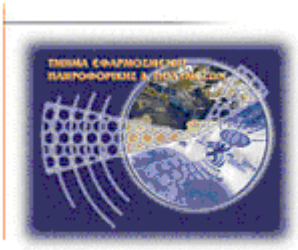
Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων



Πτυχιακή εργασία

Συστήματα Διαχείρισης Περιεχομένου (Content Management Systems): Μελέτη και αξιολόγηση ασφαλείας

Βασιλική Κουτσοτόλιου (ΑΜ: 1490)
E-mail: vassouko@hotmail.com

Ηράκλειο - Δεκέμβριος 2010

Επόπτης Καθηγητής: Φυσαράκης Κωνσταντίνος

ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ:

Βεβαιώνω ότι είμαι συγγραφέας αυτής της πτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην πτυχιακή εργασία. Επίσης έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης βεβαιώνω ότι αυτή η πτυχιακή εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τις απαιτήσεις του προγράμματος σπουδών του Τμήματος Εφαρμοσμένης Πληροφορικής και Πολυμέσων του Τ.Ε.Ι. Κρήτης.

Κουτσοτόλιου Βασιλική, Ηράκλειο 2010

Ευχαριστίες

Θα ήθελα να ευχαριστήσω την οικογένεια και τους φίλους μου για την αμέριστη υποστήριξη κατά τη διάρκεια εκπόνησης της πτυχιακής μου εργασίας, καθώς και τον κ.Αντώνη Παπαγρηγορίου (επιστημονικός και εργαστηριακός συνεργάτης του Γ.Τ.Θ.Ε. του ΤΕΙ Ηρακλείου) για όση βοήθεια μου προσέφερε για την κατανόηση του Joomla! και των εργαλείων του.
Επίσης θα ήθελα να ευχαριστήσω τον επόπτη καθηγητή κ.Φυσαράκη Κωνσταντίνο για την πολύτιμη καθοδήγηση και υποστήριξη σε όλη τη διάρκεια εκπόνησης αυτής της εργασίας.

Περίληψη

Στην παρούσα πτυχιακή θα ασχοληθούμε με την μελέτη και την αξιολόγηση των δημοφιλέστερων συστημάτων διαχείρισης περιεχομένου με έμφαση στην ασφάλεια τους και θα συγκριθούν ως προς τις δυνατότητες ασφαλείας που προσφέρουν για τους υπό δημιουργία ιστοτόπους.

Έπειτα θα επιλέξουμε ένα από τα δημοφιλέστερα συστήματα διαχείρισης περιεχομένου και θα δημιουργήσουμε έναν ιστότοπο (συγκεκριμένα ηλεκτρονικό κατάστημα), θα εντοπίσουμε τις αδυναμίες του συστήματος και του δικτύου και θα βρούμε τους τρόπους θωράκισης.

Abstract

This thesis will deal with the study and evaluation of the top content management systems with emphasis on safety and will be compared to the security features offered on the sites under construction.

Then, we will choose one of the most popular content management systems to create a website (ie online shop) and after we will identify weaknesses in the system and network and find ways of shielding.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

ΠΙΝΑΚΑΣ ΠΙΝΑΚΩΝ

ΕΙΣΑΓΩΓΗ

ΣΚΟΠΟΣ ΤΗΣ ΠΤΥΧΙΑΚΗΣ

Σε αυτή την πτυχιακή θα ασχοληθούμε με την μελέτη και αξιολόγηση των δημοφιλέστερων open source CMS με έμφαση στην ασφάλειά τους. Θα μελετηθούν και θα συγκριθούν ως προς τις δυνατότητες ασφαλείας που προσφέρουν για τους υπό δημιουργία ιστοτόπους. Κατόπιν, θα γίνει ανάπτυξη ενός πρωτότυπου ιστοτόπου ηλεκτρονικού εμπορίου με χρήση μίας από τις υπό εξέταση πλατφόρμες και θα ακολουθήσει αξιολόγηση της ασφάλειας του τελικού προϊόντος καθώς και τρόποι θωράκισής του από τις επιθέσεις που θα εντοπιστούν.

Συγκεκριμένα θα αναλυθούν τα εξής:

- ✓ Γενικές πληροφορίες για τα Συστήματα Διαχείρισης Περιεχομένου (CMS).
- ✓ Που βασίζεται ένα CMS.
- ✓ Χαρακτηριστικά της επιλεγμένης πλατφόρμας και οι προδιαγραφές εγκατάστασής της
- ✓ Μέθοδοι πληρωμής που υποστηρίζει το εργαλείο που χρησιμοποιούμε.
- ✓ Τα επικίνδυνα σημεία.
- ✓ Οι τεχνικές που χρησιμοποιούνται για την διασφάλιση του ιστοτόπου που θα αναπτυχθεί.
- ✓ Τεχνικές διασφάλισης της εξεταζόμενης πλατφόρμας για τον υποκατασκευή ιστότοπο.

ΚΕΦΑΛΑΙΟ 1

1.1 ΣΥΣΤΗΜΑΤΑ ΔΙΑΧΕΙΡΙΣΗΣ ΠΕΡΙΕΧΟΜΕΝΟΥ (CMS)

1.1.1 Ορισμός του CMS:

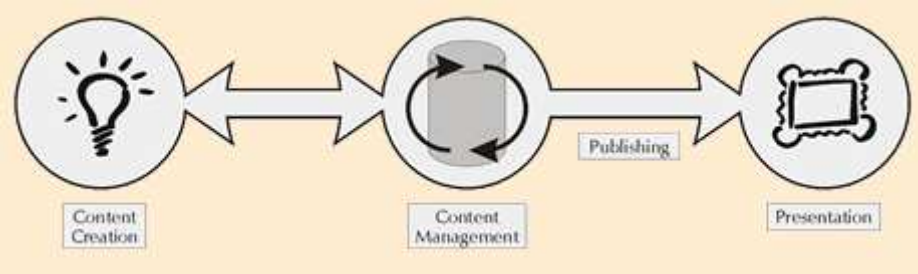
Τα **Συστήματα Διαχείρισης Περιεχομένου** (**Content Management Systems**, **CMS**) είναι διαδικτυακές εφαρμογές που επιτρέπουν την εύκολη δημιουργία και δημοσίευση ιστοσελίδων και την online τροποποίηση του περιεχομένου ενός δικτυακού τόπου.

Οι διαχειριστές μέσω του διαδικτύου ενημερώνουν το περιεχόμενο στο CMS, το οποίο είναι εγκατεστημένο σ' ένα διακομιστή. Οι αλλαγές αυτές γίνονται αυτόματα διαθέσιμες πάλι μέσω του διαδικτύου, σε όλους τους επισκέπτες και χρήστες του δικτυακού τόπου.[1]

Οι εφαρμογές διαχείρισης περιεχομένου επιτρέπουν οποιαδήποτε αλλαγή του περιεχόμενου χωρίς να είναι απαραίτητες ειδικές γνώσεις σχετικές με τη δημιουργία ιστοσελίδων ή γραφικών, καθώς συνήθως τα κείμενα γράφονται μέσω κάποιων online WYSIWYG ("What You See Is What You Get") html editors, ειδικών δηλαδή κειμενογράφων, παρόμοιων με το MS Word, που επιτρέπουν τη μορφοποίηση των κειμένων όποτε υπάρχει ανάγκη.

Η λειτουργικότητα ενός συστήματος διαχείρισης περιεχομένου, μπορεί να αναλυθεί στις εξής βασικές κατηγορίες:

- δημιουργία περιεχομένου (content creation)
- διαχείριση περιεχομένου (content management)
- δημοσίευση (publishing)
- παρουσίαση (presentation)



Εικόνα 1: Λειτουργικότητα ενός Συστήματος Διαχείρισης Περιεχομένου (CMS)

[1] http://el.wikipedia.org/wiki/Σύστημα_Διαχείρισης_Περιεχομένου

Το Σύστημα Διαχείρισης Περιεχομένου, είναι ένα πρόγραμμα ειδικά σχεδιασμένο για τη διαχείριση ιστοτόπων. Δημιουργείται και εγκαθίσταται από τους σχεδιαστές ιστοσελίδων, αλλά προορίζεται για δική μας χρήση.

Χρησιμοποιώντας τον browser της επιλογής μας, εισάγουμε το νέο κείμενο, το αποθηκεύουμε και με αυτόν τον εύκολο και εύχρηστο τρόπο ενημερώνεται το περιεχόμενο του site μας. Με απλό τρόπο επίσης, μπορούμε να προσθέτουμε σελίδες, να διαγράφουμε και γενικότερα να διαχειριζόμαστε τη δομή του site μας.

Το Σύστημα Διαχείρισης Περιεχομένου μας διευκολύνει αυτοματοποιώντας διάφορες διαδικασίες, όπως η διατήρηση της ίδιας εμφάνισης των σελίδων μας σε όλο το site, επίσης την εύκολη δημιουργία των σχετικών μενού, συνδέσμων κλπ.. Με την ύπαρξη αρκετών άλλων εργαλείων διαχείρισης μας επιτρέπει να επικεντρωθούμε στο περιεχόμενο και όχι στην τεχνολογία και στις γλώσσες προγραμματισμού.

