρχεία διάταξης PHP/HTML).

3.2.2 Πλεονεκτήματα Joomla

Στη συνέχεια του κεφαλαίου αυτού θα αναλύσουμε μερικά από τα πλεονεκτήματα του συστήματος αυτού. [24]

- Ένα ευέλικτο σύστημα, εύκολο στην επέκταση και προσαρμογή: Το Joomla έχει χιλιάδες επαληθευμένες επεκτάσεις τρίτων και χιλιάδες διαθέσιμα πρότυπα υψηλής ποιότητας, πολλά από τα οποία είναι δωρεάν παρέχοντα συνεχόμενη υποστήριξη κάνοντας τελικά τη κατασκευή ιστοσελίδας αρκετά εύκολη.
- Βελτιστοποιημένη μηχανή αναζήτησης: Το Joomla διαθέτει ισχυρά εργαλεία SEO κατασκευασμένα και έτοιμα αμέσως
- Δωρεάν λογισμικό ανοιχτού κώδικα: Το Joomla δημιουργείται, συντηρείται και υποστηρίζεται από μια μοναδική εθελοντική κοινότητα που πιστεύει ότι πρέπει να είναι δωρεάν σε όλους, πάντα.
- Ασφάλεια: Η αποκλειστική ομάδα Security Strike του Joomla προσπαθεί πάντα να είναι μπροστά από την καμπύλη. Θα μιλήσουμε εκτεταμένα για το θέμα αυτό στα επόμενα κεφάλαια
- Πολυγλωσσικότητα: Με περισσότερα από 70 μεταφραστικά πακέτα διαθέσιμα για βασική υποστήριξη η διαδικασία την μετάφρασης γίνεται απλή και γρήγορη
- Οργανική ανάπτυξη: Το joomla μπορεί να προσαρμοστεί σε όλες τις ανάγκες των χρηστών
- Δυνατότητα δημιουργίας εφαρμογών: Σταθερό και ελαφρύ πλαίσιο PHP που επιτρέπει τη δημιουργία εφαρμογών web και γραμμής εντολών σε PHP

3.2.3 Joomla Cms

Το Joomla! CMS είναι ένα σύστημα που βασίζεται σε PHP για τη δημιουργία δυναμικών ιστοσελίδων. Η ευέλικτη και επεκτάσιμη δομή του Joomla! επιτρέπει την προσθήκη λειτουργιών με χρήση επεκτάσεων, για την αλλαγή της εμφάνισης (εμφάνιση και διάταξη) του περιεχομένου.

Όταν ένας χρήστης λοιπόν επιλέξει έναν σύνδεσμο, το Joomla συναρμολογεί τη σελίδα ανακτώντας το περιεχόμενο από τη βάση δεδομένων. Έπειτα με τη χρήση των αρχείων joomla template παίρνει οδηγίες για το πως θα εμφανίσει τα τελικά αποτελέσματα στην σελίδα. Τελικά η ολοκληρωμένη πληροφορία στέλνετε στο browser του χρήστη προκειμένου να δει το περιεχόμενο. Σχεδόν όλες οι πληροφορίες βρίσκονται στη βάση δεδομένων. Τα κείμενα, οι εικόνες, οι αποθηκευμένοι χρήστες, οι κωδικοί βρίσκονται όλα στη βάση δεδομένων. Ενα template στην πραγματικότητα είναι ένα σύνολο απο Css, PHP, HTML, XML, και αρχεία

εικόνων. Προκειμένου να αλλάξει κάτι απο αυτά θα χρειαστεί να γίνει επεξεργασία του template.

2.2.4 Joomla Platform

Σε αντίθεση με το Joomla CMS του οποίου ο στόχος είναι ένα ευρύ κοινό χρηστων, το Joomla platform στοχεύει στους προγραμματιστές. Η κυκλοφορία της πλατφόρμας Joomla στις 5 Ιουλίου 2011 [26] σηματοδότησε μια σημαντική αλλαγή στο Joomla! Χρησιμοποιώντας την πλατφόρμα οι προγραμματιστές μπορούν να κάνουν σχεδόν τα πάντα, ακόμα και να δημιουργήσετε το δικό τους CMS. Το platform είναι στην ουσία ένα περιεχόμενο απο /libraries ευρετηρίων. Τα ευρετήρια αυτά περιέχουν έναν μεγάλο αριθμό απο κλάσεις και μεθόδους μέσω της γραμμής εντολών.

Το Joomla! Το CMS ήταν αρχικά μια μονολιθική εφαρμογή. Ένας από τους στόχους του Joomla! ήταν να διαχωρίσει τη βάση κώδικα στο CMS με μια υποκείμενη βιβλιοθήκη, η οποία θα μπορούσε στη συνέχεια να επαναχρησιμοποιηθεί για άλλα έργα. Στην έκδοση 1.5 και 1.6, αυτός ο διαχωρισμός προετοιμάστηκε δημιουργώντας μια διαίρεση μεταξύ του Framework (CMS 1.5 και 1.6) (το οποίο δεν σχετίζεται με το νέο έργο Framework) και του Joomla! CMS Εφαρμογή. Το έργο της πλατφόρμας συνέχισε αυτή την προσπάθεια διαχωρίζοντας αυτό το πλαίσιο σε ένα ανεξάρτητο έργο (τόσο από άποψη κώδικα όσο και από άποψη οργάνωσης). Η πλατφόρμα λοιπόν είναι ένα ανεξάρτητο σύνολο βιβλιοθηκών που δεν εξαρτώνται από το Joomla! CMS. Αυτές οι βιβλιοθήκες αποτελούνται από βιβλιοθήκες που διατηρούνται από το Joomla! Έργο και βιβλιοθήκες που διατηρούνται από άλλους προγραμματιστές τρίτων.

3.3 Drupal

Το Drupal είναι ένα ανοιχτό κώδικα CMS γραμμένο σε PHP και χρησιμοποιεί MySQL, PostgreSQL ή MS SQL για βάση δεδομένων. Το Drupal μπορεί να είναι σε λειτουργικό σύστημα Linux, Windows ή MacIntosh. Η αρχιτεκτονική του Drupal έχει σχεδιαστεί με τέτοιο τρόπο ώστε τα τρία διαφορετικά στρώματα να λειτουργούν ανεξάρτητα και να συσχετίζονται το ένα με το άλλο για να δώσει την τελική έξοδο. Αυτά τα τρία στρώματα είναι το περιεχόμενο που αποτελεί τον ιστότοπο, τον αλγόριθμο εφαρμογής που οργανώνει αυτό το περιεχόμενο για παρουσίαση και το στρώμα αναπαράστασης που ενσωματώνεται από το Drupal θεματικό σύστημα. Η αρχιτεκτονική του λοιπόν βασίζεται στην Presentation Abstraction Control ή PAC.Δέχεται είσοδο μέσω μιας μόνο πηγής (HTTP GET και POST), δρομολογεί αιτήματα στις κατάλληλες βοηθητικές συναρτήσεις, εξάγει δεδομένα από την Abstraction (κόμβους και, από το Drupal 5 και μετά, σχηματίζει) και στη συνέχεια τα σπρώχνει μέσα από ένα φίλτρο για να πάρει ένα Παρουσίαση του (το σύστημα θεμάτων).

Πιο συγκεκριμένα η ιστοσελίδα που έρχεται στο πρόγραμμα περιήγησης του θεατή περνάει μια διαδοχική διαδικασία στην οποία οι μονάδες Drupal παίρνουν όλο το σχετικό περιεχόμενο από τις βάσεις δεδομένων και στη συνέχεια το θέμα ετοιμάζεται για την τελική παρουσίαση. Σε αντίθεση με τον Joomla, Η αρχιτεκτονική του Drupal δεν ακολουθεί το σχέδιο σχεδιασμού του MVC αλλά αντ 'αυτού ακολουθεί την παρουσίαση -abstractionControl (PAC).

Η τυπική έκδοση του, γνωστή ως Drupal core, έχει χαρακτηριστικά τα οποία περιλαμβάνουν εγγραφή και συντήρηση λογαριασμού χρήστη, διαχείριση μενού, ροές RSS, ταξινόμηση, προσαρμογή διάταξης σελίδας και διαχείριση συστήματος όπου είναι κοινά σε όλα τα συστήματα διαχείρισης περιεχομένου. Το Drupal τελικά μπορεί να λειτουργήσει ως ένας

απλός ιστότοπος, ένα ιστολόγιο ενός ή πολλών χρηστών, ένα φόρουμ στο Διαδίκτυο ή ένας ιστότοπος κοινότητας που παρέχει περιεχόμενο που δημιουργείται από τους χρήστες. Τέλος αν και το Drupal προσφέρει ένα εξελιγμένο API στους προγραμματιστές, η βασική εγκατάσταση στον ιστότοπο και η διαχείριση του πλαισίου δεν απαιτούν δεξιότητες προγραμματισμού έχοντας ως αποτέλεσμα την εύκολη διαχείριση του από πολλούς χρήστες.[27]

2.3.1 Ιστορία

Το Drupal γράφτηκε για πρώτη φορά απο τον Dries Buytaert ως πίνακας μηνυμάτων και έγινε έργο ανοιχτού κώδικα το 2001. Η πραγματική σημασία της λέξης είναι μια αγγλική εκδοχή της ολλανδικής λέξης druppel, που σημαίνει «σταγόνα». Το όνομα προήλθε απο τον ιστότοπο Drop.org ο οποίος πλέον είναι ανενεργός, και η εξέλιξη του κώδικα του έφτασε στο σημερινό Drupal. Η αρχική επιλογή του ονόματος του ήταν η λέξη "dorp" (Ολλανδικά σημαίνει "χωριό") για τις πτυχές της κοινότητας αλλά μετά από λάθος πληκτρολόγηση του Buytaert θεωρήθηκε οτι το λάθος ήταν καλύτερη απόδοση καθώς ακουγόταν πιο καλά.

Μετά την βοήθεια που προσέφερε το Drupal στη δημιουργία του "DeanSpace" για τον Χάουαρντ Ντιν, έναν από τους υποψηφίους στην προκριματική εκστρατεία του Δημοκρατικού Κόμματος των ΗΠΑ για τις προεδρικές εκλογές των ΗΠΑ, το 2003 η ζήτηση του αυξήθηκε αρκετά.

Το DeanSpace χρησιμοποίησε την κοινή χρήση ανοιχτού κώδικα του Drupal για να υποστηρίξει ένα αποκεντρωμένο δίκτυο περίπου 50 διαφορετικών, ανεπίσημων ιστοσελίδων υπέρ του Dean που επέτρεπαν στους χρήστες να επικοινωνούν απευθείας μεταξύ τους καθώς και με την καμπάνια. Αφού ο Dean τελείωσε την εκστρατεία του, τα μέλη της ομάδας του Ιστού συνέχισαν να επιδιώκουν το ενδιαφέρον τους για την ανάπτυξη μιας πλατφόρμας Ιστού που θα μπορούσε να βοηθήσει τον πολιτικό ακτιβισμό, εγκαινιάζοντας το CivicSpace Labs τον Ιούλιο του 2004, «...την πρώτη εταιρεία με εργαζόμενους πλήρους απασχόλησης που ανέπτυξε και διανομή της τεχνολογίας Drupal». Έπειτα όλο και περισσότερες εταιρείες ξεκίνησαν να ειδικεύονται στην ανάπτυκη εφαρμογών με τη χρήση του συστήματος Drupal φτάνοντας το 2013 να απαριθμεί εκατοντάδες προμηθευτές που πρόσφεραν υπηρεσίες σχετικές με το σύστημα Drupal. [28]

Από το 2014 το Drupal αναπτύσσεται από μια κοινότητα.[29] Από τον Ιούλιο του 2007 έως τον Ιούνιο του 2008 ο ιστότοπος Drupal.org παρείχε περισσότερες από 1,4 εκατομμύρια λήψεις λογισμικού Drupal, μια αύξηση περίπου 125% από το προηγούμενο έτος. Από τον Ιανουάριο του 2017 περισσότεροι από 1.180.000 ιστότοποι χρησιμοποιούν το Drupal. Αυτά περιλαμβάνουν εκατοντάδες γνωστούς οργανισμούς, συμπεριλαμβανομένων εταιρειών, μέσων ενημέρωσης και εκδοτικών εταιρειών, κυβερνήσεων, μη κερδοσκοπικών οργανισμών, σχολείων και ιδιωτών. Απο το 2015 έως και το 2021 το Drupal είχε την έκδοση 8.0.0 όπου μετά αντικαταστάθηκε τελείως από την έκδοση 9.0.0 η οποία όμως είχε πρώτη κυκλοφορία το 3 Ιουνίου 2020. Τέλος σε λίγους μήνες αναμένουμε την νέα έκδοση του την 10.0.0.

Από τον Ιανουάριο του 2017, το Drupal είχε γίνει διαθέσιμο σε 100 γλώσσες και Αγγλικά (η προεπιλογή) καθώς επίσης είναι από τα συστήματα που υποστηρίζουν και τα Αραβικά, τα Περσικά και τα Εβραϊκά όπου η γραφή τους είναι από τα δεξιά προς τα αριστερά. Η τοπική προσαρμογή Drupal είναι χτισμένη πάνω από το gettext, τη βιβλιοθήκη διεθνοποίησης και τοπικής προσαρμογής GNU (i18n). Το Drupal πριν την έκδοση 7 είχε λειτουργίες που εκτελούσαν εργασίες που σχετίζονται με βάσεις δεδομένων, όπως εκκαθάριση ερωτημάτων

SQL, πρόθεμα ονόματος πίνακα πολλών τοποθεσιών και δημιουργία κατάλληλων ερωτημάτων SQL. Η έκδοση 6 όμως πρόσθεσε ένα επίπεδο αφαίρεσης που επέτρεπε στους προγραμματιστές να δημιουργούν ερωτήματα SQL χωρίς να γράφουν SQL. Με την έκδοση 9 λοιπόν επεκτείνεται το επίπεδο αυτό προσφέροντας στους προγραμματιστές την ευελιξία να μην γραφουν ερωτήματα SQL ως συμβολοσειρές κειμένου χρησιμοποιώντας PHP Data Objects για να αφαιρέσει τη βάση δεδομένων

Από ένα τόσο διάσημο σύστημα διαχείρισης περιεχομένου δεν θα μπορούσαν να λείπουν τα θέματα (themes). [30] Το 2019 πάνω απο 2.800 δωρεάν θέματα είναι διαθέσιμα. Πολλά θέματα στο Drupal χρησιμοποιούν τυποποιημένες μορφές όπου πολλά ειναι γραμμένα στη μηχανή PHP Template. Ορισμένα πρότυπα χρησιμοποιούν ενσωματωμένη PHP . Απο την έκδοση Drupal 8 και έπειτα ενσωματώνεται η μηχανή προτύπων Twig .

2.3.2 Ασφάλεια

Το Drupal ανακοινώνει την κάθε ευπάθεια μόλις είναι διαθέσιμη η επιδιόρθωση της με τους διαχειριστές να ειδοποιούνται αυτόματα για τις νέες εκδόσεις.

Η πολιτική του Drupal είναι να ανακοινώνει τη φύση κάθε ευπάθειας ασφαλείας μόλις κυκλοφορήσει η επιδιόρθωση προκειμένου να αποφύγει τυχόν επιθέσεις εφόσον γίνει γνωστή η ευπάθεια. Οι διαχειριστές των τοποθεσιών του Drupal μπορούν να ειδοποιηθούν αυτόματα για αυτές τις νέες εκδόσεις μέσω της ενότητας Κατάσταση ενημέρωσης (Drupal 6) ή μέσω του Διαχειριστή ενημερώσεων (Drupal 7).

Το Drupal διατηρεί μια λίστα αλληλογραφίας ανακοινώσεων ασφαλείας, και γενικά ενα ιστορικό όλων των συμβάντων ασφαλείας. Στα μέσα Οκτωβρίου 2014, το Drupal εξέδωσε μια "άκρως κρίσιμη" συμβουλή ασφαλείας σχετικά με ένα σφάλμα έγχυσης SQL στο Drupal 7, γνωστό και ως Drupageddon. Η λήψη και η εγκατάσταση μιας αναβάθμισης στο Drupal 7.32 επιδιορθώνει την ευπάθεια, αλλά δεν αφαιρεί κανένα backdoor που έχει εγκατασταθεί από χάκερ, εάν ο ιστότοπος έχει ήδη παραβιαστεί. Οι επιθέσεις όπως ήταν αναμενόμενο άρχισαν αμέσως μετά την ανακοίνωση της ευπάθειας. Έχοντας αποτέλεσμα οι λογαριασμοί οι οποίοι δεν ενημερώθηκαν αμέσως μετά την ανακοίνωση να θεωρούνται παραβιασμένοι και να αναγκάζονται να αντικαταστήσουν την σελίδα τους με μια στατική HTML ιστοσελίδα έως ότου υπάρξει η δυνατότητα ενός backup πριν τις 15 Οκτωβρίου.

Στα τέλη Μαρτίου 2018, κυκλοφόρησε μια ενημέρωση κώδικα για την ευπάθεια CVE-2018-7600, [31] η οποία ονομάστηκε επίσης Drupalgeddon2 . Το υποκείμενο σφάλμα επιτρέπει σε απομακρυσμένους εισβολείς χωρίς ειδικούς ρόλους ή άδειες να αναλάβουν τον πλήρη έλεγχο των τοποθεσιών Drupal 6, 7 και 8. Το Drupal 6 έφτασε στο τέλος της ζωής του στις 24 Φεβρουαρίου 2016 και δεν λαμβάνει επίσημες ενημερώσεις ασφαλείας (η εκτεταμένη υποστήριξη είναι διαθέσιμη από δύο πληρωμένους προμηθευτές μακροπρόθεσμων υπηρεσιών). Από τις αρχές Απριλίου, παρατηρήθηκαν αυτοματοποιημένες επιθέσεις μεγάλης κλίμακας εναντίον ευάλωτων τοποθεσιών και στις 20 Απριλίου, αναφέρθηκε υψηλό επίπεδο διείσδυσης μη επιδιορθωμένων τοποθεσιών. Στις 23 Δεκεμβρίου 2019, το Drupal κατάφερε να επιδιορθώσει ένα σημαντικό αυθαίρετο σφάλμα μεταφόρτωσης αρχείων. Το ελάττωμα μεταφόρτωσης αρχείων επηρεάζει το Drupal 8.8.x πριν από το 8.8.1 και το 8.7.x πριν από το 8.7.11 και η ευπάθεια αναφέρεται ως μέτρια κρίσιμη από το Drupal. [32]

2.3.3 Πλεονεκτήματα Drupal

- Ευελιξία: Δυνατότητα προσθήκης επεκτάσεων
- Προσαρμόσιμο περιεχόμενο: Όλα τα στοιχεία του drupal είναι πλήρως προσαρμόσιμα
- Επεκτάσιμο περιεχόμενο: Μπορούν να δημιουργηθούν πολλών ειδών τύποι περιεχομένου (π.χ blogs)
- Έμφυτη βελτιστοποίηση μηχανών αναζήτησης
- Ρόλοι χρηστών

3.4 Σύγκριση συστημάτων CMS

Τα 3 συστήματα που αναλύσαμε σε προηγούμενες ενότητες μπορεί να αποτελούν διαφορετικές επιλογές αλλα έχουν μεταξύ τους και αρκετά κοινά σημεία. Μερικά από αυτά είναι:

- Ανοιχτού κώδικα
- Database: MyQSL
- Γλώσσα προγραμματισμού: PHP
- Δωρεάν Captcha (δυνατότητα αναγνώρισης αν ο χρήστης είναι άνθρωπος ή bot)
- Email Verification
- SSL certificate
- Ιστορικό διασύνδεσης (Το wordpress το προσφέρει με δωρεάν επιπρόσθετο(plugin)

Όσον αφορά το κόστός τους είναι δύσκολα να συγκριθεί καθώς υπάρχουν πολλά παράπλευρα κόστοι. Καθένα από αυτά τα τρία καλύτερα CMS είναι 100% δωρεάν έχοντας όμως πολλά επιπρόσθετα κόστοι, όπως είναι η φιλοξενία hosting και η αγορά όνομας (domain) τα οποία κοστίζουν το ίδιο και στις τρεις επιλογές. Έπειτα όμως παρόλο που η εγκατάσταση τους είναι δωρεάν, οι χρήστες επιθυμούν να κατεβάσουν πολλά επιπρόσθετα (plugins) τα οποία σε αρκετές περιπτώσεις κοστίζουν. Το wordpress έχει τις περισσότερες δωρεάν επιλογές αλλά σε κάποιες περιπτώσεις και τις πιο ακριβές. Επίσης το wordpress αποτελεί την πιο εύκολη από τις τρεις επιλογές καθώς προσφέρει μια απο τις πιο φιλικές διεπαφές προς τον χρήστη. Αντίθετα το Joomla! και το Drupal επικεντρώνονται πιο πολύ σε προγραμματιστές μη έχοντας τόσο ευέλικτα επιπρόσθετα.

Αδιαμφισβήτητα το wordpress είναι από τα πιο δημοφιλή cms κατακτόντας την κορυφή έχοντας πάνω από το 43% των ιστοσελίδων να έχουν δημιουργηθεί σε αυτό. [27]

4. Ευπάθειες συστημάτων CMS

Η ασφάλεια του διαδικτύου αποτελεί ένα θέμα που τα τελευταία χρόνια απασχολεί ολοένα και περισσότερο τους χρήστες. Ο φόβος των ευπαθειών που δημιουργούν οι hacker είναι μεγάλος καθώς ο χρόνος ανταπόκρισης των ειδικών για να προστατέψουν ένα σύστημα μερικές φορές είναι αρκετά μεγάλο έχοντας ως αποτέλεσμα την μη αποφυγή προβλημάτων ασφάλειας. Το πόσο ασφαλές θεωρείται ένα σύστημα δεν διασφαλίζεται μόνο την ώρα της ανάπτυξης του κώδικα. Προγραμματιστές, οργανισμοί, αναλυτές αλλά και οι απλοί χρήστες πρέπει να προφυλάσσονται συχνά από τυχόν επιθέσεις. Καθημερινά προκύπτουν νέα προβλήματα καθώς

οι ευπάθειες λογισμικού συνεχώς μεταβάλλονται διαρκώς καθιστώντας απαραίτητες αλλαγές στον κώδικα προκειμένου να διασφαλίσουν την μέγιστη προστασία από πιθανές επιθέσεις.