Τρεις είναι οι βασικές κατηγορίες των CMS:

- ✓ **Enterprise CMS** : αναφέρεται στις τεχνολογίες, τις στρατηγικές, τις μεθόδους και τα εργαλεία που χρησιμοποιούνται για την συλλογή, διαχείριση, αποθήκευση, διατήρηση αλλά και παράδοση των περιεχομένων και των εγγράφων που σχετίζονται με έναν οργανισμό και τις διαδικασίες του. Τα Enterprise CMS εργαλεία επιτρέπουν τη διαχείριση των πληροφοριών ενός οργανισμού.
- ✓ **Component CMS** : διαχειρίζεται τα περιεχόμενα σε ένα σπυρωτό επίπεδο περιεχομένων και όχι σε επίπεδο εγγράφου. Κάθε συστατικό αντιπροσωπεύει ένα μεμονωμένο θέμα, έννοια ή περιουσιακό στοιχείο (π.χ. εικόνα, πίνακας, περιγραφή προϊόντος). Τα συστατικά συγκεντρώνονται σε πολλαπλά περιεχόμενα και μπορούν να εμφανιστούν σαν ψηφιακά ή παραδοσιακά έγγραφα. Κάθε συστατικό έχει το δικό του κύκλο ζωής(ιδιοκτήτης, έκδοση, έγκριση, χρήση) και μπορεί να εντοπιστεί μεμονωμένα ή ως μέρος μιας σύναξης. Το CCM μπορεί να είναι ένα ξεχωριστό σύστημα ή να είναι μια λειτουργία ενός άλλου τύπου συστήματος διαχείρισης περιεχομένου (π.χ., ECM ή Web Content Management).
- ✓ **Web CMS** : χρησιμοποιείται για να δημιουργήσει, να επεξεργαστεί, να διαχειριστεί και να δημοσιεύσει ιστοσελίδες στο διαδίκτυο. Τα συστήματα διαχείρισης περιεχομένου μπορούν να χρησιμοποιηθούν για να κατασκευάσουν ιστοτόπους που μπορούν να καλύψουν σχεδόν όλη την γκάμα των ενδιαφερομένων (π.χ. Εταιρικούς, εκπαιδευτικούς, ηλεκτρονικά καταστήματα κ.ά.) . Το περιεχόμενο που μπορεί να χρησιμοποιηθεί περιλαμβάνει κείμενα, εικόνες, ήχο, video, ηλεκτρονικά αρχεία και γενικά οτιδήποτε μπορεί να διανεμηθεί μέσω του διαδικτύου.

Ένα σύστημα διαχείρισης περιεχομένου (CMS) πρέπει να υποστηρίζει:

- Εύκολη διαχείριση περιεχομένου μέσω ενός browser.
- Διαφορετικούς ρόλους και επίπεδα για τους χρήστες.
- Δυνατότητα δημοσίευσης περιεχομένου από χρήστες έπειτα από την έγκριση του διαχειριστή.
- Δυνατότητα κατηγοριοποίησης του περιεχομένου ώστε να είναι ευκολότερη η διαχείριση του.

- Διαχωρισμό περιεχομένου και εμφάνισης (να είναι εφικτό δηλαδή να γίνει οποιαδήποτε εικαστική παρέμβαση στη σελίδα, π.χ. αλλαγή του φόντου και η εφαρμογή του να γίνει σε όλες τις σελίδες).

1.2 ΕΝΝΟΙΕΣ ΟΡΩΝ

1.2.1 Ελεύθερο Λογισμικό

Το ελεύθερο λογισμικό όπως ορίζεται από το Ίδρυμα Ελευθέρου Λογισμικού (Free Software Foundation), είναι λογισμικό που μπορεί να χρησιμοποιηθεί, αντιγραφεί, μελετηθεί, τροποποιηθεί και αναδιανεμηθεί χωρίς περιορισμό. Η ελευθερία από τέτοιους περιορισμούς είναι βασικό στοιχείο στην ιδέα του "ελεύθερου λογισμικού", έτσι ώστε το αντίθετο του ελεύθερου λογισμικού να είναι το ιδιόκτητο λογισμικό και όχι το λογισμικό που πωλείται για κέρδος, όπως το εμπορικό λογισμικό. Το ελεύθερο λογισμικό ορισμένες φορές αναφέρεται και σαν ανοιχτό λογισμικό ή λογισμικό ανοιχτού κώδικα αλλά οι δύο έννοιες δεν είναι ταυτόσημες.

1.2.2 Άδειες Ελεύθερου Λογισμικού

Σύμφωνα με την ισχύουσα νομοθεσία περί πνευματικής ιδιοκτησίας, η ελεύθερη αντιγραφή, διανομή αλλά και οποιαδήποτε τροποποίηση του λογισμικού δεν επιτρέπεται. Οι εκδόσεις ελεύθερου λογισμικού ωστόσο, λόγω της συγκεκριμένης νομοθεσίας, κάνουν χρήση της ειδικής άδειας (free software license). Σύμφωνα λοιπόν με την free software license, παραχωρείται το δικαίωμα αντιγραφής, τροποποίησης και αναδιανομής του λογισμικού προς όλους τους χρήστες.

Το ίδρυμα Ελεύθερου Λογισμικού ορίζει τις εξής ελευθερίες για τις άδειες χρήσης ελεύθερου λογισμικού:

- Ελευθερία 0: Ελευθερία χρήσης του προγράμματος για οποιονδήποτε σκοπό.
- Ελευθερία 1: Ελευθερία μελέτης και τροποποίησης του προγράμματος.
- Ελευθερία 2: Ελευθερία αντιγραφής του προγράμματος.
- Ελευθερία 3: Ελευθερία βελτίωσης του προγράμματος και επανέκδοσης του, προς το συμφέρον της κοινότητας των χρηστών.

Οι ελευθερίες 1 και 3 προϋποθέτουν την πρόσβαση των χρηστών στον πηγαίο κώδικα του λογισμικού.

1.2.3 Τι σημαίνει να είναι κάτι Ανοικτού Κώδικα;

Όλες οι εφαρμογές ανοιχτού κώδικα επιτρέπουν την πρόσβαση και την αλλαγή του πηγαίου κώδικα, είναι διαθέσιμες στο διαδίκτυο (συχνά) χωρίς κανένα κόστος και τυπικά απαιτούν κάποιες τεχνικές γνώσεις για το «στήσιμο» και τη λειτουργία τους. Συχνά συνοδεύονται από plug-ins τα οποία δημιουργούνται και προσφέρονται από

μια κοινότητα χρηστών και προγραμματιστών, οι όποιοι είναι υπεύθυνοι και για την υποστήριξη των εφαρμογών ανοιχτού κώδικα.

1.2.4 Εμπιστευτικότητα (Confidentiality), Ακεραιότητα (Integrity), Διαθεσιμότητα Υπηρεσιών (Availability), Μη αποποίηση ευθύνης (Non-repudiation)

Εμπιστευτικότητα (Confidentiality)

Η εμπιστευτικότητα αφορά στην προστασία της ιδιωτικότητας του χρήστη και την αποφυγή κλοπής δεδομένων που είναι είτε αποθηκευμένα είτε διακινούμενα στο διαδίκτυο.

Η ιδιωτικότητα του χρήστη καλύπτει κάθε πτυχή των προσωπικών του δεδομένων που εν δυνάμει διακινούνται στο διαδίκτυο: από τη διεύθυνση του ηλεκτρονικού του ταχυδρομείου ως το είδος των ιστοσελίδων που επισκέπτεται και τον τρόπο που πλοηγείται σε αυτές.

Ακεραιότητα (Integrity)

Η ακεραιότητα είναι έννοια συνυφασμένη με την αποφυγή μη εξουσιοδοτημένης τροποποίησης των δεδομένων που ανταλλάσσονται, είτε βρίσκονται υπό διακίνηση είτε είναι αποθηκευμένα.

Διαθεσιμότητα Υπηρεσιών (Availability)

Τελευταίος βασικός στόχος των μέτρων ασφάλειας είναι η αδιάλειπτη διαθεσιμότητα των διαδικτυακών υπηρεσιών, με επαρκή ποιότητα ώστε να διασφαλίζεται στο χρήστη η ορθή παροχή τους. Η ποιότητα παροχής υπηρεσιών είναι ουσιαστική καθώς ενδέχεται να υπάρχει διαθεσιμότητα υπηρεσιών αλλά, παραδείγματος χάρη, με χαμηλή ταχύτητα διασύνδεσης ή με τροποποιημένα στοιχεία ώστε να είναι ανώφελη ή και ζημιογόνος η υπηρεσία για τον χρήστη.

Μη αποποίηση ευθύνης (Non-repudiation)

Η αδυναμία αποποίησης ευθύνης ενός χρήστη – ή οργανισμού – του διαδικτύου είναι σημαντική σε κάθε μορφή διαδικτυακής επικοινωνίας: είτε αυτή αφορά, για παράδειγμα, τον αποστολέα ενός υβριστικού μηνύματος είτε τον πάροχο ζημιογόνου διαδικτυακού λογισμικού.

1.2.5 Hacker, Cracker, Owned, Exploit

Hacker: Είναι ένα άτομο που μαθαίνει για κάποια τεχνολογία για να μπορέσει να γράψει ένα καλύτερο κώδικα, να κατασκευάσει καλύτερα μηχανήματα ή για να το χρησιμοποιήσει είτε σαν επάγγελμα είτε σαν χόμπι.[2]

[2] http://el.wikipedia.org/wiki/%CE%A7%CE%AC%CE%BA%CE%B5%CF%81

Cracker: Αυτό είναι ένα άτομο (ή ομάδα ατόμων) που αποπειράται να αποκτήσει πρόσβαση σε υπολογιστικό σύστημα για την οποία όχι μόνο δε διαθέτει εξουσιοδότηση, αλλά με στόχο να το βλάψει με οποιοδήποτε τρόπο. Οι κράκερ είναι εξ'ορισμού κακόβουλοι, αντίθετα προς τους χάκερ, ενώ διαθέτουν και πολλά εργαλεία για τις κακόβουλες ενέργειές τους.

Owned: Πρόκειται για την κατάσταση μιας μηχανής μετά από μια επίθεση ενός Cracker, ο οποίος έχει «σπάσει» με επιτυχία άμυνες της και έχει τοποθετήσει έναν κωδικό για να «ακούει», να «κλέβει», να κατασκοπεύει, ή να καταστρέψει το μηχάνημα.

Exploit: Είναι ένα τμήμα λογισμικού, ένα «κομμάτι» δεδομένων ή μία ακολουθία δεδομένων που μπορούν να εκμεταλλευτούν κάποιο bug, κάποια βλάβη ή κάποια ευπάθεια με σκοπό να προκληθούν ανεπιθύμητες ή απρόβλεπτες συμπεριφορές στο software ή στο hardware ενός ηλεκτρονικού υπολογιστή.[3]

[3] http://en.wikipedia.org/wiki/Exploit_%28computer_security%29

ΚΕΦΑΛΑΙΟ 2

2.1 ΔΥΝΑΤΟΤΗΤΕΣ ΚΑΙ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΩΝ CMS

2.1.1 Κατηγορίες Συστημάτων Διαχείρισης Περιεχομένου

Τα WEB-CMS χωρίζονται σε τρεις κατηγορίες:

1. CMS κλειστού κώδικα
2. CMS ανοιχτού κώδικα
3. Παραμετροποιημένα CMS βασισμένα σε πλαίσια ανοιχτού κώδικα

2.1.2 Πλεονεκτήματα, χαρακτηριστικά και δυνατότητες ενός ολοκληρωμένου CMS

Επιγραμματικά, ένα ολοκληρωμένο CMS έχει τα εξής χαρακτηριστικά, πλεονεκτήματα και δυνατότητες:

- Μπορεί και παρέχει τη δυνατότητα της διαχείρισης-συντήρησης ενός ιστοτόπου από απλούς χρήστες χωρίς να απαιτεί την εμπλοκή ειδικού τεχνικού προσωπικού. Ο διαχειριστής του μπορεί να επικεντρωθεί στο περιεχόμενο και όχι στην τεχνολογία, με αποτέλεσμα την ταυτόχρονη ενημέρωση από πολλούς χρήστες και διαφορετικούς υπολογιστές, όπως επίσης την γρήγορη ενημέρωση, διαχείριση και αρχειοθέτηση του περιεχομένου του ιστοτόπου.
- Αυτοματοποιούνται εργασίες ρουτίνας π.χ. ίδια μορφοποίηση (layout) σε όλες τις ιστοσελίδες, ενώ οι επιλογές (menus) και γενικότερα η πλοήγηση αναπαράγεται αυτόματα.
- Παρέχει απλά εργαλεία (επεξεργαστές σαν το Word) για τη δημιουργία του περιεχομένου, τα οποία είναι εύκολα στη χρήση και έτσι επιτυγχάνεται άμεση εμφάνιση του τελικού αποτελέσματος
- Δίνει τη δυνατότητα αναζήτησης του περιεχομένου που καταχωρείται και την αυτόματη δημιουργία αρχείου.
- Παρέχει ασφάλεια και προστασία του σχεδιασμού του site από λανθασμένες ενέργειες, που θα μπορούσαν να δημιουργήσουν προβλήματα στην εμφάνισή του.
- Διαχωρίζει το περιεχόμενο από το σχεδιασμό και την πλοήγηση (navigation) του ιστοτόπου και επίσης παρέχει τη δυνατότητα αλλαγής σχεδιασμού ή τρόπου πλοήγησης χωρίς να είναι απαραίτητη η ενημέρωση όλων των σελίδων από τον ίδιο το χρήστη.
- Αυτοματοποιεί τη δημιουργία των συνδέσμων μεταξύ των σελίδων προς αποφυγή προβληματικών ανύπαρκτων σελίδων (404 error pages).
- Όλες τις τεχνικές λεπτομέρειες τις χειρίζεται το ίδιο το σύστημα, επιτρέποντας έτσι σε οποιονδήποτε να διαχειριστεί και να ενημερώνει τον ιστότοπο.

- Πετυχαίνει μικρότερο φόρτο στον εξυπηρετητή (server) και χρήση λιγότερου χώρου, αφού δεν υπάρχουν πολλές επαναλαμβανόμενες στατικές σελίδες, από τη στιγμή που η ανάπτυξη των σελίδων γίνεται δυναμικά.
- Όλο το περιεχόμενο καταχωρείται στην/στις βάσεις δεδομένων, τις οποίες μπορούμε πιο εύκολα και γρήγορα να τις προστατεύσουμε τηρώντας αντίγραφα ασφαλείας.
- Όλα τα δυναμικά χαρακτηριστικά του συστήματος επιτρέπουν στον ιστότοπο να αναπτύσσεται συγχρόνως με την εκάστοτε επιχείρηση.
- Έχουμε μεγαλύτερη ομοιομορφία και συνοχή στον ιστότοπο, καθώς επίσης και βελτιωμένο σύστημα πλοήγησης και αυξημένη ευελιξία.
- Μειώνονται έξοδα συντήρησης-διαχείρισης.
- Υπάρχει αυξημένη ικανότητα ανάπτυξης.

2.2 ΠΕΡΙΓΡΑΦΗ ΤΩΝ ΔΥΟ ΒΑΣΙΚΩΝ ΚΑΤΗΓΟΡΙΩΝ CMS

2.2.1 CMS κλειστού κώδικα

Τα πλεονεκτήματα τους είναι ότι έχουν εμπορική υποστήριξη (προσδιορισμένες υπηρεσίες), είναι συνήθως ετοιμοπαράδοτα, έχουν καλύτερη τεκμηρίωση και εκπαίδευση και τέλος παρέχουν ασφάλεια.

Το μεγαλύτερο μειονέκτημα τους είναι το κόστος. Βασικό κόστος, κόστος παραμετροποίησης και το κόστος της ολοκλήρωσης με υπάρχοντα εταιρικά συστήματα.

Τα πιο δημοφιλή CMS Κλειστού Κώδικα είναι τα εξής:

- Vignette Content Management
- IBM Workplace Web Content Management
- Jalios JCMS
- Powerfront CMS

2.2.2 Αναλυτικότερα για τα πιο δημοφιλή CMS κλειστού κώδικα:

Vignette Content Management: Τα προϊόντα της Vignette βοηθούν τις επιχειρήσεις να αποκτήσουν τις πληροφορίες που χρειάζονται και να τις διαχειρίζονται. Η Vignette είναι μια έμπειρη εταιρία αποδοτικότητας, η οποία στοχεύει στην αύξηση της παραγωγικότητας, στη μείωση του κόστους και στην βελτίωση της εμπειρίας του χρήστη. Οι Intranet, extranet και internet λύσεις της, συμπεριλαμβάνουν portal, integration, enterprise content management και δυνατότητες συνεργασίας που μπορούν να αποδώσουν μοναδικά προτερήματα.

IBM Workplace Web Content Management: Το συγκεκριμένο προϊόν παρέχει μία μεγάλη γκάμα λειτουργιών όπως: personalization, web content management, διαχείριση εγγράφων και λειτουργίες συνεργασίας και παραγωγικότητας στα πλαίσια της επεκτάσιμης υποδομής του WebShere Portal.

Jalios JCMS: Είναι ένα enterprise content management (ECM) που αναπτύχθηκε από την Jalios και περιλαμβάνει, μεταξύ άλλων, και τα ακόλουθα χαρακτηριστικά: διαχείρισης περιεχομένου, διαχείριση εγγράφων, collaboration, workflow και πύλες.