Η εφαρμογή του πρωτοκόλλου TCP / IP αποτέλεσε την αιτία των πρώτων επιθέσεων καθώς υπήρχαν αρκετά αυάλωντα σημεία. Με το πέρασμα των χρόνων οι ευπαθειες αυτές αντιμετωπίστηκαν μεταβάλλοντας τις επιθέσεις προς το επίπεδο εφαρμογής και κυρίως το διαδίκτυο καθώς η έκθεση του συστήματος προστασίας πολλών εταιρειών αποτελεί συχνό φαινόμενο. Τελικά καθώς η ασφάλεια των web servers βελτιώνεται συνεχώς, οι περισσότερες επιθέσεις πλέον πραγματοποιούνται στις web εφαρμογές.

Με τα χρόνια λοιπόν, η διαδικτυακή εφαρμογή έχει εξελιχθεί από ένα απλό, στατικό και μόνο για ανάγνωση σύστημα σε ένα πολύπλοκο, δυναμικό, και διαδραστικό σύστημα που παρέχει πληροφορίες και υπηρεσίες στους χρήστες. Οι διαδικτυακές εφαρμογές έχουν γίνει αναπόσπαστο μέρος της καθημερινής ζωής αφού είναι ελεύθερα διαθέσιμες και προσβάσιμες από οποιοδήποτε μηχάνημα μέσω του Διαδικτύου. Αυτες συχνά χειρίζονται ευαίσθητα δεδομένα τα οποία χρησιμοποιούνται για την εκτέλεση κρίσιμων εργασιών όπως τραπεζικές συναλλαγές, κοινωνικές συναναστροφές, διαδικτυακές αγορές και ηλεκτρονικές φορολογικές δηλώσεις. Ωστόσο, οι διαδικτυακές εφαρμογές έχουν γίνει πρωταρχικός στόχος για τους επιτιθέμενους λόγω της ευκολίας της χρήσης τους, της πανταχού παρουσίας, της ζήτησης και της αυξανόμενης βάσης χρηστών.

Οι ετερογενείς διαδικτυακές εφαρμογές αποτελούν στόχο για τους εισβολείς καθώς υλοποιούνται διαδικτυακές εφαρμογές από προγραμματιστές που εστιάζουν στην εφαρμογή των χαρακτηριστικών και τη λειτουργικότητα της εφαρμογής παρά την ασφάλεια. Ως αποτέλεσμα, οι υπάρχουσες διαδικτυακές εφαρμογές είναι περισσότερο ευάλωτο σε επιθέσεις και η εκμετάλλευση αυτών των τρωτών σημείων θέτει σε κίνδυνο την εμπιστευτικότητα, την ακεραιότητα και διαθεσιμότητα δεδομένων. Στη συνέχεια του κεφαλαίου αυτού θα περιγράψουμε μερικές απο τις πιο συχνές ευπάθειες των συστημάτων εφαρμογών.

4.1 Ευπάθειες των διαδικτυακών εφαρμογών

Μια ευπάθεια είναι ένα ελάττωμα στην εφαρμογή που προκύπτει από ελλείψεις στην κωδικοποίηση και προκαλεί σοβαρή ζημιά στην εφαρμογή κατά την εκμετάλλευση της. Αυτά τα τρωτά σημεία θα μπορούσαν να χρησιμοποιηθούν με την προσθήκη κακόβουλου κώδικα σε είσοδο που παρέχεται από έναν χρήστη για αλληλεπίδραση με την εφαρμογή. Ο κακόβουλος κώδικας μπορεί να παραβιάζει τους συντακτικούς και σημασιολογικούς περιορισμούς που επιβάλλονται στις εισόδους του χρήστη, να εκδίδει ερωτήματα σε ακατάλληλες καταστάσεις εφαρμογής και να τροποποιεί τις απαντήσεις HTTP για τη λήψη πληροφοριών των χρηστών. Η κακόβουλη είσοδος διαδίδεται μέσω της εφαρμογής λόγω ύπαρξης ελαττωμάτων εφαρμογής και οδηγεί σε επιθέσεις. Η πλειονότητα των επιθέσεων είναι πιθανές λόγω των ακόλουθων ελαττωμάτων εφαρμογής: ακατάλληλη επικύρωση εισόδου, ακατάλληλους μηχανισμούς ελέγχου ταυτότητας και εξουσιοδότησης, ακατάλληλη διαχείριση των πληροφοριών συνεδρίας και άλλα σφάλματα υλοποίησης που υπονομεύουν την προβλεπόμενη λειτουργία.

1. **Ακατάλληλη επικύρωση εισόδου**

   κακόβουλες εντολές που παραβιάζουν τη συντακτική δομή του SQL/XML ερωτήματος, εντολή OS κ.λπ. και ονομάζονται ευπάθειες Injection.Η ακατάλληλη επικύρωση εισόδου αναφέρεται στην απουσία επικύρωσης ή σε λανθασμένη

επικύρωση εισόδου που παρέχεται από έναν χρήστη μέσω της διεπαφής χρήστη της εφαρμογής. Αυτά τα ελαττώματα υλοποίησης επιτρέπουν στον εισβολέα να εισάγει

2. **Ακατάλληλοι μηχανισμοί ελέγχου ταυτότητας και εξουσιοδότησης**

   Οι ακατάλληλοι μηχανισμοί ελέγχου ταυτότητας και εξουσιοδότησης αναφέρονται σε εσφαλμένη εφαρμογή των λειτουργιών ελέγχου ταυτότητας και των πολιτικών ελέγχου πρόσβασης (ACP). Τα ελαττώματα αυτα επιτρέπουν στον εισβολέα να έχει πρόσβαση σε εμπιστευτικές ιστοσελίδες και να εκτελούνται μη εξουσιοδοτημένες ενέργειες στην εφαρμογή.

3. **Ακατάλληλη επιβολή της επιχειρηματικής λογικής**

   Η ακατάλληλη επιβολή της επιχειρηματικής λογικής αναφέρεται σε ελαττώματα λογικής που κάνουν την εφαρμογή να συμπεριφέρεται με διαφορετικό τρόπο από το επιδιωκόμενο, και οδηγεί σε οικονομική απώλεια, διαρροή πληροφοριών, υποβάθμιση της ποιότητας υπηρεσίας (QoS) κ.λπ. Τα ελαττώματα αυτά κερδίζουν την προσοχή των ερευνητών καθώς τις περισσότερες φορές οδηγούνται από οικονομικά κίνητρα.

4. **Ακατάλληλη διαχείριση περιόδων σύνδεσης**

   Η ακατάλληλη διαχείριση περιόδων σύνδεσης σχετίζεται με την αδυναμία στη δημιουργία και τον χειρισμό των διακριτικών συνεδρίας, η οποία είναι απαραίτητες για τη διατήρηση της ταυτότητας του τελικού χρήστη της εφαρμογής και τη χαρτογράφηση της σχέσης μεταξύ των διαδοχικών αιτήματα (δηλαδή για διατήρηση της κατάστασης) της αίτησης. Αυτά τα ελαττώματα επιτρέπουν στον εισβολέα να υπονομεύσει τη συνεδρία ενός έγκυρου χρήστη και να εκτελέσει αντίπαλες ενέργειες. Αυτά τα ελαττώματα ονομάζονται ευπάθειες διαχείρισης συνεδρίας. Στη συνέχεια του κεφαλαίου θα δούμε αναλυτικά τις ευπάθειες των συστημάτων εφαρμογής.

4.2 INJECTION FLAWS

Τα Injection flaws εμφανίζονται όταν ένας εισβολέας είναι σε θέση για να χειριστεί την τιμή των παραμέτρων εισαγωγής χρήστη που χρησιμοποιούνται ως μέρος ενός ερωτήματος, προκειμένου να αλλάξει η σύνταξη του ερωτήματος. Οι κακόβουλες παράμετροι όταν δεν επικυρώνονται σωστά μη ασφαλείς πληροφορίες ροής θέτουν σε κίνδυνο την ασφάλεια της εφαρμογής. Ετσι, η κύρια αιτία για την ευπάθεια στην Injection εισβολή αποτελεί η ανεπαρκής επικύρωση των ελεγχόμενων δεδομένων από τον χρήστη. Εκεί υπάρχουν πολλοί τύποι τρωτών σημείων έγχυσης σε εφαρμογές Ιστού και οι τύποι εξαρτώνται από το ερώτημα, την εντολή, ή η γλώσσα που γίνεται Injection. Αυτά περιλαμβάνουν ερωτήματα SQL, Αποκρίσεις HTML, Ελαφρύ πρωτόκολλο πρόσβασης καταλόγου δηλώσεις (Lightweight Directory Access Protocol - LDAP), εντολές λειτουργικού συστήματος, κεφαλίδες HTTP και πολλά περισσότερα. [33]

Μερικά από τα αποτελέσματα αυτών επιθέσων μπορεί να είναι:

- Να επιτρέπεται σε έναν εισβολέα να εκτελέσει κλήσεις λειτουργικού συστήματος σε μια μηχανή-στόχο
- Να επιτρέπεται σε έναν εισβολέα να παραβιάσει τα data υποστήριξης
- Να επιτρέπεται σε έναν εισβολέα να παραβιάσει συνεδρίες άλλων χρηστών

- Να επιτρέπεται σε έναν εισβολέα να εξαναγκάσει ενέργειες για λογαριασμό άλλων χρηστών ή υπηρεσιών

4.2.1 SQL injections

Οι ευπάθειες SQL Injection (SQLIV) είναι ελαττώματα που επιτρέπουν στον εισβολέα να παραβιάσει τη βάση δεδομένων της εφαρμογής με αποτέλεσμα την ανεπιθύμητη εξαγωγή/εισαγωγή δεδομένων από / στη βάση δεδομένων. Οι επιθέσεις αυτές ονομάζονται SQL Injection Attacks (SQLIA) και οι κύριοι λόγοι αυτών είναι η ακατάλληλη επικύρωση εισόδου από τον χρήστη, παραβίαση cookie και τροποποίηση μεταβλητών από την πλευρά του διακομιστή. Οι επιθέσεις τύπου injection μπορεί να είναι πολύ εύκολο να ανακαλυφθούν και να αξιοποιηθούν, αλλά μπορούν επίσης να είναι εξαιρετικά δυσνόητες. Σε κάθε περίπτωση, η χρήση των εξωτερικών κλήσεων είναι αρκετά διαδεδομένη, έτσι ώστε η πιθανότητα μια web εφαρμογή να παρουσιάσει ανάλογα σφάλματα θα πρέπει να θεωρείται υψηλό.[33]

4.2.2 Παράδειγμα επίθεσης

Έστω μια ιστοσελίδα περιέχει μια φόρμα συμπλήρωσης που ο χρήστης καλείται να πληκτρολογήσει ένα όνομα και ένα κωδικό χρήστη. Η συμπλήρωση του κωδικού θα δημιουργήσει ένα ερώτημα SQL προκειμένου να γίνει η ταυτοποίηση του χρήστη βάση των αποθηκευμένων στοιχείων. Ένα παράδειγμα αποτελεί το ακόλουθο ερώτημα.

SELECT Users.Username

FROM Users

WHERE Users.Username = 'Username'

AND Userst.Password = 'Password'

Εφόσον λοιπόν το ερώτημα αυτό επιστρέψει true ο χρήστης θα ταυτοποιηθεί και θα έχει πρόσβαση στο σύστημα. Παρόλα αυτά, εάν ο hacker εισάγει ένα έγκυρο όνομα χρήστη και

δώσει έναν έγκυρο αλλά όχι σωστό κωδικό("password' OR '1'='1"), στο πεδίο της φόρμας, τότε το ερώτημα που θα διαμορφωθεί όπως αναγράφεται στη συνέχεια.

SELECT Users.Username

FROM Users

WHERE Users.Username = 'Username'

AND Users.Password = 'password' OR '1'='1'

Εξαιτίας του γεγονότος ότι η συνθήκη '1'='1' είναι πάντα αληθές δίνει τελικά το σύστημα πρόσβαση στον χρήστη. Στο επόμενο κεφάλαιο θα αναλυθούν οι επιπτώσεις καθώς και λύσεις που μπορούν να προστατέψουν το σύστημα από τέτοιες επιθέσεις καθώς επίσης θα δοθεί ένα παράδειγμα κώδικα στο τελευταίο κεφάλαιο.

4.3 Cross-Site Scripting (XSS)

Το XSS είναι ένας τύπος ευπάθειας εισαγωγής κώδικα που επιτρέπει στον εισβολέα να εκτελέσει κακόβουλα σενάρια στο πρόγραμμα-πελάτη φυλλομετρητή (browser). Ο hacker βρίσκοντας την ευπάθεια σε κάποια εφαρμογή εκμεταλλεύονται την ιστοσελίδα προστίθοντας κακόβουλο υλικό. Όταν λοιπόν ένας χρήστης επισκέπτεται μια τέτοια ιστοσελίδα η οποία είναι υπό εκμετάλλευση από κάποιον hacker το πρόγραμμα περιήγησης εκτελεί κακόβουλα σενάρια. Το πρόγραμμα περιήγησης του τελικού χρήστη δεν έχει τρόπο να γνωρίζει ότι το σενάριο δεν πρέπει να είναι αξιόπιστο και θα εκτελέσει το σενάριο. Επειδή πιστεύει ότι το σενάριο προέρχεται από αξιόπιστη πηγή, το κακόβουλο σενάριο μπορεί να έχει πρόσβαση σε οποιαδήποτε cookie ή άλλες ευαίσθητες πληροφορίες που διατηρεί το πρόγραμμα περιήγησης και χρησιμοποιούνται με αυτόν τον ιστότοπο. Αυτά τα σενάρια μπορούν ακόμη και να ξαναγράψουν το περιεχόμενο της σελίδας HTML. Αυτή η εκμετάλλευση λοιπόν ονομάζεται επίθεση XSS. Η επίθεση XSS τελικά οδηγεί σε συνέπειες όπως πειρατεία συνεδρίας, διαρροή ευαίσθητων δεδομένων, κλοπή cookie και παραμόρφωση περιεχομένου ιστού. Στη συνέχεια θα αναλύσουμε τους τρεις τύπους επιθέσεων XSS. [34]

4.3.1 Τύποι επιθέσεων

Οι επιθέσεις XSS είναι τριών Τύπών: Reflected, Stored και DOM-based XSS.

4.3.2 Reflected XSS

Η Reflected XSS επίθεση είναι εκείνη όπου το script που έχει εισαχθεί αντικατοπτρίζεται απο τον διακομιστή ιστού, σαν ένα μήνυμα ειδοποίησης, ένα αποτέλεσμα αναζήτησης ή οποιοδήποτε άλλο αποτέλεσμα το οποίο αποτελεί μέρος ή το σύνολο των δεδομένων που αποστέλλονται στον διακομιστή ως μέρος του αιτήματος. Όταν λοιπόν ο χρήστης πειστεί και πατήσει τον σύνδεσμο που περιέχει το κακόβουλο υλικό ή να υποβάλει μια κακόβουλη φόρμα ή ακόμα και να περιηγηθεί στον ιστότοπο ο χρήστης είναι ευάλωτος καθώς ο κακόβουλος κώδικας αντανακλά πίσω στο πρόγραμμα του χρήστη καθώς ταξιδεύει στην ιστοσελίδα. Έπειτα, το πρόγραμμα περιήγησης του χρήστη εκτελεί τον κώδικα επειδή προήλθε από έναν "αξιόπιστο" διακομιστή ο οποίος στην ουσία τον εξαπάτησε.

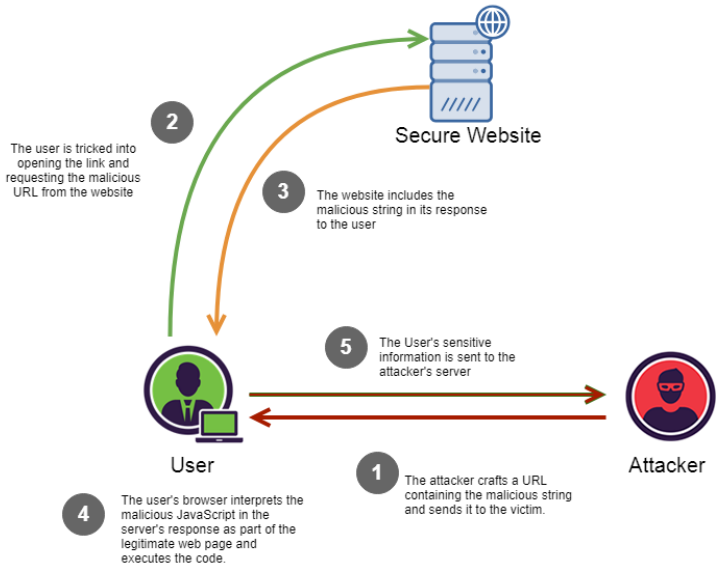
Εάν ένας εισβολέας καταφέρει να παραβιάσει τον ιστότοπο και να ελέγξει ένα σενάριο που εκτελείται στο πρόγραμμα περιήγησης του θύματος, τότε συνήθως μπορεί να παραβιάσει πλήρως αυτόν τον χρήστη. Μεταξύ άλλων, ο εισβολέας μπορεί:

- Να εκτελέσει οποιαδήποτε ενέργεια εντός της εφαρμογής που μπορεί να εκτελέσει ο χρήστης.
- Παραβίαση οποιασδήποτε πληροφορίας που μπορεί να δει ο χρήστης.
- Τροποποίηση οποιαδήποτε πληροφορία μπορεί να τροποποιήσει ο χρήστης.
- Δημιουργία νέων επιθέσων που θα φαίνεται ότι προέρχονται από τον αρχικό χρήστη-θύμα

Η επίθεση αυτή θα μπορούσε να στοχεύει απευθείας σε εναν συγκεκριμένο χρήστη η σε ένα σύνολο χρηστών.

Στη συνέχεια θα παραθέσουμε ένα παράδειγμα επίθεσης Reflected XSS

Ας υποθέσουμε ότι ένας ιστότοπος έχει μια λειτουργία αναζήτησης που λαμβάνει τον όρο αναζήτησης που παρέχεται από τον χρήστη σε μια παράμετρο URL: https://reflected-xss.com/search?term=wallets
Η εφαρμογή επαναλαμβάνει τον όρο αναζήτησης που παρέχεται στην απάντηση σε αυτήν τη διεύθυνση URL: Αναζητήσατε: wallets
Υποθέτοντας όμως η εφαρμογή δεν εκτελεί καμία άλλη επεξεργασία των δεδομένων, ένας εισβολέας μπορεί να κατασκευάσει μια επίθεση όπως αυτή:
https://reflected-xss.com/search?term=<script>/* Κανένα αποτέλεσμα*/</script>
Αυτή η διεύθυνση URL έχει ως αποτέλεσμα την ακόλουθη απάντηση:
Αναζητήσατε: <script>/* Κανένα αποτέλεσμα */</script>
Εάν κάποιος άλλος χρήστης της εφαρμογής ζητήσει τη διεύθυνση URL του εισβολέα, τότε το σενάριο που παρέχεται από τον εισβολέα θα εκτελεστεί στο πρόγραμμα περιήγησης του χρήστη θύματος, στο πλαίσιο της συνεδρίας του με την εφαρμογή. Πιο συγκεκριμένα όποιος χρήστης αναζητήσει στον ιστότοπο τα wallets το αποτέλεσμα που θα εμφανιστεί στην οθόνη είναι Αναζητήσατε: No results… αποτρέποντας τον χρήστη να δεί τα πορτοφόλια που έψαχνε. Σκοπός μιας τέτοιας επίθεσης θα μπορούσε να είναι είτε οικονομικός είτε δυσφήμισης καθώς ο hacker έχει την δυνατότητα προσθέσει οτι κείμενο επιθυμεί αν το σύστημα είναι ευπαθές.[34]



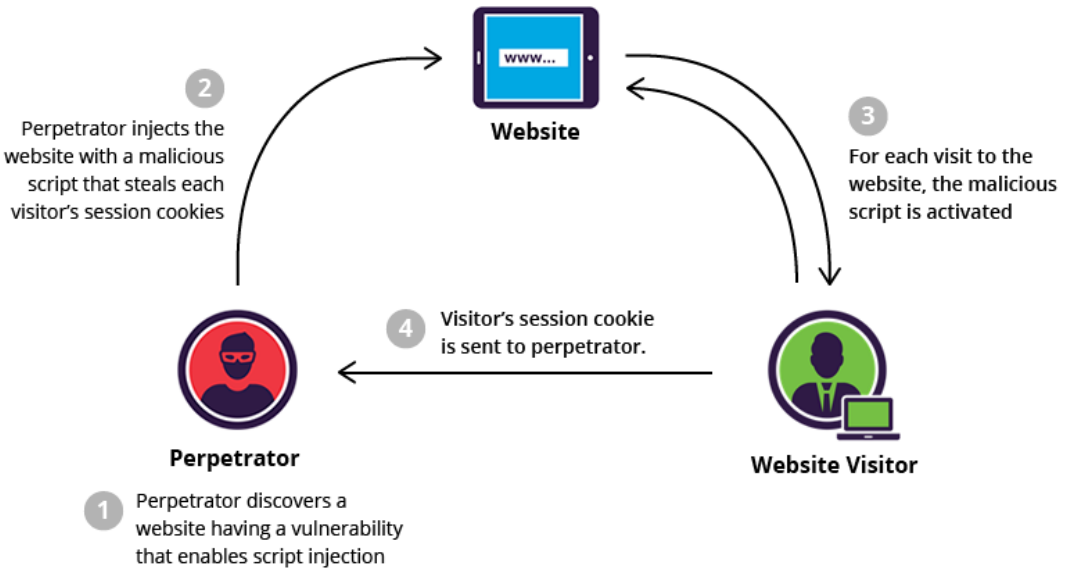
Εικόνα 4: Reflected XSS

4.3.3 Αποθηκευμένη επίθεση XSS (Stored)

Οι αποθηκευμένες επιθέσεις είναι εκείνες όπου το σενάριο που έχει εισαχθεί αποθηκεύεται μόνιμα στους διακομιστές-στόχους, όπως σε μια βάση δεδομένων, σε ένα φόρουμ μηνυμάτων, ένα αρχείο καταγραφής επισκεπτών, σε ένα πεδίο σχολίων κλπ. Στη συνέχεια, το θύμα ανακτά το κακόβουλο σενάριο από τον διακομιστή όταν ζητά τις αποθηκευμένες πληροφορίες. Το αποθηκευμένο XSS αναφέρεται επίσης μερικές φορές ως Persistent ή Type-I XSS. Η αποθηκευμένη επίθεση XSS (Stored) λαμβάνει χώρα κάθε φορά που μη επικυρωμένη είσοδος χρήστη που περιέχει κακόβουλο scripts αποθηκεύεται στη βάση δεδομένων της εφαρμογής. Τα αποθηκευμένα δεδομένα, όταν προσπελάζονται απο μια ιστοσελίδα, εκτοξεύουν μια επίθεση.