Powerfront CMS: Το PowerFront παρέχει μια ολοκληρωμένη λύση διαχείρισης περιεχομένου που να περιλαμβάνει: τη διαχείριση περιεχομένου, τον σχεδιασμό ιστοσελίδων, θέματα ασφάλειας, το ηλεκτρονικό εμπόριο, procurement, reporting options και υποστήριξη. Στόχος της είναι η υποστήριξη ιστοσελίδων των επιχειρήσεων, intranets, extranets ή procurement websites.

2.2.3 CMS ανοιχτού κώδικα

Τα Συστήματα Διαχείρισης Περιεχομένου ανοιχτού κώδικα έχουν χαμηλό κόστος, καθώς πληρώνεις για την υπηρεσία/υποστήριξη και όχι για το ίδιο το λογισμικό. Έχουν ευκολία στην παραμετροποίηση και στην ολοκλήρωση με υπάρχοντα λογισμικά και έχουν υποστήριξη από την προαναφερθείσα κοινότητα χρηστών και προγραμματιστών. Η διόρθωση σφαλμάτων είναι ταχεία και υπάρχει μελλοντική εξασφάλιση συνέχειας. Τέλος μπορούν να δοκιμαστούν πριν αγοραστούν.

Τα μειονεκτήματά τους είναι εξίσου αρκετά με τα πλεονεκτήματα. «Ελεύθερο Λογισμικό» δεν συνεπάγεται και λογισμικό χωρίς κόστος, υπάρχει έλλειψη εμπορικής υποστήριξης, εστιάζεται περισσότερο στην τεχνική αρχιτεκτονική και στο σύνολο των χαρακτηριστικών, έχει φτωχή χρηστικότητα γιατί δεν λαμβάνει υπόψη την εμπειρία του χρήστη και υπάρχει έλλειψη τεκμηρίωσης.

Εικόνα 2: Τα πιο δημοφιλή CMS

Τα πιο δημοφιλή CMS Ανοικτού Κώδικα είναι τα εξής:

- Joomla
- Drupal
- Plone
- Typo3
- Xoops
- WordPress

2.2.4 Αναλυτικότερα για τα πιο δημοφιλή CMS ανοιχτού κώδικα:

Το **Joomla!** [4] είναι ένα δωρεάν σύστημα διαχείρισης περιεχομένου. Χρησιμοποιείται για τη δημοσίευση περιεχομένου στον παγκόσμιο ιστό (World Wide Web) και σε τοπικά δίκτυα - intranets. Είναι γραμμένο σε PHP και αποθηκεύει τα δεδομένα του στη βάση MySQL. Το βασικό χαρακτηριστικό του είναι ότι οι σελίδες που εμφανίζει είναι δυναμικές, δηλαδή δημιουργούνται την στιγμή που ζητούνται. Ένα σύστημα διακομιστή (server), όπως είναι ο Apache, λαμβάνει τις αιτήσεις των χρηστών και τις εξυπηρετεί.

Με ερωτήματα προς τη βάση λαμβάνει δεδομένα τα οποία μορφοποιεί και αποστέλλει στον εκάστοτε φυλλομετρητή (web browser) του χρήστη. Το Joomla! έχει και άλλες δυνατότητες εμφάνισης όπως η προσωρινή αποθήκευση σελίδας, RSS feeds, εκτυπώσιμες εκδόσεις των σελίδων, ειδήσεις, blogs, δημοσκοπήσεις, έρευνες, καθώς και πολύγλωσση υποστήριξη των εκδόσεών του.

Το **Drupal** είναι ένα αρθρωτό σύστημα διαχείρισης περιεχομένου ανοικτού/ ελεύθερου λογισμικού, γραμμένο στη γλώσσα προγραμματισμού PHP. Το Drupal, όπως πολλά σύγχρονα CMS, επιτρέπει στο διαχειριστή συστήματος να οργανώνει το περιεχόμενο, να προσαρμόζει την παρουσίαση, να αυτοματοποιεί διαχειριστικές εργασίες και να διαχειρίζεται τους επισκέπτες του ιστοτόπου και αυτούς που συνεισφέρουν. Παρόλο που υπάρχει μια πολύπλοκη προγραμματιστική διεπαφή, οι περισσότερες εργασίες μπορούν να γίνουν με λίγο ή και καθόλου προγραμματισμό. Το Drupal ορισμένες φορές περιγράφεται ως "υποδομή για εφαρμογές ιστού", καθώς οι δυνατότητές του προχωρούν παραπέρα από τη διαχείριση περιεχομένου, επιτρέποντας ένα μεγάλο εύρος υπηρεσιών και συναλλαγών.

Το Drupal μπορεί να εκτελεστεί σε διάφορες πλατφόρμες, συμπεριλαμβανομένων των λειτουργικών συστημάτων Windows, Mac OS X, Linux, FreeBSD, ή οποιασδήποτε πλατφόρμας που υποστηρίζει είτε το διακομιστή ιστοσελίδων Apache HTTP Server (έκδοση 1.3+), είτε το Internet Information Services (έκδοση IIS5+), καθώς επίσης και τη γλώσσα προγραμματισμού PHP (έκδοση 4.3.3+). Το Drupal απαιτεί μια βάση δεδομένων όπως η MySQL και η PostgreSQL για την αποθήκευση του περιεχομένου και των ρυθμίσεών του.

Το **Plone** είναι βασισμένο στον κορυφαίο διακομιστή εφαρμογών Zope. Μπορεί να χρησιμοποιηθεί για οποιοδήποτε είδος ιστοσελίδας (για παράδειγμα blogs, e-commerce, internal websites). Είναι επίσης εύχρηστο για συστήματα δημοσίευσης εγγράφων. Προσφέρει ευελιξία και προσαρμοστικότητα στη ροή εργασιών,

[4] http://en.wikipedia.org/wiki/Joomla

ασφάλεια, επεκτασιμότητα και ευχρηστία. Έχει κυκλοφορήσει από την GNU και έχει σχεδιαστεί να είναι επεκτάσιμο. Αξίζει να σημειωθεί πως το layout του "MONOBOOK" της MediaWiki είναι βασισμένο στα style sheets του Plone.

Το **TYPO3** είναι ένα Επαγγελματικό Σύστημα Διαχείρισης Δικτυακού Περιεχομένου ανοικτού κώδικα για εταιρικούς σκοπούς στο διαδίκτυο ή σε ενδοδίκτυο (intranet). Προσφέρει πλήρη ευελιξία και επεκτασιμότητα υιοθετώντας πολλά προχωρημένα χαρακτηριστικά. Είναι κατάλληλο για την ανάπτυξη μεγάλων δικτυακών τόπων με υψηλή επισκεψιμότητα. Επιτρέπει τη δημιουργία και διαχείριση νέων τύπου δεδομένων/οντοτήτων και μεταδεδομένων για τις οντότητες αυτές, όπως και τη δυνατότητα να δημιουργούνται εύκολα νέες εφαρμογές διαχείρισης αυτών των οντοτήτων.

Όλη η παραμετροποίηση του συστήματος που έχει γίνει είτε με τη μορφή δημιουργίας νέων οντοτήτων είτε με τη μορφή συγγραφής κώδικα που ακολουθεί τις παραδοχές του TYPO3 δεν χρειάζεται καμία τροποποίηση σε μελλοντική αναβάθμιση. Έχει δοκιμαστεί σε δικτυακούς τόπους μεγάλης εμβέλειας και έχει αποδειχθεί αρκετά ισχυρό. Ταυτόχρονα έχει τη δυνατότητα σύνδεσης με οποιαδήποτε βάση δεδομένων (MySQL, Oracle, PostgreSQL).

Το **Xoops** είναι το ακρωνύμιο των "eXtensible Object Oriented Portal System". Χρησιμοποιεί μια σπονδυλωτή αρχιτεκτονική που επιτρέπει στους χρήστες του να προσαρμόσουν να ενημερώνουν και να διαφοροποιήσουν θεματικά τους ιστοχώρους τους. Είναι γραμμένο σε PHP και κυκλοφορεί υπό τους όρους της GNU Γενικής Δημόσιας Άδειας (GPL).

Το **WordPress**[5] είναι μια πλατφόρμα δημιουργίας προσωπικού ιστολογίου (blog). Οι δυνατότητες του είναι προσθήκη εικόνων, προσαρμοσμένα πεδία ενώ υπάρχει η δυνατότητα προσθήκης άπειρων δυνατοτήτων από μια πολύ μεγάλη βιβλιοθήκη προσθέτων (Plugins). Όσον αφορά την εμφάνιση υπάρχουν χιλιάδες θέματα για επιλογή αλλά και για προσαρμογή στο δικό σας ιστολόγιο. Όλα αυτά είναι δωρεάν και το λογισμικό του WordPress είναι GPL.

[5] http://www.wordpress.gr/about/

ΚΕΦΑΛΑΙΟ 3

ΣΥΓΚΡΙΣΗ ΕΠΙΚΡΑΤΕΣΤΕΡΩΝ CMS ΚΑΙ ΕΠΙΛΟΓΗ ΕΝΟΣ

Το τελευταίο διάστημα η επιλογή ενός συστήματος διαχείρισης δεν είναι εύκολη, καθώς οι προτάσεις είναι πολλές και εξίσου αξιόλογες. Το καθένα από τα CMS έχει πλεονεκτήματα και μειονεκτήματα, έτσι η καλύτερη επιλογή είναι αυτή η οποία καλύπτει τις απαιτούμενες προδιαγραφές των υπό κατασκευή ιστοτόπων. Σύμφωνα με έρευνες που έγιναν στο διαδίκτυο, τα τρία επικρατέστερα open source CMS είναι τα: Drupal, Joomla! και WordPress. Ας τα συγκρίνουμε λοιπόν!

Εικόνα 3: Τα τρία CMS που θα συγκρίνουμε στο Κεφάλαιο 3

Το **Drupal** είναι ένα εργαλείο το οποίο σου δίνει τη δυνατότητα να δημιουργήσεις πολλούς διαφορετικούς τύπους ιστοσελίδων (από απλά blogs μέχρι online κοινότητες), αλλά για μη ειδικευμένους χρήστες είναι δυσνόητη η ορολογία που χρησιμοποιεί στο διαχειριστικό περιβάλλον. Έχει ενσωματωμένο εργαλείο αναζήτησης και ως επιπλέον module παρέχει «φιλικές» αναζητήσεις προς τι μηχανές αναζήτησης URL.

Το **WordPress** είναι περισσότερο γνωστό σαν blogging platform, μια εφαρμογή γραμμένη σε php και σε γενικές γραμμές αρκετά απλοποιημένο στη χρήση του. Η διαμόρφωση της εμφάνισης γίνεται με την επιλογή ενός προτύπου και με CSS. Το WordPress δεν είναι ένα ολοκληρωμένο CMS και πάνω σε αυτό βασίζονται οι περισσότερες διαφορές που υπάρχουν με τα άλλα δύο CMS που εξετάζουμε.

Το **Joomla**! είναι ίσως το πιο κατάλληλο για όσους θέλουν να αναπτύξουν απλά και εύκολα έναν ιστότοπο (από προσωπική ιστοσελίδα με έναν χρήστη με δυναμικά στοιχεία μέχρι και δεκάδες χρήστες και διαχειριστές βάση των group policies που ορίζονται από τους διαχειριστές) και να είναι εξίσου απλή η διαχείρισή του και η χρήση του, χωρίς να σημαίνει ότι θα υστερεί σε ποιότητα και αξιοπιστία. Το Joomla! είναι ένα από τα πιο ισχυρά open source CMS λόγω της αρχιτεκτονικής του κώδικα αλλά και στο ότι «πίσω» του υπάρχει μια κοινότητα που το υποστηρίζει. Η δομή του είναι απλή και το περιβάλλον διαχείρισης είναι έτσι ώστε να δίνει στον χρήστη ξεκάθαρη την εικόνα για τις κινήσεις που πρέπει να κάνει. Είναι πολυγλωσσικό και έχει μεγάλη ποικιλία προτύπων αρκετά από αυτά διατίθενται δωρεάν στο διαδίκτυο.

Το αρνητικό του Joomla σε αντίθεση με το Drupal είναι ότι το δεύτερο σου δίνει τη δυνατότητα με μια εγκατάσταση να δημιουργήσεις και να διαχειριστείς πολλαπλούς ιστοτόπους. Επίσης το Joomla σε περιορίζει στη διανομή των ρόλων των χρηστών και στις άδειες πρόσβασης που δίνει. Πολύ σημαντικό επίσης είναι ότι τα URL του δεν είναι τόσο φιλικά στις μηχανές αναζητήσεις όπως στο Drupal, αλλά υπάρχει module που επί πληρωμή το βελτιώνει. Από την άλλη ο σχεδιασμός στο Drupal δεν είναι όσο ζωηρόχρωμος είναι του Joomla, αλλά προσαρμόζεται εύκολα.

Το WordPress αν και είναι πιο εύκολο σαν interface από τις άλλες δυο εξεταζόμενες πλατφόρμες, είναι πιο «αργό». Αν ο χρήστης βελτιστοποιήσει μόνος του τα ερωτήματα της βάσης, αποβάλει ορισμένα τμήματα κώδικα, συμπιέσει τα CSS αρχεία και εφαρμόσει προσωρινή αποθήκευση (caching), θα μπορέσει να διορθώσει την βραδύτητα του CMS.

Χρησιμοποιώντας την ιστοσελίδα : www.cmsmatrix.org μπορούμε να συγκρίνουμε τα Συστήματα Διαχείρισης Περιεχομένου που μας ενδιαφέρουν. Μπορεί και γίνεται σύγκριση των Συστημάτων Διαχείρισης Περιεχομένου ως προς την ασφάλεια (Security), την υποστήριξη (Support), την ευχρηστία (Ease of Use), την επίδοση (Performance), την διαχείριση (Management), τη δια-λειτουργικότητα (Interoperability), την ευελιξία (Flexibility), τις εφαρμογές (Built-in Applications) και το εμπόριο (Commerce).

Σύμφωνα λοιπόν με την προαναφερόμενη ιστοσελίδα και συγκρίνοντας 3 (τρία) από τα δημοφιλέστερα CMS ανοιχτού κώδικα που προαναφέραμε, έχουμε τα παρακάτω στοιχεία του πίνακα ως προς το θέμα της ασφάλειας. Με μια πρώτη ματιά λοιπόν, στα δεδομένα του πίνακα (ΠΙΝΑΚΑΣ 1), βλέπουμε πως καμία από τις τρεις εξεταζόμενες πλατφόρμες δεν μας καλύπτει στο θέμα της ασφάλειας κατά το εκατό τις εκατό. Όλες οι πλατφόρμες έχουν ελλείψεις, μερικές από τις οποίες όμως μας τις προσφέρουν με Free Add On plugins, και κάποιες άλλες δεν μας τις καλύπτουν καθόλου!

Η ασφάλεια είναι πάντα σημαντική αλλά και τα τρία συστήματα παρέχουν αρκετά ισχυρή ασφάλεια στη βάση του συστήματος. Δεν συμβαίνει το ίδιο και με τα διάφορα 3rd party plugins/ modules και widgets που μπορεί να χρησιμοποιήσει κάποιος συμπληρωματικά κατά το στήσιμο ενός ιστοτόπου με τα παραπάνω CMS.

DRUPAL http://secunia.com/advisories/search/?search=drupal&page=0

JOOMLA: http://secunia.com/advisories/search/?search=joomla&page=0

WORDPRESS: http://secunia.com/advisories/search/?search=wordpress

και στα τρία παρατηρούμε πως υπάρχει σημαντικό πρόβλημα στις ενημερώσεις αυτών των 3rd party plugins.

	Drupal 6.10	Joomla! 1.5.10	WordPress 2.2.1
Audit Trail	Yes	No	Limited
Captcha	Free Add On	Free Add On	No
Content Approval	Yes	Yes	Yes
Email Verification	Yes	Yes	Free Add On
Granular Privileges	Yes	No	Yes
Kerberos Authentication	No	No	No
LDAP Authentication	Free Add On	Yes	No
Login History	Yes	Yes	Free Add On
NIS Authentication	No	No	No
NTLM Authentication	Free Add On	No	No
Pluggable Authentication	Yes	Yes	Yes
Problem Notification	No	No	Free Add On
Sandbox	No	No	Limited
Session Management	Yes	Yes	Free Add On
SMB Authentication	No	No	No
SSL Compatible	Yes	Yes	Yes
SSL Logins	No	Yes	Free Add On
SSL Pages	No	Yes	Limited
Versioning	Yes	Free Add On	Free Add On

Πίνακας 1: Σύγκριση των Drupal, Joomla! και WordPress ως προς την ασφάλεια

Λόγω της αρκετά μεγάλης βάσης χρηστών και της ευχρηστίας, στην παρούσα πτυχιακή επιλέξαμε να ασχοληθούμε με το Joomla! και τα θέματα ασφαλείας του. Θα δούμε τα τρωτά σημεία στην ασφάλεια και τους τρόπους που χρησιμοποιούνται για να διασφαλιστεί η ασφάλεια του ιστότοπου που θα δημιουργήσουμε.

ΚΕΦΑΛΑΙΟ 4

Joomla!

4.1 Η ιστορία του Joomla!