Για παράδειγμα σε μια ιστοσελίδα όπου επιτρέπεται η εισαγωγή σχολίων από τους χρήστες και η αποθήκευση τους στη βάση ένας κακόβουλος χρήστης μπορεί να το εκμεταλλευτεί προσθέτοντας στα σχόλια το εξής script: <script>alert(΄XSS΄)</script> . Έπειτα όταν ο χρήστης φορτώσει το σημείο που βρίσκεται το script θα εκτελεστεί και ο κώδικας του. Τότε θα εμφανιστεί το μήνυμα xss στην σελίδα του χρήστη.

Αυτοί οι δύο τύποι τρωτών σημείων προκύπτουν λόγω ακατάλληλης επικύρωσης των εισόδων από τον χρήστη στην πλευρά του διακομιστή δίνοντας την ευκαιρία στον κακόβουλο χρήστη να εισβάλει στον ιστότοπο. [34]



Εικόνα 5: Stored XSS

4.3.4 Επίθεση XSS με βάση το DOM

Η επίθεση XSS με βάση το DOM λαμβάνει χώρα στην πλευρά του πελάτη της εφαρμογής . Η επίθεση κάνει το σενάριο από την πλευρά του πελάτη να συμπεριφέρεται με απρόβλεπτο τρόπο, όταν το σενάριο χρησιμοποιεί μη επικυρωμένες πληροφορίες από τη δομή DOM (μοντέλο αντικειμένου εγγράφου) για επεξεργασία στην αίτηση. Οταν δηλαδή μια εφαρμογή περιέχει κάποια JavaScript από την πλευρά του πελάτη που επεξεργάζεται δεδομένα από μια μη αξιόπιστη πηγή με μη ασφαλή τρόπο, συνήθως γράφοντας τα δεδομένα πίσω στο DOM.

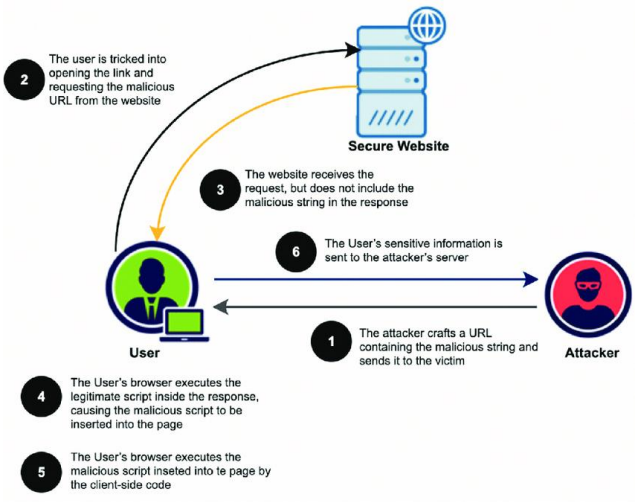
Στη συνέχεια θα παραθέσουμε ένα παράδειγμα επίθεσης DOM XSS
Οι γραμμές JavaScript που ακολουθούν χρησιμοποιούνται για να διαβάσουν την τιμή από μια εισαγωγή και έπειτα να την εμφανίσει εντός του HTML:

```
var search = document.getElementById('search').value;

var results = document.getElementById('results');

results.innerHTML = 'Αναζητήσατε: ' + search;
```

Εφόσον λοιπόν ο εισβολέας μπορεί να εισάγει την τιμή, μπορεί εύκολα να δημιουργήσει μια κακόβουλη τιμή που προκαλεί την εκτέλεση του δικού του σεναρίου:
You searched for: <img src=1 onerror='/* Κανένα αποτέλεσμα. */'>



Εικόνα X: DOM XSS

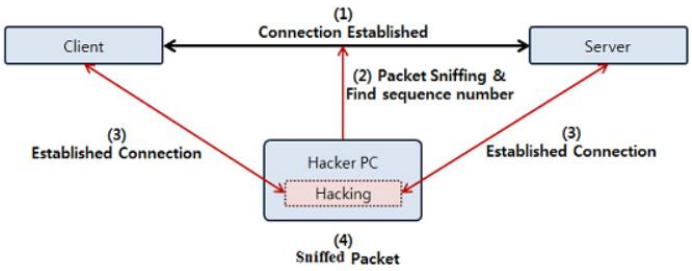
4.4 Broken Authentication and Session Management

Αναμφισβήτητα προκειμένου να υλοποιηθεί ένα σύστημα το οποίο θα θεωρείται ασφαλές απαραίτητη προϋπόθεση αποτελεί η κατάλληλη πιστοποίηση της ταυτότητας του χρήστη πριν την εισαγωγή του στο σύστημα. Ένα αίτημα συνεδρίας εγείρεται από έναν χρήστη μιας διαδικτυακής εφαρμογής μέσω της σελίδας σύνδεσης όπου βρίσκεται το διαπιστευτήριο χρήστη υπό την προϋπόθεση μόλις σταλεί το δεδομένο αίτημα από τον πελάτη από την πλευρά του διακομιστή, ο διακομιστής ξεκινά ένα ερώτημα στη βάση δεδομένων για να ελέγξετε εάν τα διαπιστευτήρια που παρέχονται από τον χρήστη αντιστοιχούν στην εγγραφή της βάσης δεδομένων ή όχι. Μόλις η διαδικασία επιτευχθεί, θα πραγματοποιηθεί μια περίοδος σύνδεσης με συγκεκριμένο αναγνωριστικό το οποίο διατίθεται στον χρήστη για την επικοινωνία της εφαρμογής. Ένας χρήστης τότε μπορεί να έχει πρόσβαση στο σύστημα με δεδομένα που παρέχονται από το διαχειριστή του συστήματος για τη λήψη διαφορετικών υπηρεσιών.

Μια έγκυρη περίοδος λειτουργίας λειτουργεί για μια ορισμένη διάρκεια που είναι προκαθορισμένη από τον σχεδιαστή του συστήματος. Τα προγράμματα περιήγησης αποθηκεύουν τον χρήστη έτσι ώστε η περίοδος λειτουργίας θα συνεχίσει μόλις λήξει η περίοδος της συνεδρίας αποστολή των πληροφοριών ελέγχου ταυτότητας στην πλευρά του διακομιστή. Αυτη η διαδικασία εκτελείται αυτόματα πίσω από τη διεπαφή χρήστη γεγονός που θα μειώσει την προσπάθεια του χρήστη για έλεγχο ταυτότητας. Ωστόσο, ο εισβολέας μπορεί να πιάσει και να αποκτήσει πρόσβαση στην ενεργή συνεδρία του άλλου χρησιμοποιώντας διαφορετικές εφαρμογές όπως, διαχειριστής cookie, eat my cookie, advanced cookie manager+, κλπ., σε περίπτωση που ο χρήστης παρέλειψε να κλείσει τη συνεδρία σύμφωνα με τις οδηγίες του ο σχεδιαστής της εφαρμογής.

Μετά την ολοκλήρωση μιας διαδικασίας ελέγχου ταυτότητας, θα πραγματοποιηθεί μια περίοδος λειτουργίας η οποία θα ενεργοποιηθεί για επικοινωνία δεδομένων μεταξύ του

διακομιστή και ενός συγκεκριμένου χρήστη. Εάν κάποιος εισβολέας μπορεί να αποκτήσει πρόσβαση στο προφίλ οποιουδήποτε συγκεκριμένου χρήστη παρακάμπτοντας τον έλεγχο ταυτότητας, το σενάριο αντιμετωπίζεται ως κατεστραμμένο και θεωρείται ως μια ευπάθεια ελέγχου ταυτότητας της δεδομένης εφαρμογής. Η ευπάθεια αυτή θα οδηγήσει σε ολική πρόσβαση των λογαριασμών των χρηστών έχοντας ως αποτέλεσμα την πρόσβαση των κακόβουλων χρηστών σε προσωπικές πληροφορίες παραβιάζοντας την προσωπική ζωή των χρηστών. Ο αντίκτυπος μιας τέτοιας εισβολής θα ήταν σοβαρός καθώς ο εισβολέας μπορεί να συνδεθεί στο λογαριασμό ως κανονικός χρήστης.



Εικόνα 6: Broken Authentication and Session Management

Η ευπάθεια παραβίασης της ταυτοποίησης του χρήστη είναι αρκετά συχνή, καθώς το πρόβλημα δημιουργείται απο την προσθήκη επιπροσθέτων με σκοπό την διαχείριση των κωδικών πρόσβασης όπως είναι η λήξη τους, η αυτόματη αποσύνδεση του χρήστη μετά από πιθανή αδράνεια ακόμα και η δυνατότητα προσθήκης μυστικών ερωτήσεων σε περίπτωση που ο χρήστης ξεχάσει τον κωδικό του. [35]

Υπάρχουν ορισμένες τεχνικές εκμετάλλευσης που χρησιμοποιούνται για την εκμετάλλευση Κατεστραμμένος έλεγχος ταυτότητας και διαχείριση περιόδου λειτουργίας(Broken Authentication and Session Management). Τύποι του τα παραπάνω δίνονται παρακάτω:

Α. Γενικός έλεγχος ταυτότητας και διαχείριση περιόδων λειτουργίας

Η μέθοδος δοκιμής χειροκίνητης διείσδυσης χρησιμοποιείται για τον έλεγχο της παραπάνω ευπάθειας των εφαρμογών web. Στέλνει συνεχώς το αίτημα του παραγόμενου χρήστη μέχρι το σύστημα να το βρει σωστό. Μόλις τα διαπιστευτήρια αντιστοιχίζονται με τη βάση δεδομένων, το σύστημα στέλνει μια απάντηση στον εισβολέα με την πρόσβαση στον λογαριασμό ή τον πίνακα διαχείρισης. Αναφέρεται εδώ ότι υπάρχουν πολλά συστήματα εύκολα εκμεταλλεύσιμο λόγω της χρήσης των αδύναμων κωδικών πρόσβασης όπως admin admin, admin123, κ.λπ.[36]

Β. Τεχνικές Εκμετάλλευσης

Οι τεχνικές εκμετάλλευσης διαχείρισης συζητούνται παρακάτω.

Α) Επίθεση εσφαλμένης διαμόρφωσης περιόδου λειτουργίας:

Η διάρκεια της συνεδρίας είναι ένα από τα σημαντικότερα γεγονότα στη διατήρηση ενός ασφαλούς ελέγχου ταυτότητας των διαδικτυακών εφαρμογών. Μόλις τα διαπιστευτήρια χρήστη επικυρώνονται από ένα σύστημα, εκχωρείτε μια συνεδρία για το συγκεκριμένο χρήστη με αναγνωριστικό περιόδου σύνδεσης για περιορισμένο χρονικό διάστημα. Σε περίπτωση που ο προγραμματιστής της εφαρμογής Ιστού ορίσει τη παράμετρο της διάρκειας της περιόδου

σύνδεσης με μεγάλη τιμή, η συνεδρία θα παραμείνει ενεργή για τη συγκεκριμένη περίοδο εάν ο χρήστης δεν αποσυνδεθεί από τον λογαριασμό του. Επομένως, αυτή η περίοδος λειτουργίας μπορεί να αποκατασταθεί για επαναχρησιμοποίηση από έναν εισβολέα ο οποίος θα οδηγήσει σε Broken Authentication συνεδρία Η εσφαλμένη διαμόρφωση είναι ένας από τους πιο κρίσιμους τομείς για τη Broken Ευπάθεια ελέγχου ταυτότητας και διαχείρισης περιόδου σύνδεσης. Ο εισβολέας χρησιμοποιεί το πρόγραμμα περιήγησης, το Google dork και τη μη ανακατεύθυνση για παράκαμψη του πίνακα διαχείρισης κατά τη διάρκεια της συνεδρίας διαδικασία εκμετάλλευσης εσφαλμένης διαμόρφωσης. Οι διαδικασίες εκμετάλλευσης εσφαλμένης διαμόρφωσης περιγράφονται στα τέσσερα βήματα.

Βήμα 01: Ο εισβολέας χρησιμοποιεί το Google dork για αναζήτηση ευάλωτων ιστού τοποθεσίες π.χ. inurl:apanel/admin/;

Βήμα 02: Η Google επιστρέφει τη λίστα των πιθανών ευάλωτων ιστών λίστα τοποθεσιών

Βήμα 03: Ο εισβολέας επιλέγει τις συγκεκριμένες διευθύνσεις URL με αρχείο ευρετηρίου όπως, inurl:apanel/admin/index.php. Το πρόγραμμα περιήγησης στέλνει ένα αίτημα στον διακομιστή για πρόσβαση στον Πίνακα διαχειριστή χρήστη απευθείας χωρίς τη χρήση ονόματος χρήστη και Κωδικό πρόσβασης. Το σενάριο Java ανακατευθύνει το αίτημα και την αποστολή εισβολέα στη σελίδα σύνδεσης.

Βήμα 04: Τα πρόσθετα Noredirect στην firefox βοηθούν τον εισβολέα να παρακάμψει τον πίνακα διαχείρισης με επιτυχία, και ο εισβολέας αποκτά το προνόμιο της πρόσβασης στο Σύστημα.

Β) Μη ασφαλείς κωδικοί πρόσβασης

Λόγω έλλειψης ενημέρωσης των χρηστών σχετικά με τη διαχείριση κωδικών πρόσβασης, ορισμένοι μη τεχνικοί χρήστες διατηρούν τον κωδικό πρόσβασής τους σε γενική μορφή φόρμα όπως admin, password, mypassword, password123, admin1997 κ.λπ. και επίσης σε ορισμένες περιπτώσεις, ο χρήστης κρατάει τον προεπιλεγμένο κωδικό πρόσβασης για την πρόσβασή του στο σύστημα ο οποίος θα είναι εύκολο να μαντέψει καποιος εισβολέας και να αποκτήσει πρόσβαση στο σύστημα.

Γ) Εκμετάλλευση του προβλήματος ελέγχου ταυτότητας.

Ο χειρισμός των συστημάτων ελέγχου ταυτότητας εφαρμογών Ιστού γίνεται χρησιμοποιώντας ερωτήματα υπό όρους για να γίνει έλεγχος του ονόματος του χρήστη και του κωδικού πρόσβασης έναντι άλλων χρηστών για έλεγχο ταυτότητας. Εάν αυτά υπό όρους τα ερωτήματα μολύνονται ή δεν αντιμετωπίζονται σωστά, θα μπορούσε εύκολα παραβιάζεται από έναν εισβολέα για να αποκτήσει πρόσβαση στο σύστημα χωρίς τον κατάλληλο έλεγχο ταυτότητας.

Δ) Αποκωδικοποίηση Ανεπαρκούς Κρυπτογράφησης

Σε ορισμένες εφαρμογές Ιστού τα μέτρα απορρήτου δεν χειρίζονται σωστά από τους προγραμματιστές. Επομένως, ένας hacker μπορεί να κλέψει το αναγνωριστικό περιόδου σύνδεσης από έναν χρήστη εκμεταλλευόμενος το ελαττώματα ασφαλείας της αποκάλυψης του αναγνωριστικού περιόδου σύνδεσης στη διεύθυνση URL του σύστημα, π.χ. http://www.broken-authentication.com/id=78923&name=maria. Το παράδειγμα δείχνει το αναγνωριστικό περιόδου λειτουργίας της γενικής συναλλαγής maria η οποία έχει αποκαλυφθεί

δημόσια στη διεύθυνση URL. Ως εκ τούτου, αυτό δεν είναι πολύ σημαντικό για έναν εισβολέα να κλέψει το αναγνωριστικό περιόδου σύνδεσης κάποιου άλλου χρήστη, Απλώς αλλάζει μόνο την τιμή του αναγνωριστικού περιόδου σύνδεσης στο URL. Η διαδικασία επίθεσης είναι εφικτή για τις ανεπαρκείς κρυπτογράφησεις στην τιμή του αναγνωριστικού περιόδου σύνδεσης. Μετά την αλλαγή του στη τιμή στο αναγνωριστικό περιόδου σύνδεσης, θα μοιάζει όπως παρακάτω:

http://www.broken-authentication.com/id=78923&name=attackername

Ποια σενάρια μπορεί να προκαλέσουν σπασμένο έλεγχο ταυτότητας;

Στις κατεστραμμένες επιθέσεις ελέγχου ταυτότητας, ο στόχος του χάκερ είναι να συμβιβαστεί ή να αποκτήσει πρόσβαση στους λογαριασμούς του θύματος. Αυτό μπορεί να συμβεί λόγω μιας σειράς σεναρίων όπως:

- Τα διαπιστευτήρια χρήστη δεν προστατεύονται όταν αποθηκεύονται όπως στη βάση δεδομένων.
- Τα διαπιστευτήρια σύνδεσης είναι προβλέψιμα ή χρησιμοποιούνται προεπιλεγμένα διαπιστευτήρια.
- Τα διακριτικά περιόδου σύνδεσης αποκαλύπτονται σε διευθύνσεις URL.
- Τα διακριτικά περιόδου λειτουργίας είναι ευάλωτα σε επιθέσεις καθήλωση περιόδου λειτουργίας.
- Οι περίοδοι σύνδεσης χρήστη δεν ακυρώνονται σωστά μετά την αποσύνδεση ή η περίοδος λειτουργίας δεν λήγει μετά από μια ορισμένη διάρκεια.
- Τα διαπιστευτήρια και οι πληροφορίες συνεδρίας μεταδίδονται μέσω καθαρού κειμένου χρησιμοποιώντας κρυπτογραφημένα πρωτόκολλα.

4.5 CROSS SITE REQUEST FORGERY

Η πλαστογράφηση αιτημάτων μεταξύ τοποθεσιών είναι ο τύπος επίθεσης όταν ένας εισβολέας αναγκάζει το πρόγραμμα περιήγησης ιστού του θύματος να εκτελέσει μια ανεπιθύμητη ενέργεια σε έναν αξιόπιστο ιστότοπο χωρίς την αλληλεπίδραση του χρήστη για αυτή την ενέργεια. Αυτή η επίθεση εκμεταλλεύεται την εμπιστοσύνη ενός ιστότοπου στο πρόγραμμα περιήγησης του χρήστη.

Η επίθεση CSRF θεωρείται σοβαρή απειλή για διαδικτυακές εφαρμογές όπου βασίζεται στις ευπάθειες που υπάρχουν στην κανονική απόκριση αιτήματος του πρωτοκόλλου HTTP. Είναι δύσκολο να εντοπιστεί και ως εκ τούτου, υπάρχει στο μεγαλύτερο μέρος των εφαρμογών του υπάρχοντος ιστού. Η CSRF ωστόσο συζητείται λιγότερο στις τάξεις ασφαλείας. Οι περισσότεροι προγραμματιστές ιστού δεν γνωρίζουν για αυτήν την επίθεση η οποία συγχέεται με το XSS (Cross site scripting). Οι περισσότεροι από αυτούς που γνωρίζουν για το CSRF πιστεύουν ότι η προστασία έναντι του XSS αρκεί και στη περίπτωση CSRF. Αλλά το CSRF και το XSS δεν είναι το ίδιο πράγμα. Σε αυτό το κεφάλαιο παρουσιάζουμε τον τρόπο ευπάθειας αυτού του ιστότοπου.

Μια επίθεση CSRF περιλαμβάνει 3 πράγματα. Αυτά είναι οι χρήστες-θύμα, έναν αξιόπιστο ιστότοπο και έναν κακόβουλο ιστότοπο. Ο χρήστης θύμα κρατά μια ενεργή συνεδρία με έναν αξιόπιστο ιστότοπο και ταυτόχρονα επισκέπτεται και έναν κακόβουλο ιστότοπο. Ο κακόβουλος ιστότοπος εισάγει ένα HTTP αίτημα για τον αξιόπιστο ιστότοπο στη συνεδρία χρήστη θύματος θέτοντας σε κίνδυνο την ακεραιότητά του. Για παράδειγμα ας υποθέσουμε ότι υπάρχει ένας χρήστης που βρίσκεται σε έναν ιστότοπο Α. Αφού συνδεθεί στον ιστότοπο, θα εργαστεί σε αξιόπιστη σελίδα. Τώρα χωρίς να αποσυνδεθεί από το ιστοσελίδα, θέλει επίσης να επισκεφτεί έναν ιστότοπο Β. Αλλά στον ιστότοπο Β, ο εισβολέας έχει ήδη μεταβεί σε έναν κακόβουλο σύνδεσμο στον ιστότοπο Β όπου και το κακόβουλο περιεχόμενο πλεον μπορεί να ζητήσει ένα HTTP αίτημα από τον ιστότοπο Α ώστε να εκτελέσει κάποια έγκυρη ενέργεια όπου απαιτείται έγκυρη συνεδρία. Όταν ο χρήστης επισκέπτεται λοιπόν την ιστοσελίδα Β και κατά λάθος κάνει κλικ στον κακόβουλο σύνδεσμο, το αίτημα HTTP αποστέλλεται στον ιστότοπο Α που χρησιμοποιεί την έγκυρη συνεδρία του χρήστη για να εκτελέσει κάποια έγκυρη ενέργεια στο ιστοσελίδα Α. Εδώ το κύριο κόλπο του εισβολέα είναι να δημοσιεύσει τον κακόβουλο σύνδεσμο και να προσελκύσει το θύμα να κάνει κλικ σε αυτόν τον σύνδεσμο. Η δημοσίευση κακόβουλου συνδεσμου στον ιστότοπο περιλαμβάνει κυρίως ετικέτα εικόνας. <img src="http: //ethniki-bank.com/transfer.php?account=sender& amount=10&name=maria">.

Τα αποτελέσματα των επιθέσεων CSRF ενδέχεται να διαφέρουν ανάλογα με τα τρωτά σημεία και τα προνόμια του χρήστη που εκμεταλλεύονται οι hackers. Μια επιτυχημένη επίθεση CSRF μπορεί να θέσει σε κίνδυνο τα δεδομένα και τις λειτουργίες του τελικού χρήστη όταν στοχεύει έναν κανονικό χρήστη, για παράδειγμα μεταφορά ποσού από λογαριασμό χρήστη σε λογαριασμό εισβολέα. Εάν ο στοχευμένος τελικός χρήστης είναι ο λογαριασμός διαχειριστή, μια επίθεση CSRF μπορεί να θέσει σε κίνδυνο ολόκληρη την εφαρμογή Ιστού. Δεν κινδυνεύουν μόνο οι δημόσιες εφαρμογές Ιστού σας, αλλά οι τακτικές CSRF μπορούν να χρησιμοποιηθούν για επίθεση σε διακομιστές πίσω από εταιρικά τείχη προστασίας. Η παρακάτω ετικέτα εικόνας δείχνει ένα τέτοιο παράδειγμα.

<img src="http://intranet/admin/purgedatabase?rowslike=%2A&confirm=yes">

Εάν ο εισβολέας πρέπει γνωρίζει αρκετά για να δημιουργήσει μια διεύθυνση URL και μπορέσει να πείσει τον χρήστη / θύμα να ανοίξει το κακόβουλο μήνυμα. Σε τέτοιες περιπτώσεις το CSRF μπορεί να οδηγήσει σε τεράστια απώλεια στην εφαρμογή του χρήστη καθιστώντας αναγκαία την αντιμετώπιση της επίθεσης αυτής. Το CSRF μπορεί να επηρεάσει τις συσκευές Ιστού όπως και τους ιστότοπους. Για παράδειγμα, έστω ότι οι εισβολείς έστειλαν e-mail έχοντας ενσωματωμένο αίτημα σε ετικέτα εικόνας με URI 192.168.1.1 που είναι η προεπιλεγμένη διεύθυνση IP του web, τότε ο δρομολογητής που βασίζεται σε Linux, εάν η διεπαφή ιστού είναι ευάλωτη στο CSRF και ο έλεγχος ταυτότητας είναι επίσης ευάλωτος τότε η επίθεση ενεργοποιείται. Στη συνέχεια, κατά το άνοιγμα της ετικέτας εικόνας email, φορτώνεται και οι εντολές μπορούν να εκτελεστούν στο δρομολογητή του λογαριασμού email του κατόχου. Η παρακάτω διεύθυνση URL δείχνει τον τρόπο εντολής του κελύφους μπορεί να εκτελεστεί σε δρομολογητή. Σε αυτή την ενότητα θα κάνουμε μια ανασκόπηση ευπαθειων που παρουσιάζονται σε εφαρμογές web. [37]

4.5.1 Μηχανισμός χειρισμού συνεδρίας HTTP

Μεγάλος αριθμών ιστοσελίδων απαιτούν έλεγχο ταυτότητας χρήστη ώστε να αποκτήσουν πρόσβαση σε αυτό. Η απαίτηση αυτή είναι απόλυτα σημαντική καθώς εκτελεί εργασίες ειδικές για τον χρήστη και παροχή απορρήτου στα δεδομένα και τις πληροφορίες του χρήστη. Προκειμένου να απλοποιηθεί αυτή απαίτηση το πρωτόκολλο HTTP παρέχει τη δυνατότητα συνεδρίας και cookie, τα οποία επιτρέπουν στον διακομιστή ιστού να διαφοροποιεί το αίτημα που προέρχεται από διαφορετικούς χρήστες. Μόλις ο χρήστης πάρει τον έλεγχο, λαμβάνονται αυτές πληροφορίες cookie περιόδου λειτουργίας οι οποίες μεταβιβάζονται σε κάθε αίτημα από διακομιστή σε πελάτη και αντίστροφα. Έτσι, κάθε φορά που ο διακομιστής λαμβάνει αίτημα με έγκυρη περίοδο λειτουργίας με πληροφοριες, εκτελείτε αυτό το αίτημα χωρίς να ενοχλεί το προέλευση του αιτήματος. Ως εκ τούτου, όταν ο εισβολέας CSRF στέλνει αίτημα μέσω του προγράμματος περιήγησης ενσωματώνοντάς το στον έγκυρο ιστότοπο, εκτελείται στον διακομιστή με επιτυχία και χωρίς να μπορεί κανείς να εντοπίσει ότι το αίτημα έχει προέλθει από άλλο τομέα και είναι άκυρο.

4.5.2 HTML tags

Οι εισβολείς CSRF ενσωματώνουν το αίτημα που θέλουν να εκτελέσουν σε ετικέτες HTML λόγω των οποίων η επίθεση γίνεται αόρατη και κατά τη φόρτωση μιας συγκεκριμένης σελίδας (με τη σελίδα, φορτώνει όλα στοιχεία που υπάρχουν στη σελίδα), το αίτημα εκτελείται. Επίσης κάποια στιγμή ενσωματώνεται στις ετικέτες όπου θα φτάσει και εκτελείται μόνο εάν ο χρήστης κάνει κλικ στη διεπαφή χρήστη αυτής της ετικέτας όπως «href tag». Σε αυτήν την περίπτωση ο εισβολέας αναγκάζει τον χρήστη να κάνει κλικ τέτοιες ετικέτες εμφανίζοντας κείμενο που προσελκύει τον χρήστη π.χ. «50% έκπτωση σε κοσμήματα» κλπ. Υπάρχουν τόσες πολλές ετικέτες σε HTML που μπορούν να στείλουν αίτημα στον διακομιστή, αλλά κάθε ετικέτα γίνεται για συγκεκριμένο τύπο αιτήματος όπως για αρχείο εικόνας, αρχείο JavaScript κ.λπ.. Η HTML δεν ελέγχει αν η ιδιότητα πηγής ετικέτας περιέχει την έγκυρη διεύθυνση URL, και οι επιτιθέμενοι CSRF το εκμεταλλεύονται αυτό.

Παραδείγματα επίθεσης με τη χρήση html tag

1. **HTML tag:** body

   <body { background: url(‘attack_request’)}> <body onload=”attack_request”>

2. **HTML tag:** img

3. **HTML tag:** input

   <input type = “image” src = “ attack_request” alt = “Submit” />

4. **HTML tag:** link

   <link rel = “stylesheet” type = “text/css” href = “attack_request” />

5. **HTML tag:** script

   <script type = “text/javascript” src =“attack_request” > </script>

6. **HTML tag:** table

<table background = “attack_request” >

7. **HTML tag:** td

   <td background = “attack_request”>

8. **HTML tag:** th

   <th background = “attack_request”>

9. **HTML tag:** iframe

   <iframe src=”attack_request”>

Μέθοδος υποβολής φόρμας GET και POST Οι πληροφορίες στα πεδία φόρμας αποστέλλονται στον διακομιστή χρησιμοποιώντας δύο μεθόδους GET και POST, όπου η μέθοδος GET δημιουργεί ένα αίτημα το οποίο περιέχει όλες τις πληροφορίες από μόνη της στο αίτημα και είναι επίσης ορατό στον χρήστη, έτσι ώστε ο εισβολέας να μπορεί χρησιμοποιήστε αυτές τις εύκολα διαθέσιμες πληροφορίες για να δημιουργήσετε έγκυρο αίτημα. Προτάθηκε η χρήση της μεθόδου POST αντί της μεθόδου GET για να σταματήσει αυτή η ευπάθεια. Αλλά η μέθοδος POST δεν βοηθά επίσης στην προστασία των εφαρμογών Ιστού από την επίθεση CSRF. Μόλις ο εισβολέας λάβει όλα τα πεδία φόρμας, μπορεί να ενσωματώσει αυτά τα πεδία στην ιστοσελίδα του, την οποία θα αναγκάσει το θύμα να ανοίξει και μπορεί να βάλει τη συνάρτηση JavaScript που επιτρέπει την υποβολή της φόρμας στο συμβάν onload. Το παρακάτω παράδειγμα περιγράψτε αυτό το σενάριο.

4.5.3 Μέθοδος υποβολής φόρμας GET και POST

Οι πληροφορίες στα πεδία φόρμας αποστέλλονται στον διακομιστή χρησιμοποιώντας δύο μεθόδους GET και POST, όπου η μέθοδος GET δημιουργεί ένα αίτημα το οποίο περιέχει όλες τις πληροφορίες από μόνη της στο αίτημα και είναι επίσης ορατό στον χρήστη, έτσι ώστε ο εισβολέας να μπορεί να τις χρησιμοποιήσει εύκολα ώστε να δημιουργήσει ένα έγκυρο αίτημα. Συστήνεται η χρήση της μεθόδου POST αντί της μεθόδου GET για να σταματήσει αυτή η ευπάθεια. Αλλά η μέθοδος POST δεν βοηθά επίσης στην προστασία των εφαρμογών Ιστού από την επίθεση CSRF. Μόλις ο εισβολέας λάβει όλα τα πεδία φόρμας, μπορεί να ενσωματώσει αυτά τα πεδία στην ιστοσελίδα του, την οποία θα αναγκάσει το θύμα να ανοίξει και έπειτα να μπορεί να βάλει τη συνάρτηση JavaScript που επιτρέπει την υποβολή της φόρμας στο συμβάν onload. Το παρακάτω παράδειγμα περιγράφει αυτό το σενάριο.

Where {element} = HTML element <{element}
onload=javascript:document.myform.submitO > <form name="myform" method="POST"
action=" {vulnerable site}"> <input name="variablel" value="attackl" > <input
name="variable2" value="attack2" > <input name="variable3" value="attack3" > </form>

Εδώ μπορούμε να δούμε ότι η φόρμα υποβάλλεται απευθείας στην φόρτωση της σελίδας (onload), χωρίς να το γνωρίζει ο χρήστης.

4.5.4 Επιλογή προγράμματος περιήγησης

Υπάρχουν πολλοί διαφορετικοί τρόποι με τους οποίους ο εισβολέας αποκτά γνώση της λειτουργικότητας που χρησιμοποιείται από την εφαρμογή Ιστού, κάτι που τον βοηθά να δημιουργήσει έγκυρο αίτημα. Ο εισβολέας μπορεί ο ίδιος να συνδεθεί στον ιστότοπο και να ελέγξει ολόκληρη τη λειτουργικότητα. Οι πληροφορίες σχετικά με τη λειτουργία των φορμών

στις ιστοσελίδες μπορούν να είναι εύκολα διαθέσιμες από τη δυνατότητα που παρέχεται από το ίδιο το πρόγραμμα περιήγησης χρησιμοποιώντας την επιλογή «Προβολή πηγής», η οποία εμφανίζει όλες τις πληροφορίες των πεδίων που υπάρχουν στο φόρμες, η επικύρωση για κάθε πεδίο μπορεί να προσπελαστεί χρησιμοποιώντας αρχεία JavaScript και πολλές περισσότερες πληροφορίες μπορεί να συλλέξει ο εισβολέας. Εάν η εφαρμογή Ιστού χρησιμοποιεί επιπλέον μεταβλητή συνεδρίας σε κάθε αίτημα για την προστασία της εφαρμογής από το CSRF και εάν αυτές οι πληροφορίες συνεδρίας αποθηκεύονται σε κρυφό πεδίο, χρησιμοποιώντας την επιλογή προέλευσης προβολής ο εισβολέας μπορεί να χρησιμοποιήσει τη λογική για τη δημιουργία αυτού του πεδίου περιόδου σύνδεσης.

4.5.5 Σφάλμα επικύρωσης εισόδου

Το CSRF μπορεί να χωριστεί σε δύο μορφές, το Αποθηκευμένο CSRF το οποίο είναι όταν ο εισβολέας εκτελεί το CSRF στον τομέα των στοχευμένων ιστοτόπων, και το ανακλώμενο CSRF όταν η επίθεση πυροδοτείται από διαφορετικό τομέα. Στην περίπτωση του Αποθηκευμένου CSRF μπορούμε να δώσουμε παράδειγμα ιστότοπου κοινωνικής δικτύωσης όπου ο χρήστης μπορεί να προσθέσει μια ανάρτηση που περιέχει κακόβουλο αίτημα που μπορεί να εκτελέσει κάποια κακόβουλη ενέργεια σε αυτόν τον ιστότοπο. Σε αυτήν την περίπτωση ο εισβολέας χρησιμοποιεί τα τρωτά σημεία που υπάρχουν στη λειτουργία επικύρωσης εισόδου. Κατά την επεξεργασία των δεδομένων εισόδου που υποβάλλονται από τον χρήστη, η μορφή τους θα πρέπει να είναι καλά καθορισμένη και καλά ελεγμένη. Εάν αυτή η λειτουργία επικύρωσης είναι αδύναμη, μπορεί να επιτρέψει στους εισβολείς κακόβουλου περιεχομένου να εισέλθουν στο σύστημα, κάτι που θα τους βοηθήσει να πραγματοποιήσουν το CSRF.[38]

4.5.6 Διαφορές μεταξύ XSS και CSRF επιθέσων

Η δέσμη ενεργειών μεταξύ τοποθεσιών (ή XSS) επιτρέπει σε έναν εισβολέα να εκτελέσει αυθαίρετη JavaScript μέσα στο πρόγραμμα περιήγησης ενός χρήστη-θύματος. Η πλαστογράφηση αιτημάτων μεταξύ τοποθεσιών (ή CSRF) επιτρέπει σε έναν εισβολέα να παρακινήσει έναν χρήστη-θύμα να εκτελέσει ενέργειες που δεν σκοπεύει να κάνει.

Ποια είναι όμως η διαφορά μεταξύ XSS και CSRF; Η XSS επιτρέπει σε έναν εισβολέα να εκτελέσει αυθαίρετη JavaScript μέσα στο πρόγραμμα περιήγησης ενός χρήστη-θύματος. Η πλαστογράφηση αιτημάτων μεταξύ τοποθεσιών (ή CSRF) επιτρέπει σε έναν εισβολέα να παρακινήσει έναν χρήστη-θύμα να εκτελέσει ενέργειες που δεν σκοπεύει να κάνει. Οι συνέπειες των τρωτών σημείων XSS είναι γενικά πιο σοβαρές από ό,τι για τις ευπάθειες CSRF: Το CSRF συχνά εφαρμόζεται μόνο σε ένα υποσύνολο ενεργειών που μπορεί να εκτελέσει ένας χρήστης. Πολλές εφαρμογές εφαρμόζουν άμυνες CSRF, αλλά παραβλέπουν μία ή δύο ενέργειες που αφήνονται εκτεθειμένες. Αντίθετα, μια επιτυχημένη εκμετάλλευση XSS μπορεί κανονικά να παρακινήσει έναν χρήστη να εκτελέσει οποιαδήποτε ενέργεια είναι σε θέση να εκτελέσει ο χρήστης, ανεξάρτητα από τη λειτουργικότητα στην οποία προκύπτει η ευπάθεια. Το CSRF μπορεί να περιγραφεί ως «μονόδρομη» ευπάθεια, καθώς ενώ ένας εισβολέας μπορεί να παρακινήσει το θύμα να εκδώσει ένα αίτημα HTTP, δεν μπορεί να ανακτήσει την απάντηση από αυτό το αίτημα. Αντίθετα, το XSS είναι "αμφίδρομη", καθώς το σενάριο που εισάγεται από τον εισβολέα μπορεί να εκδίδει αυθαίρετα αιτήματα, να διαβάζει τις απαντήσεις και να διεγείρει δεδομένα σε έναν εξωτερικό τομέα της επιλογής του εισβολέα.

Μπορεί όμως μια προσπάθεια προστασίας από επιθέσεις CSRF να αποτρέψουν και επιθέσεις XSS; Υποθέτοντας ότι ο διακομιστής επικυρώνει σωστά το διακριτικό CSRF και απορρίπτει αιτήματα χωρίς έγκυρο διακριτικό, τότε το διακριτικό αποτρέπει και την εκμετάλλευση της ευπάθειας XSS. Αποτρέποντας έναν εισβολέα να πλαστογραφήσει ένα αίτημα μεταξύ τοποθεσιών, η εφαρμογή αποτρέπει την εκμετάλλευση της ευπάθειας XSS. Ορισμένες σημαντικές προειδοποιήσεις προκύπτουν εδώ:

- Εάν μια ανακλώμενη ευπάθεια XSS υπάρχει οπουδήποτε αλλού στον ιστότοπο μέσα σε μια συνάρτηση που δεν προστατεύεται από ένα διακριτικό CSRF, τότε αυτό το XSS μπορεί να αξιοποιηθεί με τον κανονικό τρόπο.
- Εάν υπάρχει μια εκμεταλλεύσιμη ευπάθεια XSS οπουδήποτε σε έναν ιστότοπο, τότε η ευπάθεια μπορεί να αξιοποιηθεί για να κάνει έναν χρήστη-θύμα να εκτελέσει ενέργειες ακόμα κι αν αυτές οι ενέργειες προστατεύονται από διακριτικά CSRF. Σε αυτήν την περίπτωση, το σενάριο του εισβολέα μπορεί να ζητήσει από τη σχετική σελίδα να αποκτήσει ένα έγκυρο διακριτικό CSRF και στη συνέχεια να χρησιμοποιήσει το διακριτικό για να εκτελέσει την προστατευμένη ενέργεια.
- Τα διακριτικά CSRF δεν προστατεύουν από αποθηκευμένα τρωτά σημεία XSS. Εάν μια σελίδα που προστατεύεται από ένα διακριτικό CSRF είναι επίσης το σημείο εξόδου για μια αποθηκευμένη ευπάθεια XSS, τότε αυτή η ευπάθεια XSS μπορεί να αξιοποιηθεί με τον συνήθη τρόπο και το ωφέλιμο φορτίο XSS θα εκτελεστεί όταν ένας χρήστης επισκεφτεί τη σελίδα.

4.6 INSECURE DIRECT OBJECT REFERENCES

Το IDOR έχει θεωρηθεί ως μια σοβαρή ευπάθεια διαδικτυακής εφαρμογής. Σε αντίθεση με άλλες επιθέσεις όπως το XSS και το SQL-Injection, ο εντοπισμός ευπάθειας IDOR είναι λίγο δύσκολος ακόμα και με τη χρήση αυτοματοποιημένων εργαλεία. Αυτό συμβαίνει επειδή για να επιτεθούμε με επιτυχία χρησιμοποιώντας αυτό το ελάττωμα, πρέπει να ξεχωρίσουμε την ελαττωματική web interface καθώς και το μοτίβο για τον εντοπισμό ενός μη ασφαλούς αντικειμένου. Προκειμένου να εντοπιστεί μια διεπαφή που παρέχει πρόσβαση σε ευαίσθητα περιεχόμενα, πρέπει να προηγηθεί αναθεώρηση κώδικα και περιήγηση στον ιστότοπο. Παρόλο που το Insecure DOR δεν είναι μια νέα ευπάθεια, ο αντίκτυπος αυτής της ευπάθειας είναι κρίσιμος όταν συμβεί. Το Insecure DOR είναι ένα πρόβλημα που σχετίζεται με άδεια και δεν μπορεί να επιλυθεί αυτόματα ή από προεπιλογή, καθώς οι περιπτώσεις χρήσης αδειών διαφέρουν από web εφαρμογή σε web εφαρμογή. Το σχήμα απεικονίζει την ευπάθεια IDOR.

Όταν ο χρήστης δημιουργεί μια αναφορά - σύνδεσμο ο οποίος καταλήγει στο όνομα κάποιου αρχείου, σε μια βάση δεδομένων ή έναν κατάλογο, τότε εμφανίζεται και η ευπάθεια της άμεσης αναφοράς. Οι κακόβουλοι χρήστες βλέποντας τον σύνδεσμο μπορούν να τον εκμεταλλευτούν και να αποκτήσουν πρόσβαση και σε άλλα αρχεία χωρίς όμως να έχουν την εξουσιοδότηση του ιδιοκτήτη- χρήστη θύμα. Οι αναφορές αυτές μπορούν να προκύψουν είτε ως σύνδεσμοι - διευθύνσεις url είτε ώς φόρμες. Παραδείγματα τέτοιων αναφορών είναι τα αρχεία, οι κατάλογοι, αρχεία δεδομένων ή / και πρωτεύοντα κλειδιά. [39]

3.6.1 Τύποι επιθέσεων IDOR

Δύο τύποι προκύπτουν από την επίθεση ανασφαλούς απευθείας αναφοράς σε αντικείμενα, οι Open Redirects και Directory Traversal.

3.6.1.1 Ανοιχτές ανακατευθύνσεις

Η ανοιχτή ανακατεύθυνση είναι ένας τύπος ζητήματος ασφαλείας εφαρμογών ιστού που επιτρέπει στους εισβολείς να χρησιμοποιήσουν τη φήμη της επιχείρησής για να κάνουν τις επιθέσεις πιο αποτελεσματικές. Εάν επιτρέπονται ανοιχτές ανακατευθύνσεις, ένας εισβολέας μπορεί να στείλει ένα ηλεκτρονικό μήνυμα ηλεκτρονικού ψαρέματος που περιέχει έναν σύνδεσμο με το όνομα τομέα του χρήστη και το θύμα θα ανακατευθυνθεί από τον διακομιστή ιστού στον ιστότοπο του εισβολέα.

Μια ανακατεύθυνση συμβαίνει όταν ο ιστότοπος ή η εφαρμογή Ιστού αλλάζει τη διεύθυνση URL στην οποία έχει πρόσβαση ο πελάτης (συνήθως οι εξωτερικές – εσωτερικές ανακατευθύνσεις ονομάζονται συνήθως προς τα εμπρός). Υπάρχουν διάφοροι τρόποι για να το γίνει αυτό από το back-end. Συνήθως, οι ανακατευθύνσεις γίνονται στέλνοντας συγκεκριμένες κεφαλίδες HTTP στον πελάτη, αλλά μπορεί επίσης να δημιουργηθούν ανακατευθύνσεις, για παράδειγμα, χρησιμοποιώντας κώδικα JavaScript. Υπάρχει μια ευπάθεια ανοιχτής ανακατεύθυνσης όταν ο προορισμός της ανακατεύθυνσης παρέχεται από τον πελάτη και δεν φιλτράρεται ή επικυρώνεται. Ακολουθούν ορισμένα παραδείγματα ασφαλών ανακατευθύνσεων και μη ασφαλών ανακατευθύνσεων:

Εάν ο νόμιμος ιστότοπος ανακατευθύνει τον πελάτη σε μια σταθερή διεύθυνση URL, είναι μια ασφαλής ανακατεύθυνση. Εάν ο νόμιμος ιστότοπος κατασκευάζει με ασφάλεια τη διεύθυνση URL ανακατεύθυνσης με βάση τις παραμέτρους που παρέχονται από τον χρήστη, είναι μια ασφαλής ανακατεύθυνση.