Όσον αφορά το σχεδιασμό ιστοσελίδων, το Joomla! [6] υπήρξε μία από τις πρωτοπόρες εταιρείες του κλάδου. Η ιδέα για το Joomla! χρονολογείται από τον Αύγουστο του 2005, όταν το έργο αποσπάστηκε από ένα άλλο έργο web design. Η ιδέα ήταν για μια open source κοινότητα, όπου οι χρήστες θα μπορούσαν να έχουν την εξουσία. Ο σκοπός του σχεδιασμού του portal του Joomla! πέτυχε και έχει σίγουρα πολλούς χρήστες. Κατά τα τελευταία τέσσερα χρόνια, το Joomla! φαίνεται να έχει κάνει κάποιες αλλαγές, αλλά οι βασικές αρχές έχουν παραμείνει οι ίδιες, ως επί το πλείστον.

Το Joomla! διέκοψε από το Mambo εν μέσω ανησυχιών για την open source φύση του έργου και στη συνέχεια-η CEO Peter Lamont φρόντισε να πάρει την ανατροφοδότηση των χρηστών σχετικά με το πώς το έργο θα πρέπει να τρέξει για να πάει μπροστά. Οι χρήστες του Διαδικτύου στην ιστοσελίδα OpenSourceMatters πρόσφεραν την υποστήριξή τους για το νέο έργο και στις αρχές Σεπτεμβρίου του 2005, το Joomla! γεννήθηκε. Το όνομά του δόθηκε από τους ίδιους τους χρήστες μετά από ερώτηση που τέθηκε από την ίδια την εταιρία, το οποίο προέρχεται από την Αραβική και σημαίνει «όλοι μαζί», το οποίο είναι και μια σημαντική ιδέα για αυτό το web design έργο.

Η ημερομηνία που κυκλοφόρησε το πρώτο Joomla! Interface ήταν στις 7 Σεπτεμβρίου του 2005 και έγινε δεκτό με πολύ καλές κριτικές. Από τότε, είναι πρωτοπόρος στον τομέα του open source design, παρέχοντας στους σχεδιαστές ιστοσελίδων τη δυνατότητα να κάνουνε πράγματα όπως εκείνοι επιθυμούν. Πολλές εταιρείες προσπάθησαν να ακολουθήσουν το προσχέδιο του Joomla!, καθώς όλο και περισσότεροι χρήστες του Διαδικτύου αναγνωρίζουν το εξαιρετικό δυναμικό που έχει καταφέρει η εταιρεία. Πολυάριθμες είναι οι ενημερώσεις που έχουν κυκλοφορήσει για το Joomla! από το 2005, και σύμφωνα με δημοσιεύματα, το Joomla! είναι τώρα η πιο δημοφιλής ανοικτή πύλη του διαδικτύου του open source design.

4.2 Χαρακτηριστικά του Joomla!

Σύμφωνα με τις παρατηρήσεις των χρηστών του Joomla! μπορούμε να συμπεράνουμε ότι να πιο βασικά χαρακτηριστικά του είναι τα εξής:

[6] http://analogik.org/HistoryofJoomla.html

- Χρησιμοποιεί από τις καλύτερες διαθέσιμες τεχνολογίες:
    1. MySQL για τη Βάση Δεδομένων
    2. PHP για την προγραμματιστική λογική
    3. XML
    4. CSS2
    5. Δυνατότητα RSS.
- Ο πλήρης μηχανισμός διαχείρισης της βάσης δεδομένων του εκάστοτε site.
- Τμήματα για Νέα Προϊόντα ή Υπηρεσίες είναι πλήρως επεξεργάσιμα, διαχωρίσιμα και εύχρηστα.
- Παρέχεται οι δυνατότητα δημιουργίας χρηστών σε διάφορα επίπεδα πρόσβασης
- Τμήματα με θεματικές ενότητες μπορούν να προστεθούν από διαφορετικούς συντάκτες (ανάλογα πάντα με τα δικαιώματα που τους έχουν δοθεί από τον δημιουργό).
- Το περιεχόμενο και το περιβάλλον είναι πλήρως παραμετροποιημένα, όπως επίσης και οι θέσεις του αριστερού, κεντρικού και δεξιού μενού.
- Είναι εύχρηστο για όλους τους χρήστες (ακόμη για αρχάριους χρήστες Η/Υ)
- Υποστηρίζει πολλές γλώσσες.
- Υποστηρίζει ανέβασμα φωτογραφιών μέσω του browser του χρήστη, σε δική του βιβλιοθήκη για χρήση οπουδήποτε στο site.
- Παρέχει δυναμική υποστήριξη Forum/Ψηφοφορίας για τα επί τόπου αποτελέσματα.
- Υπάρχει ειδικός μηχανισμός για της μηχανές αναζήτησης.
- Και τέλος και πολύ βασικό είναι ότι το Joomla "τρέχει" σε όλα τα λειτουργικά (Linux, FreeBSD, MacOSX server, Solaris και AIX).

4.3 Η αρχιτεκτονική του Joomla!

Το Joomla! θα μπορούσαμε να πούμε πως έχει την αρχιτεκτονική μιας εφημερίδας. Αποτελείται από πολλά διαφορετικά μέρη, και η δομή αυτή μας επιτρέπει να κάνουμε επεκτάσεις εύκολα.

Ένα CMS θα μπορούσαμε να το χωρίσουμε σε τρία βασικά υποσυστήματα:

- **υποσύστημα συλλογής** (Collection System), το οποίο είναι υπεύθυνο για τη συγγραφή (Authoring), την απόκτηση (Acquisition), την μετατροπή (Conversion), την συσσώρευση (Aggregation) και τέλος για τις υπηρεσίες συλλογής (Collection Services). Είναι υπεύθυνο δηλαδή για όλες τις ενέργειες που πρέπει να γίνουν προτού η πληροφορία ετοιμαστεί για να δημοσιευθεί.
- **υποσύστημα διαχείρισης** (Management System), το οποίο είναι υπεύθυνο για την αποθήκευση των συστατικών του περιεχομένου όπως επίσης και για κάθε είδος αρχείου που χρησιμοποιείται. Επίσης συμπεριλαμβάνει την «αποθήκη» περιεχομένου, το workflow καθώς και τις δυνατότητες διαχείρισης. Έτσι, πρέπει να περιέχει αποθηκευτικό χώρο, ένα σύστημα διαχείρισης για τις ρυθμίσεις του CMS, καθορισμένα σύνολα βημάτων για την πραγματοποίηση της εργασίας (για να μπορέσει να ετοιμαστεί το περιεχόμενο για δημοσίευση), και τέλος ένα σύνολο συνδέσεων υλικού και λογισμικού (μεταξύ δικτύων, εξυπηρετητών και αποθηκών δεδομένων)

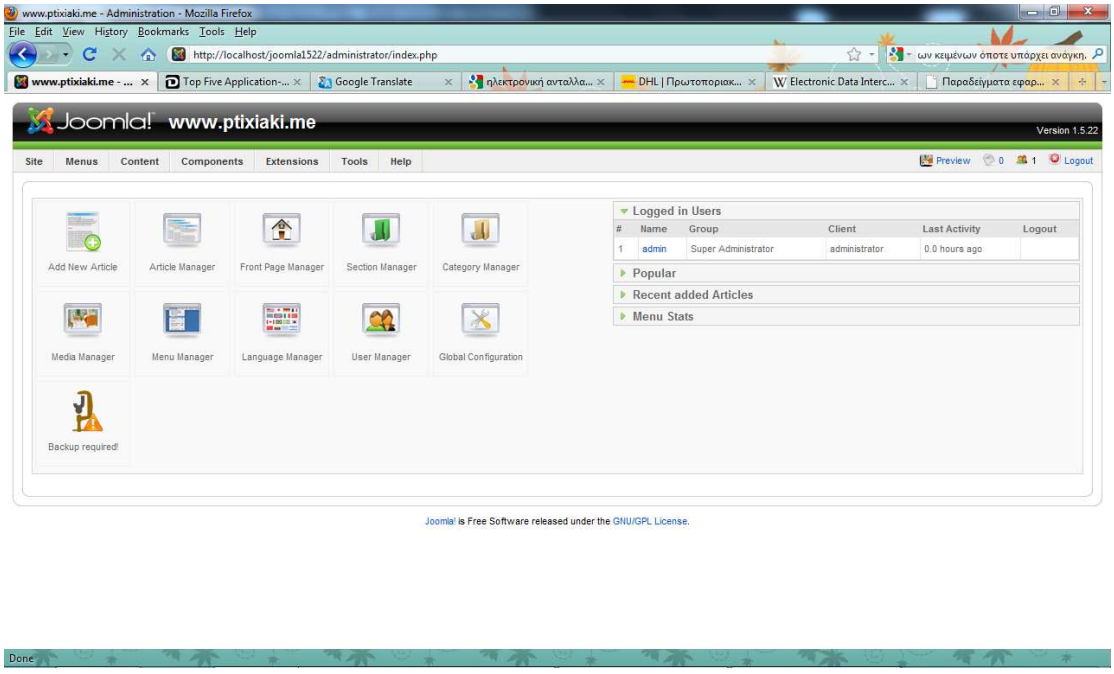
- **υποσύστημα δημοσίευσης** (Publishing System), το οποίο είναι υπεύθυνο για την εξαγωγή περιεχομένου από τις «αποθήκες» δεδομένων και την αυτόματη δημιουργία δημοσιεύσεων. Περιλαμβάνει φόρμες και υπηρεσίες δημοσιεύσεων, συνδέσεις (μεθόδους και εργαλεία που χρησιμοποιούνται για την εισαγωγή δεδομένων από εξωτερικά συστήματα από το CMS), δημοσιεύσεις ιστού και άλλες δημοσιεύσεις (π.χ. εκτύπωσης ή ηλεκτρονικές δημοσιεύσεις)

4.4 Η δομή του Joomla! (Front End-Back End)

Το Joomla! αποτελείται από δύο τμήματα: το Front End και το Back End. Το Front End είναι το δημόσιο τμήμα, το τμήμα δηλαδή στο οποίο εμφανίζεται το περιεχόμενο της ιστοσελίδας που βλέπουν οι τελικοί χρήστες χρησιμοποιώντας τον browser της επιλογής τους. Στο Front End τμήμα βρίσκονται τα άρθρα, τα μενού και όλα τα στοιχεία τα οποία επιλέγουμε να εμφανίζονται στο site μας.