Εάν ο νόμιμος ιστότοπος κατασκευάζει τη διεύθυνση URL ανακατεύθυνσης με βάση τις παραμέτρους που παρέχονται από τον χρήστη, αλλά δεν επικυρώνει/φιλτράρει επαρκώς τα δεδομένα εισόδου, πρόκειται για μη ασφαλή ανακατεύθυνση (ο εισβολέας μπορεί να χειραγωγήσει την είσοδο). Εάν ο νόμιμος ιστότοπος επιτρέπει στον χρήστη να καθορίσει τη διεύθυνση URL ανακατεύθυνσης προορισμού, πρόκειται για μη ασφαλή ανακατεύθυνση (ανοιχτή ανακατεύθυνση). Γενικά, οι προγραμματιστές που χρησιμοποιούν δυναμικές ανακατευθύνσεις (με βάση δεδομένα από τον πελάτη) πρέπει να αντιμετωπίζουν αυτά τα δεδομένα ως μη αξιόπιστα δεδομένα. Εάν όχι, ο εισβολέας θα ανακατευθύνει το πρόγραμμα περιήγησης σε έναν κακόβουλο ιστότοπο και θα χρησιμοποιήσει το όνομα τομέα σας για να ξεγελάσει το θύμα. Για παράδειγμα, εάν ο τομέας είναι ptuxiaki_alexiou.com, ο εισβολέας μπορεί να δημιουργήσει την ακόλουθη διεύθυνση URL:

https:// ptuxiaki_alexiou.com/redirect.php?url=http://attacker.com

Στη συνέχεια, ο εισβολέας μπορεί να στείλει αυτήν τη διεύθυνση URL ως μέρος μιας προσπάθειας ηλεκτρονικού ψαρέματος (phishing) να ανακατευθύνει το θύμα σε έναν κακόβουλο ιστότοπο attacker.com . Ο εισβολέας θα ελπίζει ότι το ptuxiaki_alexiou.com στην αρχή θα έχει μια αξιόπιστη εμφάνιση και θα τον κάνει να υποκύψει στην απάτη του phishing.

Ένα παράδειγμα ανοιχτής ανακατεύθυνσης

Ο ακόλουθος απλός κώδικας PHP δημιουργεί μια ανοιχτή ανακατεύθυνση:

$redirect = $_GET['url']; header("Location: " . $redirect);

Αυτή είναι μια ευπάθεια ανοιχτής ανακατεύθυνσης επειδή ο εισβολέας μπορεί να παρέχει μια κακόβουλη διεύθυνση URL ιστότοπου στην τιμή παραμέτρου url του αιτήματος GET και αυτή η διεύθυνση URL στόχου θα αποσταλεί στη συνέχεια ως κεφαλίδα τοποθεσίας, ανακατευθύνοντας τον πελάτη σε μια κακόβουλη ιστοσελίδα.

3.6.1.2 Συνέπειες των τρωτών σημείων ανοιχτής ανακατεύθυνσης

- Phishing: Ο πιο προφανής τρόπος για να χρησιμοποιηθεί μια ανοιχτή ανακατεύθυνση είναι να απομακρυνθεί το θύμα από τον αρχικό ιστότοπο σε έναν ιστότοπο που μοιάζει με τον ίδιο, να κλαπούν τα διαπιστευτήρια χρήστη και, στη συνέχεια, να επιστρέψει στον ευάλωτο ιστότοπο σαν να μην συνέβη τίποτα.
- Cross-site Scripting (XSS): Εάν η ανακατεύθυνση επιτρέπει τη χρήση πρωτοκόλλων data: ή javascript: και ο πελάτης υποστηρίζει τέτοια πρωτόκολλα σε ανακατευθύνσεις, δίνει τη δυνατότητα στον εισβολέα να εκτελέσει μια επίθεση XSS.
- Παραχάραξη αιτημάτων από την πλευρά του διακομιστή (SSRF): Ενδέχεται να χρησιμοποιηθούν ανοιχτές ανακατευθύνσεις για την αποφυγή φίλτρων SSRF.
- Παράκαμψη πολιτικής περιεχομένου-ασφάλειας: Εάν χρησιμοποιείται CSP για προστασία από το XSS και ένας από τους τομείς της λίστας επιτρεπόμενων έχει ανοιχτή ανακατεύθυνση, αυτή η ευπάθεια μπορεί να χρησιμοποιηθεί για την παράκαμψη του CSP.
- CRLF Injection: Εάν η παράμετρος ανακατεύθυνσης επιτρέπει αλλαγές γραμμής, ο εισβολέας μπορεί να προσπαθήσει να πραγματοποιήσει διαχωρισμό κεφαλίδας

3.6.2 Directory Traversal

Η διέλευση καταλόγου (γνωστή και ως διέλευση διαδρομής αρχείου) είναι μια ευπάθεια ασφαλείας ιστού που επιτρέπει σε έναν εισβολέα να διαβάζει αυθαίρετα αρχεία στον διακομιστή που εκτελεί μια εφαρμογή. Αυτό μπορεί να περιλαμβάνει κώδικα και δεδομένα εφαρμογής, διαπιστευτήρια για συστήματα back-end και ευαίσθητα αρχεία λειτουργικού συστήματος. Σε ορισμένες περιπτώσεις, ένας εισβολέας μπορεί να είναι σε θέση να γράψει σε αυθαίρετα αρχεία στο διακομιστή, επιτρέποντάς του να τροποποιήσει τα δεδομένα ή τη συμπεριφορά της εφαρμογής και, τελικά, να αναλάβει τον πλήρη έλεγχο του διακομιστή.

Έστω μια εφαρμογή αγορών που εμφανίζει εικόνες αντικειμένων προς πώληση. Οι εικόνες φορτώνονται μέσω κάποιου HTML όπως το ακόλουθο:

<img src="/loadImage?filename=137.png">

Η loadImage διεύθυνση URL λαμβάνει μια file name παράμετρο και επιστρέφει τα περιεχόμενα του καθορισμένου αρχείου. Τα ίδια τα αρχεία εικόνας αποθηκεύονται στο δίσκο στη θέση /var/www/images/. Για να επιστρέψει μια εικόνα, η εφαρμογή προσθέτει το ζητούμενο όνομα αρχείου σε αυτόν τον βασικό κατάλογο και χρησιμοποιεί ένα API συστήματος αρχείων για να διαβάσει τα περιεχόμενα του αρχείου. Στην παραπάνω περίπτωση, η εφαρμογή διαβάζει από την ακόλουθη διαδρομή αρχείου: /var/www/uploads/images/218.png

Η εφαρμογή δεν εφαρμόζει καμία άμυνα έναντι επιθέσεων διέλευσης καταλόγου, επομένως ένας εισβολέας μπορεί να ζητήσει την ακόλουθη διεύθυνση URL για να ανακτήσει ένα αυθαίρετο αρχείο από το σύστημα αρχείων του διακομιστή:

https:/ethniki-bank.com/loadImage?filename=../../../etc/passwd

Αυτό προκαλεί την ανάγνωση της εφαρμογής από την ακόλουθη διαδρομή αρχείου:

/var/www/uploads/images/../../../etc/passwd

Η ακολουθία ../είναι έγκυρη μέσα σε μια διαδρομή αρχείου και σημαίνει να ανεβεί ένα επίπεδο στη δομή του καταλόγου. Οι τρεις διαδοχικές ../ακολουθίες ανεβαίνουν από /var/www/uploads/images/τη ρίζα του συστήματος αρχείων, και έτσι το αρχείο που διαβάζεται πραγματικά είναι: /etc/passwd

Σε λειτουργικά συστήματα που βασίζονται σε Unix, αυτό είναι ένα τυπικό αρχείο που περιέχει λεπτομέρειες των χρηστών που είναι εγγεγραμμένοι στο διακομιστή. Στα Windows, ../και οι δύο ..\είναι έγκυρες ακολουθίες διέλευσης καταλόγου και μια ισοδύναμη επίθεση για την ανάκτηση ενός τυπικού αρχείου λειτουργικού συστήματος θα ήταν:

https://insecure-site.com/loadImage?filename=..\..\..\windows\win.ini

3.6.3 Παράδειγμα επίθεσης IDOR

Έστω οτι έχουμε το ακόλουθο URL: http://www.insecure-site.com/getfile.cfm?filename=alexiou_file.txt. Παρατηρούμε ότι υπάρχει μια άμεση αναφορά σε ένα αρχείο alexiou_file.txt, δίνοντας την ευκαιρία στον κακόβουλο χρήστη να μπει στον πειρασμό να αναζητήσει και άλλα αρχεία με το path αυτό. Προκειμένου λοιπόν να πετύχει ο hacker την εύρεση κάποιου αρχείου πρέπει να μαντέψει ακριβώς το όνομα αυτού ώστε να αποκτήσει πρόσβαση. Επιπρόσθετα ένα url που καταλήγει με /account.cfm?customerid=3245 αποτελεί πρόκληση για έναν κακόβουλο χρήστη να δει τι κρύβεται αν αντικαταστήσει τα id των χρηστών.

Τελικά η ευπάθεια αυτή μπορεί να προκαλέσει μεγάλα προβλήματα και διάρρευση προσωπικών πληροφοριών καθώς ο επιτιθέμενος έχει τη δυνατότητα να ανακτήσει δεδομένα στα οποία δεν θα έπρεπε να έχει πρόσβαση

4.7 SECURITY MISCONFIGURATION

Συνήθως συμβαίνει εσφαλμένη διαμόρφωση όταν ένας διαχειριστής συστήματος ή βάσης δεδομένων ή ένας προγραμματιστής δεν δημιουργεί μια ασφαλής εφαρμογή. Αυτό οδηγεί σε λεωφόρους για εκμετάλλευση από χάκερ. Για έναν έμπειρο εγκληματία στον κυβερνοχώρο, οι εσφαλμένες διαμορφώσεις θεωρούνται συχνά ως εύκολος στόχος, καθώς μπορεί να είναι απλό να εντοπιστούν εσφαλμένοι διαμορφωμένοι διακομιστές ιστού, παρουσίες cloud και εφαρμογές, οι οποίες στη συνέχεια γίνονται εκμεταλλεύσιμοι. Αυτό μπορεί να προκαλέσει σημαντικές βλάβες και οδηγούν σε καταστροφικά δεδομένα με ζητήματα διαρροής για τις επιχειρήσεις. Δυστυχώς, όταν ένα σύστημα πέσει θύμα ευπάθειας ή έλλειψης προστασίας ασφαλείας, τα ευαίσθητα δεδομένα του χρήστη κινδυνεύουν να κλαπούν ή να αλλοιωθούν. Συχνά, το μεγαλύτερο πρόβλημα που αντιμετωπίζουν οι οργανισμοί είναι ότι αυτά τα ελαττώματα δεν εντοπίζονται ή δεν αντιμετωπίζονται αρκετά έγκαιρα σύμφωνα με τις βέλτιστες πρακτικές ασφάλειας. Αυτό αποτελεί μια ευρέως διαδεδομένη πρόκληση ασφαλείας.

Η αυξανόμενη σημασία του ψηφιακό μετασχηματισμό και η ικανότητα για τις επιχειρήσεις να επεκτείνουν τα δίκτυά τους στη μέγιστη δυνατότητά τους ώστε να αντιμετωπίσουν θέματα όπως η πανδημία Covid-19 και η στροφή στην εξ αποστάσεως εργασία ανέδειξε την σημασία

της ασφάλειας. Μπορεί να είναι απίστευτα επιζήμια μακροπρόθεσμα εάν ομάδες ασφαλείας της επιχειρήσεις δεν εφαρμόσουν τα βασικά του web ασφάλεια εφαρμογών. Ολο και περισσότερο η ανησυχητική τάση των εσφαλμένων διαμορφώσεων ασφαλείας φαίνεται να υποδηλώνει ότι η ζωτική ασφάλεια παρακάμπτεται όταν αυτή έρχεται στη διαμόρφωση των εφαρμογών ιστού, δίκτυα και λύσεις cloud, είτε λόγω ταχύτητας ή απλού ανθρώπινο λάθος.

Μερικές φορές τα πιο ασφαλές περιβάλλοντα μέσα σε έναν οργανισμό είναι αυτά που σχεδιάστηκαν και κατασκευάστηκαν όπως συστήματα για διαχειριστές και προγραμματιστές. Δυστυχώς, παρά την προσπάθεια που τίθενται στην ασφαλή ανάπτυξη και υλοποίηση διαδικτυακών εφαρμογών, παραμένουν πολλά κενά ευπάθειας, ακόμα και μετά την ολοκλήρωση της δουλειάς. Τελικά, η σημασία της ασφάλειας ακόμα και από το σχεδιασμό δεν πρέπει να υποτιμάται, ιδιαίτερα καθώς όλο και περισσότερες εφαρμογές Ιστού στοχοποιούνται από εγκληματίες στον κυβερνοχώρο. Οι εσφαλμένες εφαρμογές ενδέχεται να κοστίσουν περισσότερο από ό,τι θα περιμένατε μακροπρόθεσμα. Η καλύτερη λύση είναι η εκπαίδευση των εργαζομένων σχετικά με τις καταστροφικές συνέπειες των μη ασφαλών εφαρμογών και η ενσωμάτωση πρακτικών κωδικοποίησης ασφαλείας. [40]

Τέτοια κενά ασφαλείας θα μπορούσαν να οδηγήσουν το κινδύνους, όπως δαπανηρές πρόστιμα και ζημιά στη φήμη οι οποίοι τελικά θα επηρεάσουν τις επιχειρήσεις που δεν προσαρμόστηκαν αρκετά γρήγορα. Ωστόσο, υπάρχει λύση. Ομάδες ασφάλειας και ανάπτυξης θα πρέπει να κοιτάξουν να δημιουργήσουν εφαρμογές με ενσωματωμένη ασφάλεια. Οι ομάδες μπορούν να αναζητήσουν τα κενά που μπορούν να εκμεταλλευτούν από εγκληματίες του κυβερνοχώρου, δημιουργώντας ένα ασφαλές ψηφιακό περιβάλλον στη διαδικασία. Οι επιχειρήσεις θα πρέπει να προσέξουν τα εξής σημεία:

- Ρυθμίσεις ασφαλείας (π.χ. ονόματα χρήστη και κωδικούς πρόσβασης).
- Μη κρυπτογραφημένα αρχεία.
- Παλιές και ξεπερασμένες διαδικτυακές εφαρμογές.
- Μη ασφαλείς συσκευές.
- Εφαρμογή Web και cloud εσφαλμένη διαμόρφωση.
- Ανεπαρκής προστασία, τείχους προστασίας.

Οι προγραμματιστές θα πρέπει να προσέχουν όλα τα επίπεδα της εφαρμογής καθώς σε οποιοδήποτε επίπεδο μπορεί να εμφανιστεί η λανθασμένη παραμετροποίηση ασφαλείας, συμπεριλαμβανομένης της πλατφόρμας, του web server, του server εφαρμογών, του framework, και του προσαρμοσμένου κωδικού. Αναμφισβήτητα όμως δεν αρκεί μόνο ο σωστή διαχείριση από τους προγραμματιστές, αλλα απαιτείται η κατάλληλη εκπαίδευση και των διαχειριστών.

4.8 INSECURE CRYPTOGRAPHIC STORAGE

Η μη ασφαλής κρυπτογραφική αποθήκευση εμφανίζεται όταν η εφαρμογή αποθηκεύει ευαίσθητες πληροφορίες όπως σύνδεση και κωδικό πρόσβασης ή πίστωση τον αριθμό της κάρτας και τις ανάγκες προσωπικών στοιχείων αλλά χωρίς ισχυρή κρυπτογράφηση ή χρήση αδύναμου κλειδιού τα αποθηκευμένα δεδομένα κινδυνεύουν να κλαπούν. Το αποτέλεσμα λοιπόν αυτής της ευπάθειας είναι συνήθως αρκετά επικίνδυνος λόγω του γεγονότος ότι οι πληροφορίες που συνήθως είναι κρυπτογραφημένες αποτελούν σημαντικά πράγματα όπως

στοιχεία προσωπικής ταυτοποίησης, εμπορικά μυστικά, αρχεία υγειονομικής περίθαλψης, προσωπικές πληροφορίες και αριθμούς πιστωτικών καρτών.

Οι σύγχρονοι κρυπτογραφικοί αλγόριθμοι είναι εξαιρετικά ανθεκτικοί και μπορεί να χρειαστεί πολύς χρόνος για να σπάσουν. Το πρόβλημα δεν είναι με τους αλγόριθμους που χρησιμοποιούνται, το θέμα είναι με τον τρόπο με τον οποίο εφαρμόζονται για να διατηρούνται ασφαλή τα δεδομένα του χρήστη. Οι περισσότεροι εισβολείς παρακολουθήσουν τον τρόπο με τον οποίο χρησιμοποιείται η κρυπτογραφία, όχι η ίδια την κρυπτογραφία.

Το Insecure Cryptographic Storage τελικά δεν είναι ένα μόνο θέμα ευπάθειας, αλλά μια συλλογή από σημαντικά σημεία. Όλα τα σημεία της συλλογής έχουν να κάνουν με τη διασφάλιση ότι τα πιο σημαντικά δεδομένα είναι κρυπτογραφημένα όταν χρειάζεται. Αυτό περιλαμβάνει:

- Σωστά κρυπτογραφημένα δεδομένα
- Σωστή αποθήκευση και διαχείριση κλειδιών
- Μη χρήση γνωστών αλγορίθμων

Οι προγραμματιστές συχνά υποθέτουν ότι η αποθήκευση δεδομένων δεν θα εξεταστεί από έναν αυθαίρετο χρήστη. Αλλά πολλοί χρήστες μιας εφαρμογής ή προγράμματος έχουν πρόσβαση στο μητρώο, τα προσωρινά αρχεία και τις βάσεις δεδομένων. Είναι δυνατό για αυτούς τους χρήστες να έχουν πρόσβαση σε ευαίσθητα δεδομένα στη μη κρυπτογραφημένη μορφή τους χρησιμοποιώντας προσωρινά, κρυφά αρχεία και αρχεία μητρώου. Είναι επίσης δυνατό για έναν εισβολέα να αποκτήσει πρόσβαση χρησιμοποιώντας μια άλλη από τις κορυφαίες 10 ευπάθειες του OWASP, όπως η Direct Object Access.

Η αυτοματοποιημένη και Χειροκίνητη προσέγγιση αποτελούν λύσεις προκειμένου να πραγματοποιηθούν έλεγχοι για το αν η εφαρμογή κρυπτογραφεί τα ευαίσθητα δεδομένα και τις πληροφορίες σωστά. Αρχικά με την αυτοματοποιημένη προσέγγιση μπορεί να γίνει έλεγχος μόνο αν χρησιμοποιούνται κρυπτογραφημένα APIs αλλα οχι αν η χρήση αυτή γίνεται με τον σωστό τρόπο. Έπειτα με την χειροκίνητη προσέγγιση η καλύτερη επιλογή είναι ο επανεξέταση του κώδικα προκειμένου να ανιχνευτεί η σωστή ή μη λειτουργία του συστήματος.[41]

4.9 FAILURE TO RESTRICT URL ACCESS

Εάν η εφαρμογή αποτύχει να περιορίσει κατάλληλα την πρόσβαση στη διεύθυνση URL, η ασφάλεια μπορεί να τεθεί σε κίνδυνο μέσω μιας τεχνικής που ονομάζεται αναγκαστική περιήγηση. Η αναγκαστική περιήγηση μπορεί να είναι ένα πολύ σοβαρό πρόβλημα εάν ένας εισβολέας προσπαθήσει να συγκεντρώσει ευαίσθητα δεδομένα μέσω ενός προγράμματος περιήγησης ιστού ζητώντας συγκεκριμένες σελίδες ή αρχεία δεδομένων.

Χρησιμοποιώντας αυτήν την τεχνική τελικά, ένας εισβολέας μπορεί να παρακάμψει την ασφάλεια του ιστότοπου προσεγγίζοντας απευθείας αντί να ακολουθεί συνδέσμους. Αυτό επιτρέπει στον εισβολέα να έχει απευθείας πρόσβαση στα αρχεία προέλευσης δεδομένων αντί να χρησιμοποιεί την εφαρμογή Ιστού. Ο εισβολέας μπορεί στη συνέχεια να μαντέψει τα ονόματα των αρχείων αντιγράφων ασφαλείας που περιέχουν ευαίσθητες πληροφορίες, να εντοπίσει και να διαβάσει τον πηγαίο κώδικα ή άλλες πληροφορίες που έχουν απομείνει στον διακομιστή και να παρακάμψει τη "σειρά" των ιστοσελίδων.