Το Back End τμήμα του Joomla! είναι η περιοχή διαχείρισης του administrator, περιέχει δηλαδή το administration layer του ιστοτόπου για τους διαχειριστές. Σ' αυτό το τμήμα γίνεται η διαμόρφωση, η συντήρηση, η δημιουργία νέου περιεχομένου, ο «καθαρισμός» του site και η παραγωγή στατιστικών από τους διαχειριστές (administrators).

Το Back End βρίσκεται σε διαφορετική διεύθυνση από το Front End, για παράδειγμα σε τοπική εγκατάσταση βρίσκεται στην: http://localhost/joomlasite/administrator.



Εικόνα 4: Σελίδα διαχείρισης - Back End

ΚΕΦΑΛΑΙΟ 5

ΕΓΚΑΤΑΣΤΑΣΗ ΤΟΥ Joomla! ΚΑΙ ΤΩΝ ΑΠΑΡΑΙΤΗΤΩΝ ΕΡΓΑΛΕΙΩΝ ΓΙΑ ΤΗΝ ΛΕΙΤΟΥΡΓΙΑ ΤΟΥ

Όπως προαναφέραμε σε προηγούμενο κεφάλαιο θα γίνει ανάπτυξη πρωτότυπου ιστοτόπου με χρήση μίας από τις υποεξέτασης πλατφόρμες (Joomla!) και θα γίνει η αξιολόγηση της ασφαλείας του τελικού προϊόντος καθώς και οι τρόποι θωράκισής του από τις επιθέσεις που θα εντοπιστούν.

5.1 Εργαλεία που απαιτούνται

Για να δημιουργήσουμε μια εφαρμογή ηλεκτρονικού καταστήματος χρησιμοποιώντας το Joomla! προαπαιτούνται κάποιες άλλες λειτουργίες.
Αρχικά λοιπόν, θα πρέπει να έχει εγκατασταθεί στον ηλεκτρονικό υπολογιστή που χρησιμοποιούμε η PHP γλώσσα προγραμματισμού για τη διαμόρφωση του site και αφού η εγκατάσταση του e-καταστήματος θα γίνει σε τοπικό server θα χρειαστεί να χρησιμοποιήσουμε τον Apache HTTP.

Ο Apache είναι ένας εξυπηρετητής του παγκόσμιου ιστού και είναι ένας από τους πιο δημοφιλής γιατί λειτουργεί σε διάφορες πλατφόρμες όπως Windows, Linux, Unix και Mac. Χρησιμοποιείται για να εξυπηρετεί στατικό και δυναμικό περιεχόμενο στο διαδίκτυο, παράγεται και διανέμεται δωρεάν από το Apache Software Foundation.

Θα πρέπει να εγκαταστήσουμε τη MySQL για τη δημιουργία της βάσης δεδομένων στην οποία θα αποθηκεύονται όλες οι πληροφορίες που αφορούν το ηλεκτρονικό μας κατάστημα. Η MySQL (My Structured Query Language) αποτελεί ένα σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων που χρησιμοποιεί την SQL γλώσσα και λειτουργεί ως διακομιστής παροχής πρόσβασης πολλών χρηστών σε μια σειρά από βάσεις δεδομένων με ασφάλεια, καθώς μόνο οι εγγεγραμμένοι ως χρήστες έχουν δικαίωμα πρόσβασης στα δεδομένα της εκάστοτε βάσης.

Για τη διαχείριση του administration της MySQL μέσω του παγκόσμιου ιστού, θα χρειαστεί ένα ακόμη εργαλείο να χρησιμοποιήσουμε, το phpMyAdmin, το οποίο είναι δωρεάν λογισμικό γραμμένο σε PHP και υποστηρίζει ένα ευρύ φάσμα δράσεων της MySQL.

Ένα χρήσιμο λοιπόν εργαλείο για όλα τα παραπάνω είναι το XAMPP. Το XAMPP είναι ένα ελεύθερο και ανοιχτό cross platform web server package, το οποίο αποτελείται από τον Apache HTTP Server, τη MySQL, το phpMyAdmin και των διερμηνέων που απαιτούνται για scripts που είναι γραμμένα σε PHP και Perl γλώσσες προγραμματισμού. Αυτό το εργαλείο θα μας βοηθήσει να μετατρέψουμε τον υπολογιστή μας σε web server.

Σημείωση: Εκτός από το XAMPP, υπάρχει και το WAMP που κάνει ακριβώς την ίδια δουλεία. Η κύρια διαφορά του είναι ότι το XAMPP περιλαμβάνει και διερμηνέα για γλώσσα προγραμματισμού Perl και FileZilla server που το WAMP δεν τα έχει. Επίσης το XAMPP είναι πιο απλό στη χρήση του από αρχάριους χρήστες.

5.2 Εγκατάσταση XAMPP

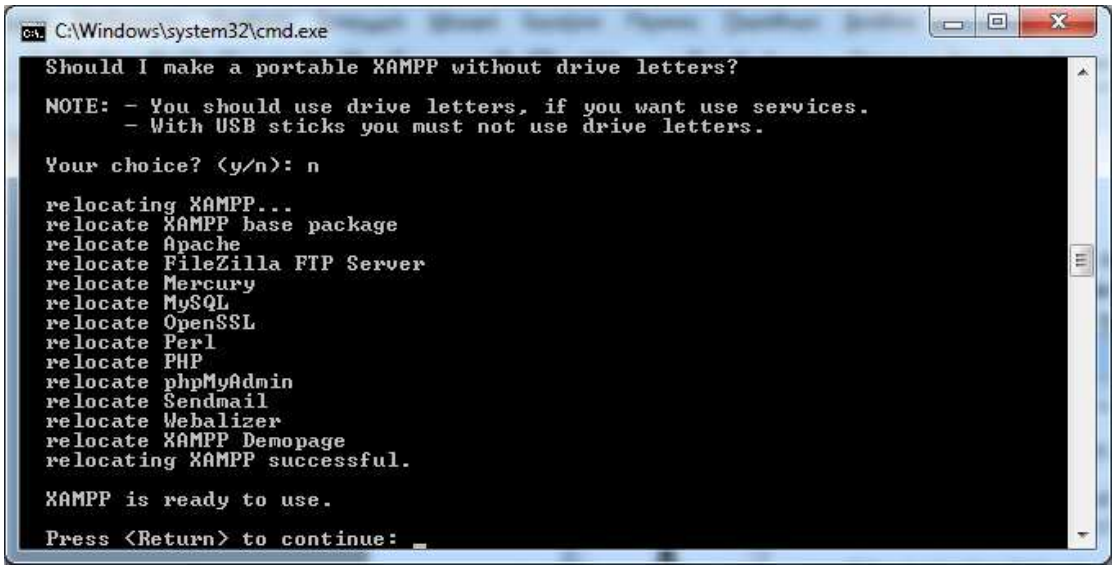
Ο χρήστης μπορεί να επισκεπτεί την ιστοσελίδα: http://www.apachefriends.org/en/xampp.html και και να κατεβάσει την έκδοση XAMPP 1.7.3 για Windows η οποία περιλαμβάνει τα εξής:

- Apache 2.2.14 (IPv6 enabled) + OpenSSL 0.9.8l
- MySQL 5.1.41 + PBXT engine
- PHP 5.3.1
- phpMyAdmin 3.2.4
- Perl 5.10.1
- FileZilla FTP Server 0.9.33
- Mercury Mail Transport System 4.72

Παρακάτω βλέπουμε το πρώτο παράθυρο κατά την εγκατάσταση του XAMPP στο οποίο επιλέγουμε τη θέση που θέλουμε να αποθηκευτούν τα απαραίτητα αρχεία για τη λειτουργία του και πατώντας το Install ξεκινάει η εγκατάστασή του.