Με απλά λόγια, η αποτυχία περιορισμού της πρόσβασης στη διεύθυνση URL εμφανίζεται όταν ένα σφάλμα στις ρυθμίσεις ελέγχου πρόσβασης έχει ως αποτέλεσμα οι χρήστες να μπορούν να έχουν πρόσβαση σε σελίδες που προορίζονται να περιοριστούν ή να κρυφτούν. Οι επιθέσεις αναγκαστικής περιήγησης μπορούν λοιπόν να πραγματοποιηθούν όταν ένας εισβολέας είναι σε θέση να μαντέψει σωστά τη διεύθυνση URL ή να χρησιμοποιήσει ωμή βία για πρόσβαση σε μια μη προστατευμένη σελίδα. Αυτή η διαδικασία είναι πολύ πιο εύκολη για τον εισβολέα εάν υπάρχει κάποιο ελάττωμα στην πολιτική ελέγχου πρόσβασης της σελίδας. Αυτά τα ελαττώματα περιλαμβάνουν συνήθως κρυφές σελίδες με πιθανές διευθύνσεις URL, εφαρμογές που επιτρέπουν την πρόσβαση σε σελίδες που προορίζονται για απόκρυψη/περιορισμό, παρωχημένο κώδικα πολιτικής ελέγχου πρόσβασης και έλλειψη πολιτικής ελέγχου πρόσβασης από την πλευρά του διακομιστή**.**

Αυτό δημιουργεί ανησυχία για την ασφάλεια, καθώς αυτές οι σελίδες συχνά προστατεύονται λιγότερο από τις σελίδες που προορίζονται για δημόσια πρόσβαση και οι μη εξουσιοδοτημένοι χρήστες μπορούν να προσεγγίσουν τις σελίδες ανώνυμα. Σε πολλές περιπτώσεις, η μόνη προστασία που χρησιμοποιείται για κρυφές ή περιορισμένες σελίδες είναι η μη σύνδεση με τις σελίδες ή η μη δημόσια εμφάνιση συνδέσμων προς αυτές.[42]

Για παράδειγμα έστω ότι ένας ιστότοπος ακολουθεί μια συνηθισμένη και εύκολα ανακαλυψιμη δομή:

[domain name]/admin , [domain name]/admin/admin.html

[domain name]/products

[domain name]/login.html

[domain name]/index.html

Ένας κακόβουλος χρήστης λοιπόν με περιορισμένη πρόσβαση μπορεί τελικά να μαντέψει κάποια URL και να αποκτήσει πρόσβαση σ΄ αυτά. Αδιαμφισβήτητα δεν θα έπρεπε παραδείγματος χάριν να έχει πρόσβαση στο λογαριασμό του χρήστη αλλά αν δεν έχουν εφαρμοστεί περιορισμοί πρόσβασης ο χακερ μπορεί να εισβάλει σε όποιο url επιθυμεί. Απαραίτητη προϋπόθεση είναι να γνωρίζει στο ακριβές path του URL κάτι το οποίο όμως δεν είναι αρκετά δύσκολα αν ο ιστότοπος χρησιμοποιεί προβλέψιμα ονόματα.

Παράδειγμα: https://[domain url]/admin/admin.html.

4.9.1 Επιπτώσεις

Δεδομένου ότι ο διαχειριστής έχει αποτύχει στον περιορισμό πρόσβασης σε απαγορευμένα URL, οι κακόβουλοι χρήστες μπορούν να αποκτήσουν πρόσβαση σε λειτουργίες για τις οποίες δεν θα έπρεπε να έχουν πρόσβαση. Επιπρόσθετα οι χακερ μπορούν να αποκτήσουν πρόσβαση σε ευαίσθητα προσωπικά δεδομένα όπως για παράδειγμα στοιχεία άλλων χρηστών, πρόσβαση σε λογαριασμούς και τελικά να εκτελέσουν κακόβουλες ενέργειες.

4.10 INSUFFICIENT TRANSPORT LAYER PROTECTION

Η ανεπαρκής προστασία επιπέδου μεταφοράς είναι μια αδυναμία ασφαλείας που προκαλείται από τις εφαρμογές που δεν λαμβάνουν μέτρα για την προστασία της κυκλοφορίας του δικτύου. Κατά τη διάρκεια του ελέγχου ταυτότητας, οι εφαρμογές μπορεί να χρησιμοποιούν SSL/TLS,

αλλά συχνά αποτυγχάνουν να το χρησιμοποιήσουν, αφήνοντας έτσι εκτεθειμένα τα δεδομένα και τα αναγνωριστικά περιόδου λειτουργίας. Τα εκτεθειμένα δεδομένα και τα αναγνωριστικά περιόδου σύνδεσης μπορούν να υποκλαπούν, πράγμα που σημαίνει ότι η εφαρμογή είναι ευάλωτη σε εκμετάλλευση.

Επειδή πολλές εκδόσεις των πρωτοκόλλων SSL/TLS χρησιμοποιούνται ευρέως σε πολλές αναπτυγμένες εφαρμογές, όπως η περιήγηση στον Ιστό, το ηλεκτρονικό ταχυδρομείο, η αποστολή email μέσω Διαδικτύου, η ανταλλαγή άμεσων μηνυμάτων, η φωνή μέσω IP (VoIP) και πολλές άλλες εφαρμογές που επικοινωνούν μέσω Διαδικτύου se ανεπαρκές επίπεδο μεταφοράς. Η ευπάθεια αυτή είναι ένατη στο OWASP Top 10 κινδύνους.

Στα cms προσφέρεται προστασία SLL στους ιστοτόπους απο τις υπηρεσίες hosting με μερικά μόλις βήματα και πολλές φορές δωρεάν. Ωστόσο μερικές φορές η ευπάθεια αυτή υπάρχει η οποία όμως δεν προσελκύει τους κακόβουλους χρήστες σε μεγάλο βαθμό καθώς θα πρέπει να γνωρίζουν τη κίνηση στο διαδίκτυο. Ένας διαχειριστής προκειμένου να ανακαλύψει αν έχει ανεπαρκή προστασία θα χρειαστεί να διερευνήσει τα ακόλουθα ζητήματα:

- Όλες οι συνδέσεις, όχι μόνο αυτές με τους διακομιστές που διαθέτει, να είναι σωστά κρυπτογραφημένες
- Να ενημερωμένα τα πιστοποιητικά SSL
- Τα πιστοποιητικά SSL να είναι αυτο-υπογεγραμμένα
- Αν το SSL Χρησιμοποιεί αρκετά υψηλές τακτικές κρυπτογράφησης

4.10.1 Επιπτώσεις

Η ευπάθεια ανεπαρκούς προστασίας στο επίπεδο μεταφοράς μπορεί να προκαλέσει διάρρευση προσωπικών δεδομένων. Σε περίπτωση λοιπόν που ένας ιστότοπος βρεθεί ευάλωτος σε μια τέτοια επίθεση από πιθανώς λανθασμένη ρύθμιση SSL ο χάκερ μπορεί να αποκτήσει πρόσβαση σε μεγάλο μέρος του site και με αυτό τον τρόπο να κλεψει τραπεζικά και προσωπικά στοιχεία καθιστώντας το site αναξιόπιστο.

5. Αντιμετώπιση ευπαθειών

Η αξιολόγηση ευπάθειας είναι μια διαδικασία εντοπισμού στα κενά ασφαλείας ή στα τρωτά σημεία στο σύστημα του υπολογιστή, δικτύου ή διαδικτυακών εφαρμογών οργανισμού με την απαραίτητη γνώση, κατανόηση της υποδομής και κατανόηση των πιθανών απειλών του περιβάλλοντος. Μετά τη συνολική διαδικασία αξιολόγησης πραγματοποιείτε μια λεπτομερής αναφορά η οποία μπορεί να χρησιμοποιηθεί περαιτέρω για τη διαδικασία δοκιμής και διείσδυσης .

Πλεονέκτημα: Η αξιολόγηση ευπάθειας είναι χρήσιμη για την οργάνωση της ασφάλειας. Επίσης, είναι χρήσιμο καθώς χιλιάδες έλεγχοι ασφάλειας γίνονται σε λιγότερο χρόνο με τη βοήθεια εργαλείων αυτοματισμού.

Μειονέκτημα: Η αξιολόγηση ευπάθειας δεν μπορεί να εντοπιστεί τα λογικά διανύσματα επίθεσης. Τα αυτοματοποιημένα εργαλεία δημιουργούν μόνο την ανάλυση σύμφωνα με τις πολιτικές που ορίζονται από τον ελεγκτή, σε περίπτωση που υπάρχει μια ευπάθεια αλλά δεν αναλύεται στο σύστημα αυτοματισμού τότε αυτη παραλείπεται και ενδέχεται να μην ληφθεί υπόψη στην ανάλυση.

Στη συνέχεια του κεφαλαίου αυτού θα δούμε τρόπους αντιμετώπισης των ευπαθειών που παρουσιάστηκαν σε προηγούμενο κεφάλαιο.

5.1 Αντιμετώπιση INJECTION FLAWS επίθεσης

1. Επικύρωση εισαγωγής

Η επικύρωση εισόδου εκτελείται για να διασφαλιστεί ότι μόνο σωστά διαμορφωμένα δεδομένα εισέρχονται στη ροή εργασιών σε ένα σύστημα πληροφοριών, αποτρέποντας τη διατήρηση λανθασμένων δεδομένων στη βάση δεδομένων και την πρόκληση δυσλειτουργίας διαφόρων στοιχείων. Για παράδειγμα στην προσπάθεια εισαγωγής ενός ονόματος χρήστη θα πρέπει να γίνεται έλεγχος και να μην επιτρέπονται οι ειδικοί χαρακτήρες (Ma<ri>a). Η εισαγωγή πρέπει να λέξχεται απο μια συνάρτηση επικύρωσης προκειμένου να επιτρέπονται μόνο τα στοιχεία που αναμένονται. Στο επόμενο κεφάλαιο θα παραθέσουμε και ένα παράδειγμα.

Η επικύρωση των εισροών θα πρέπει να γίνεται όσο το δυνατόν νωρίτερα στη ροή δεδομένων, κατά προτίμηση μόλις ληφθούν τα δεδομένα από το εξωτερικό μέρος. Τα δεδομένα από όλες τις δυνητικά αναξιόπιστες πηγές θα πρέπει να υπόκεινται σε επικύρωση εισόδου, συμπεριλαμβανομένων όχι μόνο πελατών ιστού που έχουν πρόσβαση στο Διαδίκτυο, αλλά και τροφοδοτήσεων υποστήριξης μέσω extranets, από προμηθευτές, συνεργάτες, προμηθευτές ή ρυθμιστικές αρχές, καθένας από τους οποίους μπορεί να διακυβευτεί από μόνος του και να αρχίσει να στέλνει λανθασμένη μορφή δεδομένων. Η επικύρωση εισόδου συμπερασματικά δεν θα πρέπει να χρησιμοποιείται ως η κύρια μέθοδος αποτροπής επιθέσεων XSS, SQL Injection και άλλων επιθέσεων που καλύπτονται στις αντίστοιχες μεθόδους εξαπάτησης.[43]

2. Εφαρμογή ελάχιστου προνομίου

Μια άλλη ισχυρή προστασία από επιθέσεις INJECTION FLAWS είναι να διασφαλιστεί ότι η εφαρμογή Ιστού εκτελείται μόνο με τα προνόμια που χρειάζεται οπωσδήποτε για να εκτελέσει τη λειτουργία της. Επομένως, δεν θα πρέπει να εκτελείται ο διακομιστής ιστού ως root ή να έχει πρόσβαση σε μια βάση δεδομένων ως DBADMIN, διαφορετικά ένας εισβολέας μπορεί να κάνει κατάχρηση αυτών των προνομίων διαχείρισης που παρέχονται στην εφαρμογή Ιστού. Ορισμένα από τα περιβάλλοντα J2EE επιτρέπουν τη χρήση του Java sandbox, το οποίο μπορεί να εμποδίσει την εκτέλεση εντολών συστήματος.

3. Χειρισμός εξαιρέσεων και επιστρεφόμενων κωδικών κατάστασης

Εάν πρέπει να χρησιμοποιηθεί εξωτερική εντολή, οποιεσδήποτε πληροφορίες χρήστη εισάγονται στην εντολή θα πρέπει να ελέγχονται αυστηρά. Θα πρέπει να τεθούν σε εφαρμογή μηχανισμοί για την αντιμετώπιση τυχόν σφαλμάτων, χρονικών ορίων ή μπλοκαρίσματος κατά τη διάρκεια της κλήσης. Όλοι οι κωδικοί εξόδου, επιστροφής και κωδικοί σφάλματος από την κλήση θα πρέπει να ελέγχονται για να διασφαλιστεί ότι η αναμενόμενη επεξεργασία πραγματοποιήθηκε πράγματι. Τουλάχιστον, αυτό θα επιτρέψει να διαπιστωθεί ότι κάτι πήγε στραβά. Διαφορετικά, η επίθεση μπορεί να συμβεί και να μην εντοπιστεί ποτέ.

4. Διερεύνηση τεχνικών μετριασμού για συγκεκριμένες τεχνολογίες που χρησιμοποιεί η εφαρμογή

Διαφορετικοί τύποι επίθεσης απαιτούν διαφορετικές στρατηγικές μετριασμού (π.χ. XSS έναντι έγχυσης προτύπου από την πλευρά του διακομιστή). Πραγματοποίηση ελέγχων για το ποιες

τεχνολογιες χρησιμοποιεί η εφαρμογή και τις διαθέσιμες πληροφορίες σχετικά με τα βήματα που πρέπει να ληφθούν για την αποτροπή κλάσεων επίθεσης που κάνουν κατάχρηση αυτών των τεχνολογιών.

5. Αποφυγή πρόσβασης σε εξωτερικούς διερμηνείς

Ένας άλλος τρόπος προστασίας από την επίθεση είναι να αποφεύγεται η πρόσβαση σε εξωτερικούς διερμηνείς όπου είναι δυνατόν. Για πολλές εντολές φλοιού και ορισμένες κλήσεις συστήματος, υπάρχουν βιβλιοθήκες για συγκεκριμένες γλώσσες που εκτελούν τις ίδιες λειτουργίες. Η χρήση τέτοιων βιβλιοθηκών δεν περιλαμβάνει τον διερμηνέα φλοιού του λειτουργικού συστήματος, και επομένως αποφεύγει μεγάλο αριθμό προβλημάτων με εντολές φλοιού.

5.1.2 Παράδειγμα επίθεσης SQL injection

Στο παράδειγμα που ακολουθεί αναπτύσσεται ένας κώδικας με τον χρήστη να μπορεί να εισάγει το όνομα του και τον κωδικό. Με τη χρήση ελλιπή ελέγχου εισόδου μπορεί να οδηγήσει σε επίθεση SQL injection.

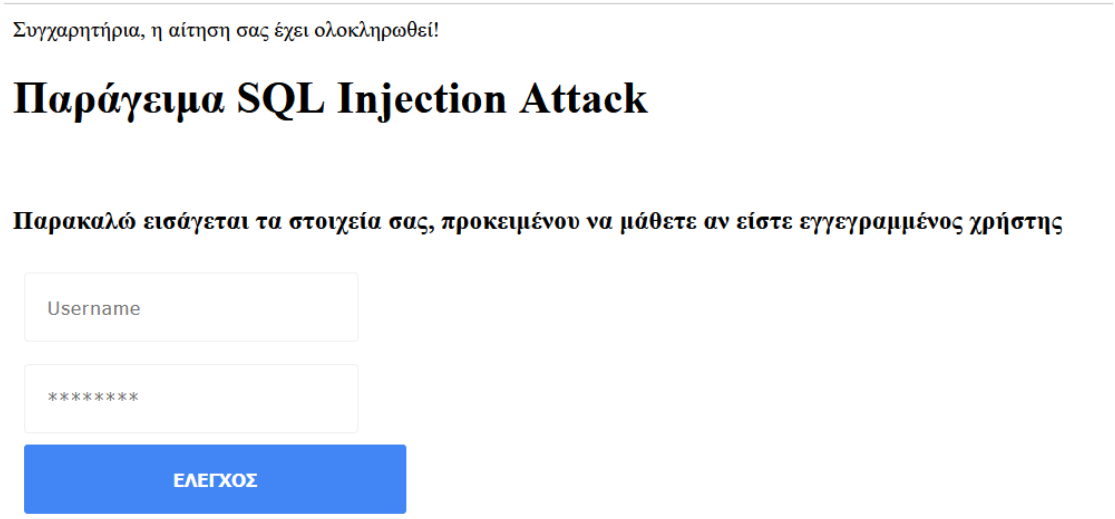
Εφόσον λοιπόν ο χρήστης εισάγει τα στοιχεία του και βρίσκεται στη βάση μας "cms_security" η εφαρμογή αυτή θα μας επιστρέψει ένα μήνυμα επιτυχίας.

Η εφαρμογή μας έχει αναπτυχθεί σε php και MySQL για την επικοινωνία με την βάση. Στον ακόλουθο κώδικα θα φανεί πως ο μη ασφαλής κώδικας μπορεί να οδηγήσει σε SQL Injection.

```php
<?php
$hostname = "localhost";
$username = "alexiou";
$password = "maria123";
$dbname = "cms_security";
$conn = mysqli_connect($hostname, $username, $password, $dbname);
if(!$conn) {
    die("Unable to connect");
}
if($_POST) {
    $uname = $_POST["username"];
    $pass = $_POST["password"];

    $sql = "SELECT * FROM users WHERE UserName = '$uname' AND Password = '$pass'";
    $result = mysqli_query($conn, $sql);
    if(mysqli_num_rows($result) == true) {
        echo "Συγχαρητήρια, η αίτηση σας έχει ολοκληρωθεί!";
    } else {
        echo "Λυπούμαστε αλλα δεν τα στοιχεία σας δεν βρέθηκαν στη βάση δεδομένων μας. ";
    }
}
?>
<!DOCTYPE html>
<html>
<head>
    <title>SQL Injection Attack</title>
</head>
<body>
    <h1> Παράγειμα SQL Injection Attack </h1>
    <br>
    <h3> Παρακαλώ εισάγεται τα στοιχεία σας, προκειμένου να μάθετε αν είστε εγγεγραμμένος χρήστης</h3>
    <form action method="POST" autocomplete="off">
        <input type="text" name="username" placeholder="Username" /><br />
        <input type="password" name="password" placeholder="********" /><br />
        <input type="submit" name="login" value="ΕΛΕΓΧΟΣ" />
    </form>
</body>
</html>
```

Βάζοντας λοιπόν ως είσοδο Name: test’ or 1=1# και Password: test’ or 1=1# παρατηρούμε οτι λανθασμένα ο χρήστης παίρνει μήνυμα επιτυχίας οτι τα στοιχεία του βρίσκονται στη βάση.

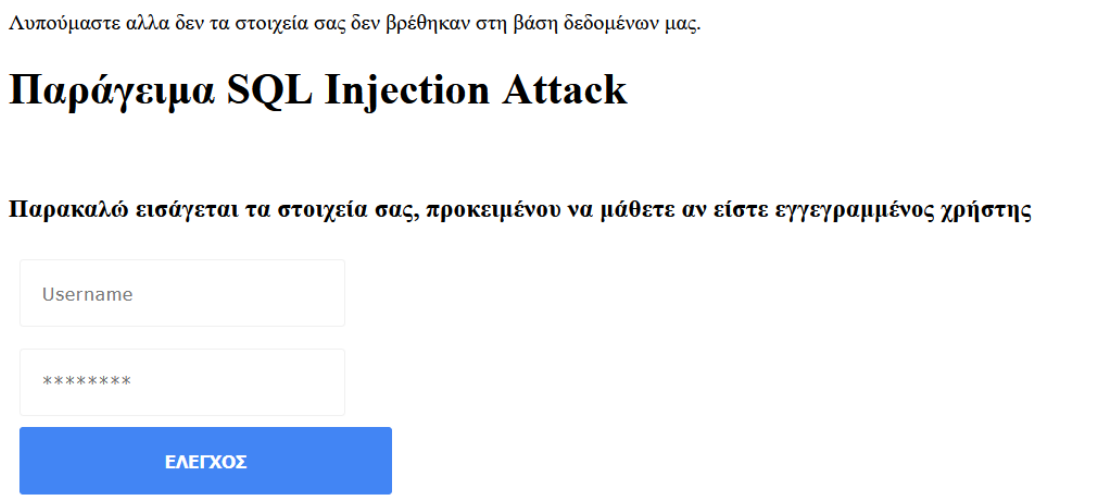


Στον κώδικα που ακολουθεί προσθέσαμε την mysqli_real_escape_string η οποία και μας προστατεύει από επιθέσεις SQL Injection απορρίπτοντας κενά και ειδικούς χαρακτήρες σε μια συμβολοσειρά σε ένα ερώτημα SQL, λαμβάνοντας υπόψη το σύνολο των χαρακτήρων της σύνδεσης.

```php
<?php
$hostname = "localhost";
$username = "alexiou";
$password = "maria123";
$dbname = "cms_security";
$conn = mysqli_connect($hostname, $username, $password, $dbname);
if(!$conn) {
    die("Unable to connect");
}
if($_POST) {
    $uname = $_POST["username"];
    $pass = $_POST["password"];
     //Making sure that SQL Injection doesn't work
     $uname = mysqli_real_escape_string($conn, $uname);// test' or 1=1#
     $pass = mysqli_real_escape_string($conn, $pass);

    $sql = "SELECT * FROM users WHERE UserName = '$uname' AND Password = '$pass'";
    $result = mysqli_query($conn, $sql);
    if(mysqli_num_rows($result) == true) {
        echo "Συγχαρητήρια, η αίτηση σας έχει ολοκληρωθεί!";
    } else {
        echo "Λυπούμαστε αλλα δεν τα στοιχεία σας δεν βρέθηκαν στη βάση δεδομένων μας. ";
    }
}
?>
<!DOCTYPE html>
<html>
<head>
    <title>SQL Injection Attack</title>
</head>
<body>
    <h1> Παράγειμα SQL Injection Attack </h1>
    <br>
    <h3> Παρακαλώ εισάγεται τα στοιχεία σας, προκειμένου να μάθετε αν είστε εγγεγραμμένος χρήστης</h3>
    <form action method="POST" autocomplete="off">
        <input type="text" name="username" placeholder="Username" /><br />
        <input type="password" name="password" placeholder="********" /><br />
        <input type="submit" name="login" value="ΕΛΕΓΧΟΣ" />
    </form>
</body>
</html>
```

Με την προσθήκη αυτή λοιπόν βλέπουμε ότι ο χρήστης δεν μπορεί να εισάγει μια αληθής συνθήκη προκειμένου να παραβιάσει το σύστημα και να μπορέσει να προχωρήσει.



5.2 Προστασία επιθέσων CROSS - SITE SCRIPTING

Με την κλοπή ταυτότητας αποκτόντας πρόσβαση σε προσωπικές πληροφορίες, παραβίαση της περιήγησης του χρήστη, πρόσβαση σε εφαρμογές επί πληρωμή, αλλοίωση εφαρμογής ή του

browser αποτελούν μερικές απο τις επιπτώσεις που μπορεί να επιφέρει η επίθεση XSS. Καμία από τις περιπτώσεις υποκλοπής δεν θεωρείται αμελητέα καθώς μπορεί να δημιουργήσει πολλά και σημαντικά προβλήματα στον κάτοχο - θύμα. Οι ίδιοι οι χρήστες πρέπει να είναι ιδιαίτερα προσεκτικοι καθώς ανα πάσα στιγμή μπορεί να πέσουν σε κάποια 'παγίδα' και ο κακόβουλος χρήστης να αποκτήσει πρόσβαση στα δεδομένα του χρήστη. Στη συνέχεια του κεφαλαίου αυτού θα αναλύσουμε μερικούς απο τους τρόπους που μπορεί να αποφευχθεί μια επίθεση XSS. [43]

1. Φιλτράρισμα περιεχομένου

Αυτή η τεχνική άμυνας χρησιμοποιεί λειτουργίες φίλτρου για την αφαίρεση δυνητικά κακόβουλων δεδομένων ή οδηγιών από τα δεδομένα εισόδου του χρήστη. Οι λειτουργίες φίλτρου εφαρμόζονται μετά την εισαγωγή του χρήστη όπου τα στοιχεία διαβάζοται από μια εφαρμογή web, αλλά πριν από τη είσοδο χρησιμοποιείται μια άλλη λειτουργία. Η κατάργηση σεναρίων από μη αξιόπιστο περιεχόμενο είναι ένα δύσκολο πρόβλημα για εφαρμογές web που επιτρέπουν την HTML σήμανση στα στοιχεία των χρηστών, όπως ιστολόγιο, wiki και κοινωνικά εφαρμογές δικτύωσης. Αυτές οι εφαρμογές επεκτείνονται και πολλαπλασιάζονται ταχέως , επομένως υπάρχει αυξανόμενη ανάγκη για ισχυρές άμυνες XSS. Το WordPress είναι εξουσιοδοτεί ανώνυμους χρήστες να ελέγχουν την παρουσίασή τους με σχόλια ιστολογίου. Το κάνει επιτρέποντας την εισαγωγή του δομημένα στοιχεία HTML για μορφοποίηση κειμένου (π.χ. <b> για έντονη γραφή, <i> για πλάγια γραφή). Με βάση το φιλτράρισμα περιεχομένου οι άμυνες αυτού του τύπου εφαρμογής αντιμετωπίζουν μια δύσκολη πρόκληση: επιτρέποντας όλες τις καλοήθεις εισαγωγές χρηστών HTML, ενώ αποκλείοντας ταυτόχρονα όλα τα δυνητικά επιβλαβή σενάρια στην μη αξιόπιστη έξοδο.[45]

Απλώς απαγορεύοντας τους χαρακτήρες ελέγχου σύνταξης HTML δεν είναι μια πρακτική λύση φιλτραρίσματος για αυτές τις εφαρμογές γιατί κάθε χαρακτήρας ελέγχου που μπορεί να χρησιμοποιηθεί η εισαγωγή κώδικα επίθεσης έχει επίσης νόμιμη χρήση σε ορισμένους. Για παράδειγμα, ο χαρακτήρας < πρέπει να υπάρχει σε υπερσυνδέσμους για μορφοποίηση κειμένου, και ο χαρακτήρας " πρέπει να υπάρχει στο γενικό κείμενο περιεχόμενο. Και οι δύο είναι νόμιμες και επιτρεπόμενες εισροές χρηστών, αλλά μπορεί τελικά να γίνει κατάχρηση για την προσάρτηση επιθέσεων XSS. Τα προηγμένα φίλτρα περιεχομένου λοιπόν προσπαθούν να προβλέψουν πώς θα ερμηνευτεί το μη αξιόπιστο περιεχόμενο από τον ιστό του πελάτη. Ο αναλυτής του προγράμματος περιήγησης,λοιπόν δημιουργεί κρίσιμες αποφάσεις για την εκτέλεση του σεναρίου. Για να είναι εντελώς αποτελεσματική στην εξάλειψη του XSS, μια λειτουργία φίλτρου πρέπει απαραίτητα να μοντελοποιείται το πλήρες εύρος των συμπεριφορών ανάλυσης που σχετίζονται με την εκτέλεση σεναρίων για πολλά προγράμματα.

Παρόλα αυτά σε κάποιες περιπτώσεις χρειάζεται να αφαιρούνται οι χωρίς έγκριση χαρακτήρες αποτρέποντας εισόδους που τους περιέχουν.

2. Συνεργασία προγράμματος περιήγηση (browser)

Ισχυρή πρόληψη επιθέσεων XSS μπορούν να επιτευχθούν εάν δημιουργηθούν προγράμματα περιήγησης Ιστού τα οποία μπορούν να διακρίνουν τον εξουσιοδοτημένο από το μη εξουσιοδοτημένο χρήστη. Αυτό το όραμα υιοθετήθηκε για πρώτη φορά στο BEEP , όπου αυτή

η προσέγγιση εφαρμόστηκε με: (α) τη δημιουργία ενός πρωτοκόλλου συνεργασίας διακομιστή-προγράμματος περιήγησης για την επικοινωνία του συνόλου των εξουσιοδοτημένων σεναρίων και στη συνέχεια (β) την τροποποίηση του πρόγραμματος περιήγησης για να κατανοήσει αυτό το πρωτόκολλο και να επιβάλει α πολιτική άρνησης μη εξουσιοδοτημένης εκτέλεσης σεναρίου. Παρόλο που η αμυντική στρατηγική που οραματίστηκαν οι συγγραφείς του BEEP είναι μια συναρπαστική και αποτελεσματική μακροπρόθεσμη λύση, η προσέγγιση εφαρμογής τους αφήνει ένα μεγάλο κενό στη βραχυπρόθεσμη προστασία. Αυτό είναι επειδή οι διαδικτυακές εφαρμογές που υιοθετούν αυτήν την προσέγγιση απαιτούν χρήστες για να χρησιμοποιούν προσαρμοσμένα προγράμματα περιήγησης με δυνατότητα BEEP προστασία από επιθέσεις XSS. Για να κλιμακωθεί αυτή η προσέγγιση πρέπει πρώτα να υπάρξει συμφωνία για τυχόν προτεινόμενα πρότυπα για τη συνεργασία διακομιστή-προγράμματος περιήγησης, και στη συνέχεια αυτά τα νέα τα πρότυπα πρέπει να ενσωματωθούν στο κανονικό πρόγραμμα περιήγησης εφαρμόζοντας το για εκατομμύρια εγκατεστημένα προγράμματα περιήγησης. Αυτή είναι μια μακρά, περίπλοκη διαδικασία που μπορεί να διαρκέσει αρκετά χρόνια. Αυτός ο εγγενής πρακτικός περιορισμός καθιστά τη συνεργασία του προγράμματος περιήγησης ακατάλληλη για υιοθέτηση στο εγγύς μέλλον υπάρχουσες διαδικτυακές εφαρμογές. [46]

3. Μη χρήση εισαγόμενων στοιχείων

Μια επιπλέον αντιμετώπιση μιας XSS επιθεσης αποτελεί η προστασία των στοιχείων που εισάγει ο χρήστης στο σύστημα. Να μην χρησιμοποιούνται δήλαδή άμεσα. Με την τακτική αυτή όμως περιορίζονται οι τεχνικές που πλέον χρησιμοποιούν τα καταστήματα παραδείγματος χάριν. Αν το σύστημα δεν καταγράφει την αναζήτηση του χρήστη με τη χρήση των λέξεων κλειδιών περιορίζει σημαντικά η προσπάθεια των εφαρμογών για SEO κάνοντας την προβολή τους σε στοχευμένους χρήστες σημαντικά δύσκολη.

4. Cookies

Τα cookies αποτελούν μια μορφή ασφάλειας ενάντια στις επιθέσεις XSS. Καθώς ενα cookie συνδέεται σε μια συγκεκριμένη ip επιτυγχάνεται η απαραίτητη επικύρωση προκειμένου να αντιμετωπιστεί μια επίθεση XSS. Ωστόσο υπάρχουν περιπτώσεις οπου μπορεί να παραβιαστεί η ασφάλεια της επιλ΄υρωσης όταν ο χρήστης - θύμα και ο κακόβουλος χρήστης που πραγματοποιεί την επίθεση προέρχονται απο τον ίδιο proxy server. Για την προστασία του χρήστη ο internet explorer εφαρμόζει μια HTTPOnly σημαία. Η HttpOnly λοιπόν είναι μια πρόσθετη σημαία που περιλαμβάνεται σε μια κεφαλίδα απόκρισης HTTP Set-Cookie. Η χρήση της σημαίας HttpOnly κατά τη δημιουργία ενός cookie συμβάλλει στον μετριασμό του κινδύνου πρόσβασης του σεναρίου από την πλευρά του πελάτη στο προστατευμένο cookie (αν το υποστηρίζει το πρόγραμμα περιήγησης). Εάν η σημαία HttpOnly περιλαμβάνεται στην κεφαλίδα απόκρισης HTTP, δεν είναι δυνατή η πρόσβαση στο cookie μέσω του σεναρίου της πλευράς του πελάτη εάν όμως το πρόγραμμα περιήγησης υποστηρίζει αυτήν τη σημαία. Ως αποτέλεσμα, ακόμη και αν υπάρχει ελάττωμα μεταξύ δέσμης ενεργειών (XSS) και ένας χρήστης αποκτήσει κατά λάθος πρόσβαση σε έναν σύνδεσμο που εκμεταλλεύεται αυτό το ελάττωμα, το πρόγραμμα περιήγησης (κυρίως ο Internet Explorer) δεν θα αποκαλύψει το cookie σε τρίτο μέρος. Το παρακάτω παράδειγμα δείχνει τη σύνταξη που χρησιμοποιείται στην κεφαλίδα απόκρισης HTTP:

Set-Cookie: <name>=<value>[; <Max-Age>=<age>]

`[; expires=<date>][; domain=<domain_name>]

[; path=<some_path>][; secure][; HttpOnly]

Εάν ένα πρόγραμμα περιήγησης δεν υποστηρίζει το HttpOnly και ένας ιστότοπος επιχειρήσει να ορίσει ένα cookie HttpOnly, η σημαία HttpOnly θα αγνοηθεί από το πρόγραμμα περιήγησης, δημιουργώντας έτσι ένα παραδοσιακό cookie προσβάσιμο σε σενάριο. Ως αποτέλεσμα, το cookie (συνήθως το cookie συνεδρίας) γίνεται ευάλωτο σε κλοπή ή τροποποίηση από κακόβουλο σενάριο.

5.3 Αντιμετώπιση μιας Broken Authentication and Session Management επίθεσης

Με την επιτυχημένη επίθεση Broken Authentication and Session Management ο κακόβουλος χρήστης μπορεί να λειτουργήσει ακριβώς όπως θα έκανε και ο πιστοποιημένος χρήστης. Πως μπορεί τελικά όμως να αποφευχθεί μια τέτοια επίθεση; Στη συνέχεια του κεφαλαίου αυτου περιγράφουμε μερικούς από τους τρόπους αντιμετώπισης.

1. **Έλεγχος χρονικού ορίου περιόδου λειτουργίας**

Ανάλογα με τον τύπο της εφαρμογής, θα πρέπει να εφαρμόζονται τα χρονικά όρια περιόδου λειτουργίας, έτσι ώστε ο χρήστης να μην παραμένει συνδεδεμένος ακόμα και μετά την ολοκλήρωση της χρήσης της εφαρμογής.

2. **Εναλλαγή και ακύρωση αναγνωριστικών συνεδρίας**

Όπως αναφέρθηκε προηγουμένως, τα αναγνωριστικά περιόδου σύνδεσης θα πρέπει να αλλάξουν αφού ένας χρήστης συνδεθεί με επιτυχία για να αποφευχθούν επιθέσεις σταθεροποίησης περιόδου σύνδεσης. Ομοίως, τα αναγνωριστικά περιόδου σύνδεσης θα πρέπει να οριστούν σωστά ως άκυρα κατά την αποσύνδεση.

3. **Αποφυγεί αποθήκευσης των κωδικών πρόσβασης σε καθαρό κείμενο**

Οι εφαρμογές που αποθηκεύουν τα διαπιστευτήρια χρήστη θα πρέπει να το κάνουν με ασφαλή τρόπο. Τέτοιες βάσεις δεδομένων θα πρέπει να είναι κρυπτογραφημένες και ο κωδικός πρόσβασης θα πρέπει να αποθηκεύεται ως κατακερματισμένος.

3. **Προστασία με παραβιασμένο κωδικό πρόσβασης**

Όπου είναι δυνατόν, οι εφαρμογές θα πρέπει να εφαρμόζουν λύσεις που ταιριάζουν με τους κωδικούς πρόσβασης χρήστη με μια γνωστή λίστα διαπιστευτηρίων που έχουν παραβιαστεί, ώστε ο χρήστης να μπορεί να ενημερωθεί και να μην χρησιμοποιεί τα ίδια διαπιστευτήρια.

4. **Περιορισμός προσπάθειας σύνδεσης**

Ο αριθμός των αποτυχημένων προσπαθειών σύνδεσης θα πρέπει να περιοριστεί, ώστε ο χρήστης να έχει μόνο έναν σταθερό αριθμό προσπαθειών σε μια καθορισμένη χρονική περίοδο. Αυτό θα μειώσει τις πιθανότητες επίθεσης και θα αποτρέψει επίσης πρόσθετο φόρτο στον διακομιστή ιστού, καθώς η επισκεψιμότητα ιστού μπορεί να αυξηθεί έως και 180 φορές κατά τη διάρκεια μιας επίθεσης διαπιστευτηρίων. Ένα κλείδωμα λογαριασμού θα πρέπει να ξεκινήσει όταν ένας χρήστης φτάσει τα μέγιστα επιτρεπόμενα αιτήματα.

5. Απαρίθμηση ονόματος χρήστη και κωδικού πρόσβασης

Σε εσφαλμένες συνδέσεις, η εφαρμογή δεν θα πρέπει να προσδιορίζει ποιο μέρος του ελέγχου ταυτότητας ήταν λανθασμένο, αλλα η εφαρμογή θα πρέπει να δίνει μια γενική απάντηση σφάλματος.

6. Προστασία ID

Το σύνολο της εφαρμογής του χρήστη μιας εφαρμογής θα πρέπει να προστατεύεται μέσω SSL. Μερικά συστήμα όπως το Wordpress προσφέρουν την κρυπτογράφηση αυτή σε μόλις μερικά βήματα. Χωρίς να απαιτούνται γνώσεις προγραμματισμού απο την διαχειριστή. Εάν αυτό τελικά γίνει, τότε το ID δεν είναι δυνατόν να υποκλαπεί από το δίκτυο. Το ID προκειμένου να είναι αποτελεσματικό όμως πρέπει να αποτελείται από έναν ελάχιστο αριθμό χαρακτήρων, περίπλοκο με τη χρήση αριθμών και συμβολοσειρών τα οποία δύσκολα μπορούν να βρεθούν από έναν κακόβουλο χρήστη. Σε περίπτωση που δεν υπάρχει δυνατότητα εφαρμογής SSL θα χρειαστεί να χρησιμοποιηθούν άλλοι τρόποι.

5.4 Προστασία απο CROSS SITE REQUEST FORGERY επίθεση

Καθώς η επίθεση CSRF έγινε δημοφιλής διάφορα αμυντικά μέτρα προτάθηκαν εναντίον του, αλλά κανένα από αυτά δεν είναι σε θέση να αντιμετωπίσει την CSRF πλήρως. Τα μέτρα αυτά όμως ελαχιστοποιούν τον κίνδυνο σε κάποιο βαθμό. Σε αυτην την υπο ενότητα θα εξετάσουμε τέτοια αμυντικά μέτρα που θα μας βοηθήσει να χτίσουμε πιο στιβαρή τεχνική για να μετριαστεί το CSRF.

5.4.1 Έλεγχος κεφαλίδας

Ένα αίτημα HTTP περιέχει διαφορετικές παραμέτρους, μία από αυτές τις παραμέτρους περιέχουν τη διεύθυνση URL του ιστότοπου από τον οποίο ζητείται το όνομα παραμέτρου που είναι το "Referer". Η παράμετρος αυτη μπορεί να χρησιμοποιηθεί από το πρόγραμμα περιήγησης για τον έλεγχο των αιτημάτων τομέα στην πλευρά του πελάτη πριν από την προώθηση του αιτήματος στον διακομιστή. Έτσι, οι προγραμματιστές ιστού ελέγχουν την κεφαλίδα Referer για τη προστασία της εφαρμογής από το CSRF. Αυτό μπορεί να εφαρμοστεί σε περίπτωση όπως η αλλαγή κωδικού πρόσβασης, μεταφορά ποσού, αγορά αντικειμένων και αλλαγή προνομίων χρήστη κλπ. Αυτό θα επιτρέψει την εκτέλεση μόνο του ίδιου αιτήματος τομέα.

5.4.2 Εισαγωγή τυχαίων tokens

Ένα διακριτικό CSRF είναι μια μοναδική, μυστική, απρόβλεπτη τιμή που δημιουργείται από την εφαρμογή διακομιστή και μεταδίδεται στον πελάτη με τέτοιο τρόπο ώστε να περιλαμβάνεται σε ένα επόμενο αίτημα HTTP που υποβάλλεται από τον πελάτη. Όταν υποβληθεί το μεταγενέστερο αίτημα, η εφαρμογή από την πλευρά του διακομιστή επικυρώνει ότι το αίτημα περιλαμβάνει το αναμενόμενο διακριτικό και απορρίπτει το αίτημα εάν το διακριτικό λείπει ή δεν είναι έγκυρο.

Τα διακριτικά CSRF μπορούν να αποτρέψουν επιθέσεις CSRF καθιστώντας αδύνατο για έναν εισβολέα να δημιουργήσει ένα πλήρως έγκυρο αίτημα HTTP κατάλληλο για τροφοδοσία σε έναν χρήστη-θύμα. Δεδομένου ότι ο εισβολέας δεν μπορεί να προσδιορίσει ή να προβλέψει την τιμή του διακριτικού CSRF ενός χρήστη, δεν μπορεί να δημιουργήσει ένα αίτημα με όλες τις παραμέτρους που είναι απαραίτητες για να ικανοποιήσει η εφαρμογή το αίτημα. Ο πιο ισχυρός τρόπος άμυνας από επιθέσεις CSRF τελικά είναι να συμπεριλάβετε ένα token στα σχετικά αιτήματα. Το διακριτικό πρέπει να είναι:

- Απρόβλεπτο με υψηλή εντροπία, όπως και για τα session token γενικά.
- Συνδεδεμένο με τη συνεδρία του χρήστη.
- Αυστηρά επικυρωμένο σε κάθε περίπτωση πριν από την εκτέλεση της σχετικής ενέργειας.

5.4.3 Η επικύρωση του διακριτικού CSRF εξαρτάται από τη μέθοδο αιτήματος

Ορισμένες εφαρμογές επικυρώνουν σωστά το διακριτικό όταν το αίτημα χρησιμοποιεί τη μέθοδο POST, αλλά παραλείπουν την επικύρωση όταν χρησιμοποιείται η μέθοδος GET. Σε αυτήν την περίπτωση, ο εισβολέας μπορεί να μεταβεί στη μέθοδο GET για να παρακάμψει την επικύρωση και να παραδώσει μια επίθεση CSRF

5.5 Προστασία απο INSECURE DIRECT OBJECT REFERENCES επίθεση

Ο ιστότοπός σας μπορεί να είναι ευάλωτος σε μελλοντικές επιθέσεις εάν οι παράγοντες απειλής IDOR παραμείνουν παρόντες στις εφαρμογές Ιστού σας. Ακολουθούν τρόποι για να αποφευχθεί η ευπάθεια IDOR:

5.5.1 Εμμεσος χάρτης αναφοράς

Ο χάρτης αναφοράς αντικαθιστά τις πραγματικές αναφορές (όπως αναγνωριστικά χρήστη, ονόματα, κλειδιά κ.λπ.) με εναλλακτικά αναγνωριστικά που αντιστοιχίζονται στις αρχικές τιμές. Η αντιστοίχιση μεταξύ των εναλλακτικών αναγνωριστικών και των πραγματικών αναφορών διατηρείται με ασφάλεια στους διακομιστές.

Για παράδειγμα έστω ότι έχουμε μια πιστωτική κάρτα. Οι αναφορές στη διεύθυνση URL www.example.com/credit/profile/id#15, που θα χρησιμοποιήθούν πρέπει να είναι έμμεσες αναφορές www.example.com/c/ab. Με αυτόν τον τρόπο, οι άμεσες αναφορές που περιέχουν τα στοιχεία της πιστωτικής κάρτας των χρηστών δεν θα εκτίθενται.

5.5.2 Επικύρωση πρόσβασης χρήστη

Οι διακομιστές αποτυγχάνουν να αναγνωρίσουν παραποιημένες διευθύνσεις URL επειδή δεν υπάρχουν έλεγχοι πρόσβασης σε επίπεδο δεδομένων-αντικειμένων. Τα στοιχεία ελέγχου πρόσβασης επιπέδου δεδομένων θα πρέπει να επιβάλλονται μόνο όταν ο διακομιστής επαληθεύει εάν ο τρέχων χρήστης κατέχει ή έχει δικαιώματα πρόσβασης στα ζητούμενα δεδομένα. Τουλάχιστον, η εφαρμογή θα πρέπει να εκτελεί μια συντακτική επικύρωση για να επαληθεύσει ύποπτες εισόδους. Η εφαρμογή θα πρέπει να καθορίσει κριτήρια για τα εισερχόμενα δεδομένα και, εάν δεν ανταποκρίνεται στις προσδοκίες, να απορρίψει την τιμή. Ακολουθούν ορισμένα κριτήρια που μπορούν να εφαρμοστούν:

- Ελάχιστο ή μέγιστο μήκος
- Ελάχιστα ή μέγιστα όρια (για αριθμητικές τιμές)
- Αποδεκτοί χαρακτήρες
- Τύπος δεδομένων (π.χ. συμβολοσειρά, ημερομηνία, ακέραιος, ορθολογικός κ.λπ.)

5.6 Αντιμετώπιση επίθεσης - SECURITY MISCONFIGURATION

Είναι αρκετά εύκολο να προκληθούν λανθασμένες ρυθμίσεις στις παραμέτρους ασφάλειας ακόμα και αν το σύστημα έχει εξασφαλίσει διαμορφώσεις για τη πλήρη ασφάλεια του καθώς ακόμα και μια απλή παράβλεψη μπορεί να προκαλέσει κενά ασφάλειας. Η καταχώρηση καταλόγου είναι ένα κοινό πρόβλημα με τις εφαρμογές ιστού, ιδιαίτερα εκείνες που βασίζονται σε προϋπάρχοντα πλαίσια όπως το WordPress. Οι χρήστες περιηγούνται και έχουν πρόσβαση στη δομή του αρχείου ελεύθερα, ώστε να μπορούν εύκολα να ανακαλύψουν και να εκμεταλλευτούν ευπάθειες ασφαλείας. Τα πιο συνηθισμένα λάθη τελικά που οδηγούν στην εσφαλμένη διαμόρφωση ασφάλειας είναι τα εξής:

- Οι περιττές θύρες παραμένουν ανοιχτές,
- Περιττές υπηρεσίες που επιτρέπεται να εκτελούνται,
- Σελίδες που έχουν απομείνει ακόμα διαθέσιμες για πρόσβαση
- Διέλευση καταλόγου που επιτρέπει σε έναν εισβολέα να έχει πρόσβαση σε καταλόγους, αρχεία και εντολές που βρίσκονται εκτός του ριζικού καταλόγου.
- Αχρησιμοποίητοι λογαριασμοί με ορισμένα προνόμια δεν διαγράφονται.

Στη συνέχεια παραθέτουμε μερικούς τρόπους αντιμετώπισης της ευπάθειας αυτής

5.6.1 Περιορισμός πρόσβασης στη διεπαφή χρήστη

Το κύριο βήμα που κάθε προγραμματιστής πρέπει να κάνει είναι να κατανοήσει καλά το σύστημα που διαχειρίζεται. Έπειτα μπορεί να προχωρήσει σε μερικές συγκεκριμένες ρυθμίσεις.

Αρχικά απαραίτητη αποτελεί η δημιουργία ενός νέου συνόλου διαπιστευτηριων απενεργοποιώντας την χρήση προεπιλεγμένων λογαριασμών και κωδικών πρόσβασης. Η εγκατάσταση ενημερώσεων κώδικα και λογισμικού τακτικά και έγκαιρα αποτελεί σημαντικό στοιχείο για ένα ασφαλές περιβάλλον. Έπειτα οι συχνοί έλεγχοι και σαρώσεις οι οποίοι θα εντοπίζουν πιθανές ενημερώσεις κώδικα και λογισμικού που λείπουν δεν πρέπει να παραλείπονται. Στη συνέχεια ένας καλά δομημένος και διατηρημένος κύκλος ανάπτυξης διευκολύνει τη δοκιμή ασφάλειας της εφαρμογής στην φάση της ανάπτυξης.

Η κρυπτογράφηση και η δημιουργία γνήσιων ελέγχων πρόσβασης τόσο σε σε αρχεία όσο και σε καταλόγους μπορεί να βοηθήσει στην αντιστάθμιση των τρωτών σημείων των αρχείων και των καταλόγων που δεν είναι προστατευμένα. Ο έλεγχος της άδειας αποθήκευσης στο cloud δεν πρέπει να παραλείπεται.

Αναμφισβήτητα δεν θα πρέπει να απουσιάζει και η καλή εκπαίδευση των υπαλλήλων σχετικά με τη σημασία των διαμορφώσεων ασφαλείας και πώς μπορούν να επηρεάσουν την ασφάλεια του γενικού οργανισμού. Η ενσωμάτωση μια αυτοματοποιημένης διαδικασίας μπορεί τελικά να διασφαλίσει ένα ασφαλές περιβάλλον και να είναι προς όφελός της επιχείρησης.

5.7 Αντιμετώπιση επίθεσης - INSECURE CRYPTOGRAPHIC STORAGE

Οι προγραμματιστές θα πρέπει να αναγνωρίζουν όλα τα ευαίσθητα δεδομένα και να τα κρυπτογραφούν, ακόμη και όταν είναι αποθηκευμένα σε σκληρό δίσκο. Να διασφαλίζουν ότι τα ευαίσθητα δεδομένα δεν μπορούν να αντικατασταθούν εύκολα. Επιπλέον, θα πρέπει να περιορίσουν τον αριθμό των χρηστών που θα γνωρίζουν μυστικά, όπως ιδιόκτητους αλγόριθμους, κλειδιά κρυπτογράφησης και DRM.

Για τη διασφαλιση της αποθήκευσης ευαίσθητων δεδομένων, απαραίτητα αποτελούν τα ακόλουθα βήματα:

- Προσδιορισμός όλων των ευαίσθητων δεδομένων και κρυπτογράφηση αυτών ακόμα και όταν είναι αποθηκευμένα σε σκληρό δίσκο
- Διασφάλιση ότι τα ευαίσθητα δεδομένα δεν μπορούν να αντικατασταθούν
- Αντικατασταση των ευαίσθητων τοποθεσιών μνήμης αμέσως αφού τα δεδομένα δεν χρειάζονται πλέον στη μνήμη
- Προσδιορισμός των ατόμων που πρέπει και δεν πρέπει να γνωρίζουν μυστικά
- Διασφάλιση μυστικων, όπως ιδιόκτητους αλγόριθμους, κλειδιά κρυπτογράφησης και DRM ακόμα και από τον διαχειριστή
- Προσδιορισμός ευαίσθητων δεδομένων που διαβάζονται στη μνήμη, αντικαταστη με τυχαία δεδομένα και χρησης ισχυρής κρυπτογράφησης για να την προστασία τους.
- Δημιουργία μόνο συγκεκριμένων δημόσιων αλγορίθμων όπως είναι AES SHA1 και MD5. Καλύτερα όμως να επιλέγονται ασφαλής αλγόριθμος όπως η SHA-256 αντί MD5 και SH1

5.8 Αντιμετώπιση επίθεσης - FAILURE TO RESTRICT URL ACCESS

Οι εισβολείς μπορούν να χρησιμοποιήσουν αρκετά απλές μεθόδους για να αποκτήσουν πρόσβαση και να αλληλεπιδράσουν με κρυφές/μη συνδεδεμένες σελίδες σε έναν ιστότοπο. Η πιο συνηθισμένη μέθοδος λοιπόν είναι αυτός ο τύπος επίθεσης.

Τα πιο τρωτά σημεία που μια επιχείρηση πρέπει να ελέγχει είναι:

- Ύπαρξη ελέγχου πρόσβασης
- Ορισμός μιας λίστας αρχείων τα οποία είναι διαθέσιμα για απομακρυσμένη ανάγνωση στον διακομιστή. Πολλοί διακομιστές επιτρέπουν να αριστούν ποιες επεκτάσεις αρχείων μπορούν να εξυπηρετηθούν εξ αποστάσεως Για παράδειγμα, τα αρχεία .log, .dat και βάσης δεδομένων δεν είναι αρχεία στα οποία θα πρέπει να έχουν πρόσβαση όλοι οι χρήστες - παρά μόνο μέσω ασφαλών καναλιών.
- Διαγραφή των μη απαραίτητων αρχείων ακόμα και αν αυτά είναι ασφαλή προκειμένου να αποφευχθούν πιθανοί κίνδυνοι
- Χρήση κατάλληλων δικαιωμάτων ώστε να απαγορευτεαι η ανώνυμη περιήγηση
- Σε περίπτωση ύπαρξης μια επέκτασης php να συμπεριλαμβάνονται αρχεία βιβλιοθήκης.
- Απαγόρευση πρόσβασης σε όλους τους τύπους αρχείων που η εφαρμογή δεν πρέπει να υποστηρίζει όπως .php, .html, .pdf

- Εγκατάσταση και ενημέρωση προγραμμάτων κατά των ιών (antivirus).

5.8.1 Παράδειγμα Restrict Url Access

Έστω ότι ο διαχειριστής θέλει να περιορίσει την πρόσβαση των χρηστών σε μια html σελίδα όπως για παράδειγμα την users.html. Το πρώτο που θα χρειαστεί να κάνει είναι να συνδεθεί είτε στο panel είτε μέσω filezilla και να βρει το αρχείο .htaccess. Έπειτα εκεί θα χρειαστεί να προσθέσει τον ακόλουθο κώδικα: [47]

—

#block access to the file blockedfile.html; permit access to all other files

<files users.html>

Order Allow,Deny

Deny from all

</files>

Αν όμως θέλει να περιορίσει την πρόσβαση των χρηστών από τις εικόνες ο κώδικας θα διαμορφωθεί ως εξής:

#block access to image files - files with jpg/gif/png/jpeg extensions

<FilesMatch "\.(jpg|gif|png|jpeg)$">

Order Allow,Deny

Deny from all

</FilesMatc>

Ένας ακόμη τρόπος περιορισμού πρόσβασης των χρηστών με βάση τα δικαιώματα τους είναι οι κατάλληλες ρυθμίσεις του διαχειριστή στο apache: [48]

<Directory "/var/www/html/users.html">

Options Indexes MultiViews

FollowSymLinks Order deny,allow

Deny from all

AuthName "Valid

user" AuthType Basic

AuthUserFile "/etc/apache2/users.html"

require valid-user

Allow from

192.168.0.1 Satisfy

Any

</Directory>

5.9 Αντιμετώπιση επίθεσης - INSUFFICIENT TRANSPORT LAYER PROTECTION

Η ευπάθεια στο επίπεδο μεταφοράς μπορεί να αποδειχθει καταστροφική για εναν ιστότοπο καθώς μπορεί να προκαλέσει πολλά προβλήματα ασφάλειας. Στη συνέχεια θα αναφέρουμε μερικούς τρόπους προστασίας της σελίδας απο την ευπάθεια αυτή:

- Ενεργοποιημένο πρωτόκολλο SSL σε όλες τις ευαίσθητες σελίδες.
- Εάν είναι δυνατόν, να εφαρμόζεται ένα ξεχωριστό επίπεδο κρυπτογράφησης σε τυχόν ευαίσθητα δεδομένα προτού δοθούν στο κανάλι SSL. Σε περίπτωση που ανακαλυφθούν μελλοντικά τρωτά σημεία στην υλοποίηση SSL, τα κρυπτογραφημένα δεδομένα θα παρέχουν μια δευτερεύουσα άμυνα έναντι της παραβίασης του απορρήτου.
- Δημιουργία ασφαλής σύνδεσης μόνο αφού επαληθευθεί η ταυτότητα του διακομιστή τελικού σημείου χρησιμοποιώντας αξιόπιστα πιστοποιητικά στην αλυσίδα κλειδιών.
- Να μην αποστέλλονται ευαίσθητα δεδομένα μέσω εναλλακτικών καναλιών (π.χ. SMS, MMS ή ειδοποιήσεις).
- Να πραγματοποιείται έλεγχος αν το πιστοποιητικό δεν έχει λήξει και είναι έγκυρο.

Για λόγους απόδοσης, ορισμένοι ιστότοποι χρησιμοποιούν SSL μόνο σε ιδιωτικές ή ευαίσθητες σελίδες. Ωστόσο, το γενικό κόστος του SSL δεν είναι τόσο μεγάλο και σπάνια αξίζει το κέρδος στην απόδοση λαμβάνοντας υπόψη τον κίνδυνο μη κρυπτογράφησης ευαίσθητων πληροφοριών. Ο κάθε ιστότοπος προκειμένου λοιπόν οι χρήστες να νιώθουν ασφάλεια πρέπει να ακολουθήσει τις απαραίτητες ρυθμίσεις ασφάλειας. [49]

Συμπεράσματα

Η δημιουργία περιεχομένου Ιστού με τη βοήθεια CMS δεν ήταν ποτέ πιο ευκολη καθώς υπάρχουν πολλές λύσεις CMS. Ο σχεδιασμός τους είναι εύκολα επεκτάσιμος που βοηθά στην εγκατάσταση σχεδόν οποιασδήποτε λύσης ιστότοπου. Δωρεάν με προσεκτική ανάλυση και χαρακτηριστικα, με σεβασμό στις συγκεκριμένες ανάγκες των χρηστών και της κοινότητας, μπορούμε να δημιουργήσουμε εύκολα ενδιαφέρουσες και περίπλοκες ιστοσελίδες έχοντας ενδιαφέρον περιεχόμενο μπορούμε να δημιουργήσουμε τον ιστότοπο που είναι ελκυστική για τους χρήστες και την κοινότητα του Διαδικτύου.

Αδιαμφισβήτητα όλα τα συστήματα αποτελούνται από γραμμές κώδικα. Προκειμένου να διασφαλιστεί η ασφάλεια των συστημάτων αυτών όμως οι προγραμματιστές και οι χρήστες πρέπει να είναι απαραίτητα πολύ προσεκτικοί προκειμένου να μην πέσουν θύματα επιθέσεων έχοντας ως αποτέλεσμα την εκμετάλλευση ευαίσθητων στοιχείων προκαλόντας μεγάλα προβλήματα όχι μόνο στους διαχειριστές αλλα και στους απλούς χρήστες που επισκέπτονται την ιστοσελίδα αυτή. Ένας προγραμματιστής γράφοντας κώδικα κύριο μέλημα του είναι να πετύχει την μέγιστη δυνατή ασφάλεια χωρίς όμως να τα καταφέρνει πάντα. Οι επιθέσεις XSS και SQL INJECTION όπως είδαμε εκμεταλλεύονται τα λάθη αυτα προκειμένου να προκαλέσουν επίθεση. Ένα ακόμα σημείο που μπορεί να εκμεταλλευτούν οι κακόβουλοι χρήστες είναι οι browser καθιστώντας απαραίτητη την σωστή ρύθμιση τους αλλά και την ορθή διαχείριση των cookies.

Επιπρόσθετα θα πρέπει να υπάρχει στο επίπεδο μεταφοράς αυστηρή κρυπτογράφηση η οποία όμως θα πρέπει να πραγματοποιείται στα σωστά σημεία προκειμένου να αποφεύγονται μεγάλες καθυστερήσεις στο σύστημα.

Τελικά η ασφάλεια των συστημάτων εξαρτάται ολοκληρωτικά από τον ίδιο τον άνθρωπο. Η σωστή κωδικοποίηση, κρυπτογράφηση, οι συχνές ενημερώσεις και έλεγχοι, ο ορθός χειρισμός του συστήματος, η προσθήκη αδειοδότησης αποτελούν μερικά από τα πιο τρωτά σημεία που πρέπει να ακολουθεί ένας προγραμματιστής για να δημιουργήσει ένα ασφαλές περιβάλλον. Όποιο τελικά σύστημα και αν επιλέξει όλα εξαρτώνται από τη χρήση του χρήστη. Μπορεί μερικά από τα πολλά συστήματα να έχουν θεωρηθεί πιο κατάλληλα από κάποια άλλα, πιο εύχρηστα και φιλικά, αλλα τελικά με τη λάθος αντιμετώπιση και διαμόρφωση τους μια επίθεση μπορεί να πραγματοποιηθεί εύκολα.

Βιβλιογραφία

References:

1. Sitecore, What is a CMS (Content Management System), Available: https://www.sitecore.com/
2. Hosting provider, papaki, Τι είναι το CMS; Available: https://www.papaki.com/
3. Andrew V. Royappa (2000), The PHP web application server, in Journal of Computing Sciences in Colleges. March, 2000
4. Database, In Wikipedia. Retrieved March 1, 2022 from https://en.wikipedia.org/wiki/Database
5. Programming language, In Wikipedia. Retrieved August 16, 2021 from https://en.wikipedia.org/wiki/Programming_language
6. Web server, In Wikipedia. Retrieved January 25, 2022 from https://en.wikipedia.org/wiki/Web_server
7. Σύστημα Διαχείρισης Περιεχομένου, In Wikipedia. Retrieved March 16,2021 from https://el.wikipedia.org/wiki/Σύστημα_Διαχείρισης_Περιεχομένου
8. S. Raghunathan; A. Prasad; B.K. Mishra; Hsihui Chang, Open source versus closed source: software quality in monopoly and competitive markets, 6 , November 2005
9. James W. Paulson, Member, IEEE, Giancarlo Succi, Member, IEEE Computer Society, and Armin Eberlein, Member, IEEE Computer Society, An Empirical Study of Open-Source and Closed-Source Software Products APRIL 2004
10. Thomas O'Daniel, Chew Kok Wai, Domain name and site hosting preferences: empirical evidence, Oct 1, 2000
11. Aayush Shukla, Web Hosting August 11, 2020 from https://aayushshukla13.wordpress.com/2020/08/11/example-post-3/
12. Bob Doyle, CMS Genesis: Who Did What When?, Jul 6, 2019 from https://www.thetilt.com/content/cms-genesis-who-did-what
13. SIFTEN HALWAI, Future of CMS Industry: Facts, Figures, Trends and Statistics, 16 August 2021
14. Daniela Louraço1 , Célio Gonçalo Marques2, CMS in Public Administration: A Comparative Analysis, Received: 27 Aug. 2021
15. Jordi Cabot, WordPress: A Content Management System to Democratize Publishing May/June 2018
16. Openhub, Languages Wordpress from https://www.openhub.net/p/wordpress/analyses/latest/languages_summary
17. Codex, Database Description from https://codex.wordpress.org/Database_Description
18. D. Field, "6 Major Tech Companies Have Doubled Their Design Hiring Goals in the Last Half Decade
19. Committer, In Wikipedia. Retrieved Jun 28 from https://en.wikipedia.org/wiki/Committer
20. . A. Brown, "WordPress Hooks Database," 2017
21. WordPress.com and WordPress .org, WordPress, 2018; en.support .wordpress.com/com-vs-org.
22. B. Krogsgard, "Interview with Matt Mullenweg on the WordPress Ecosystem—Draft Podcast," Post Status, 18 Nov. 2017; poststatus .com/interview-matt-mullenweg -wordpress-ecosystem-draft-podcast

23. Joomla! Documentation, Extension types (general definitions) from https://docs.joomla.org/Extension_types_(general_definitions)
24. Ric Shreves John Wiley & Sons, Joomla! Bible Mar 15, 2013
25. Openhub, Languages Joomla! from https://www.openhub.net/p/joomla/analyses/latest/languages_summary
26. Joomla! Developer Network™, Where to Start, Aug 2010 from https://developer.joomla.org/code/cms/history/2010/12/02.html
27. Savan K.Patel; Dr.V.R.Rathod; Jigna B. Prajapati, Performance Analysis of Content Management Systems- Joomla, Drupal and WordPress, 4, May 2011
28. DANIEL KREISS, Dean, Romney, and Drupal: Values and Technological Adoption, MARCH 5, 2012 from https://culturedigitally.org/2012/03/dean-romney-and-drupal-values-and-technological-adoption/
29. Josh Koenig, Growth Graphs, Dec 5, 2006 from https://groups.drupal.org/node/1980
30. Drupal, PHPTemplate theme engine, Last modified: November 1, 2008 from https://web.archive.org/web/20090308030334/http://drupal.org/phptemplate
31. Drupal core - Moderately critical - Denial of Service - SA-CORE-2019-009, 2019 Dec
32. Dennis Fisher, DRUPAL PATCHES ARBITRARY FILE UPLOAD, Dec 23, 2019
33. Keyur Patel, A Survey on Vulnerability Assessment & Penetration Testing for Secure Communication, Aprl 23, 2019
34. Tanjila Farah; Moniruzzaman Shojol; Maruf Hassan; Delwar Alam, Assessment of vulnerabilities of web applications of Bangladesh: A case study of XSS & CSRF, 2016
35. Md. Maruf Hassan*1,2, Shamima Sultana Nipa1 , Marjan Akter1 , Rafita Haque2 , Fabiha Nawar Deepa2 , Mostafijur Rahman1,2, Md. Asif Siddiqui1 , Md. Hasan Sharif1, Broken Authentication and Session Management Vulnerability: A Case Study Of Web Application, 2016
36. Y. Stefinko, A. Piskozub and R. Banakh, "Manual and automated penetration testing. Benefits and drawbacks. Modern tendency,"13th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET), Lviv, 2016, pp. 488-491
37. Tatiana Alexenko; Mark Jenne; Suman Deb Roy; Wenjun Zeng, Cross-Site Request Forgery: Attack and Defense, Published in: 2010 7th IEEE Consumer Communications and Networking Conference
38. Rupali D. Kombade, Dr. B.B. Meshram, CSRF Vulnerabilities and Defensive Techniques, 2012
39. KumarShrestha, Ajay ; Singh Maharjan, Pradip ; Paudel, Santosh, Identification and Illustration of Insecure Direct Object References and their Countermeasures,March 2015
40. Sergio Loureiro, Security misconfigurations and how to prevent them, 2 Nov 2021
41. Juanru Li , Zhiqiang Lin , Juan Caballero , Yuanyuan Zhang , Dawu Gu, K-Hunt: Pinpointing Insecure Cryptographic Keys from Execution Traces, October 2018
42. Hasty Atashzar; Atefeh Torkaman; Marjan Bahrololum; Mohammad H. Tadayon, A survey on web application vulnerabilities and countermeasures, 2011

43. Abdelhamid Makiou; Youcef Begriche; Ahmed Serhrouchni, Improving Web Application Firewalls to detect advanced SQL injection attacks,Published in: 2014
44. Mike Ter Louw; V.N. Venkatakrishnan, Blueprint: Robust Prevention of Cross-site Scripting Attacks for Existing Browsers,Published in: 2009 30th IEEE Symposium on Security and Privacy
45. B. Newton, "The hyper-growth of web 2.0 applications," Mar. 2008, seminar
46. T. Jim, N. Swamy, and M. Hicks, "Defeating script injection attacks with browser-enforced embedded policies," in 16th International World Wide Web Conference, Banff, AB, Canada, May 2007.
47. How to deny access to a specific file on your site via .htaccess, April 5, 2017, from https://www.plothost.com/kb/how-to-deny-access-to-a-specific-file-on-your-site-via-htaccess/
48. Apache - Blocking the access to URLs from https://techexpert.tips/apache/apache-blocking-access-urls/
49. owasp, Insufficient Transport Layer Protection Threat Agents φρομ https://owasp.org/www-project-mobile-top-10/2014-risks/m3-insufficient-transport-layer-protection