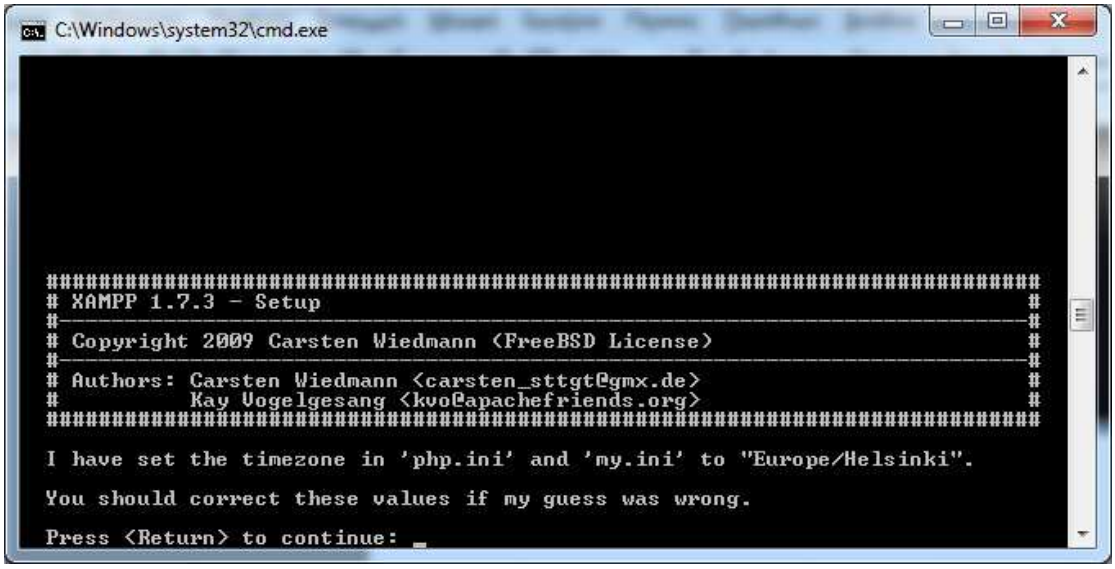
Εικόνα 5: Επιλογή φακέλου εγκατάστασης XAMPP

Μετά από μερικά βήματα εγκατάστασης και μερικές απλές επιλογές μας εμφανίζει το τελευταίο παράθυρο εγκατάστασης στο οποίο μας ενημερώνει τι εγκαταστάθηκε. Το XAMPP είναι έτοιμο.



Εικόνα 6: Εγκατάσταση XAMPP. Απόρριψη χρησιμοποίησης drive letters

Πληκτρολογώντας λοιπόν και RETURN για να συνεχίσουμε, μας εμφανίζει σε ένα νέο παράθυρο ότι έχει ορίσει την ζώνη ώρας ως “Europe/Helsinki”.



Εικόνα 7: Εγκατάσταση XAMPP. Ορισμός ζώνης ώρας

Μόλις ολοκληρωθεί η εγκατάστασή του θα πρέπει να το αλλάξουμε για να συμβαδίζει με τα δικά μας δεδομένα. Θα επισκεφτούμε λοιπόν το site: http://us2.php.net/manual/en/timezones.europe.php για να δούμε πως ορίζει τη ζώνη ώρας για την Ελλάδα. Η ζώνη ώρας αναφέρεται ως Mode/Athens. Πηγαίνουμε στο **C:\XAMPP\xampp\php\php.ini** να αλλάξουμε το αρχείο “php.ini”.

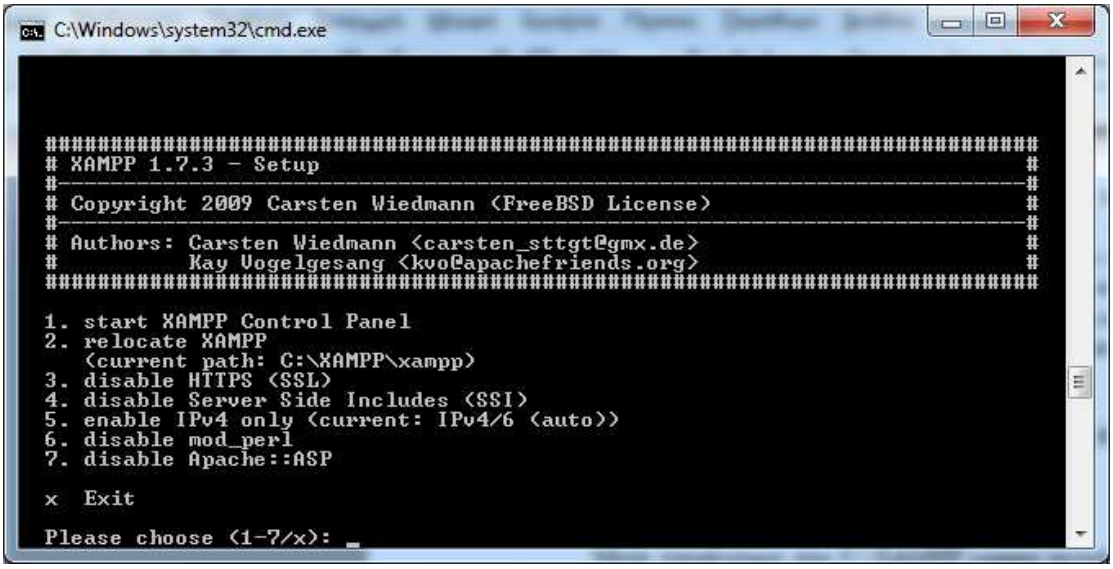
Ανοίγουμε το αρχείο και ψάχνουμε για το σημείο που θα βρούμε τον κώδικα:

```
[Date]
; Defines the default timezone used by the date functions
; http://php.net/date.timezone
date.timezone = “Europe/Athens”
```

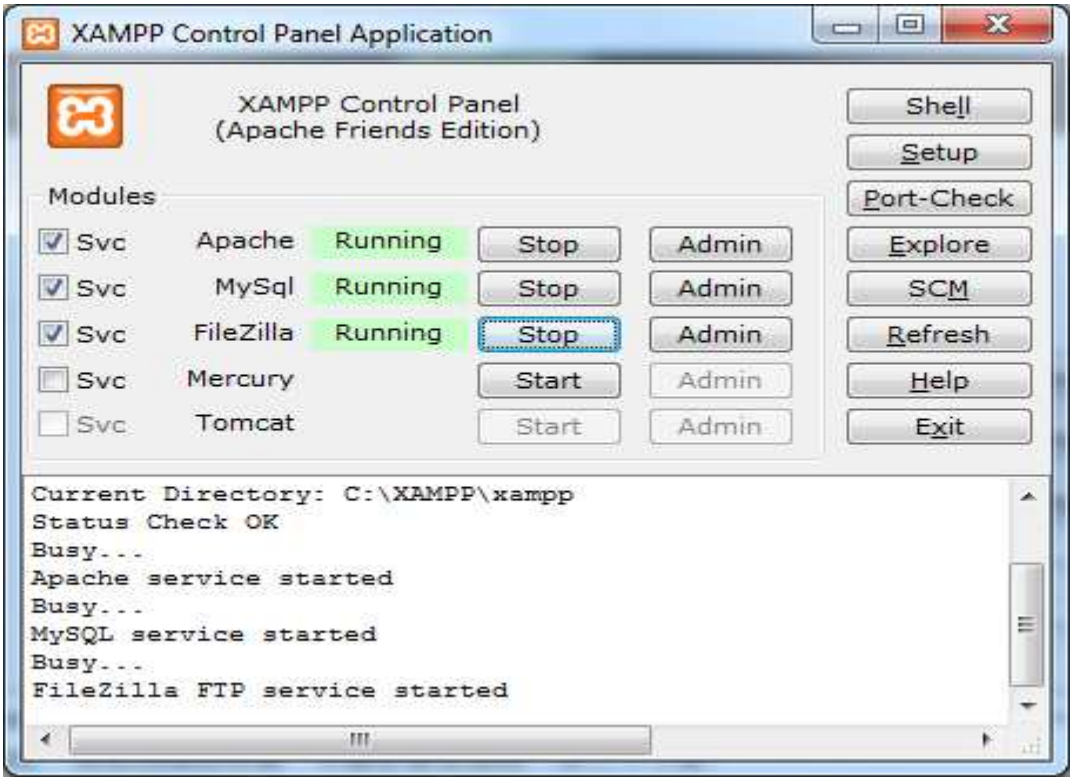
Μετά πηγαίνουμε στο C:\XAMPP\xampp\mysql\bin\my.ini για να αλλάξουμε το my.ini αρχείο. Το ανοίγουμε και αλλάζουμε τον κώδικα:

```
Default-time-zone = “Europe/Athens”
```

Αφού σώσαμε τις αλλαγές και στα δύο αρχεία, ξανανοίγουμε το XAMPP και επιλέγουμε να μπούμε στο Control Panel για να εκκινήσουμε τον Apache, την MySQL και τον FileZilla.

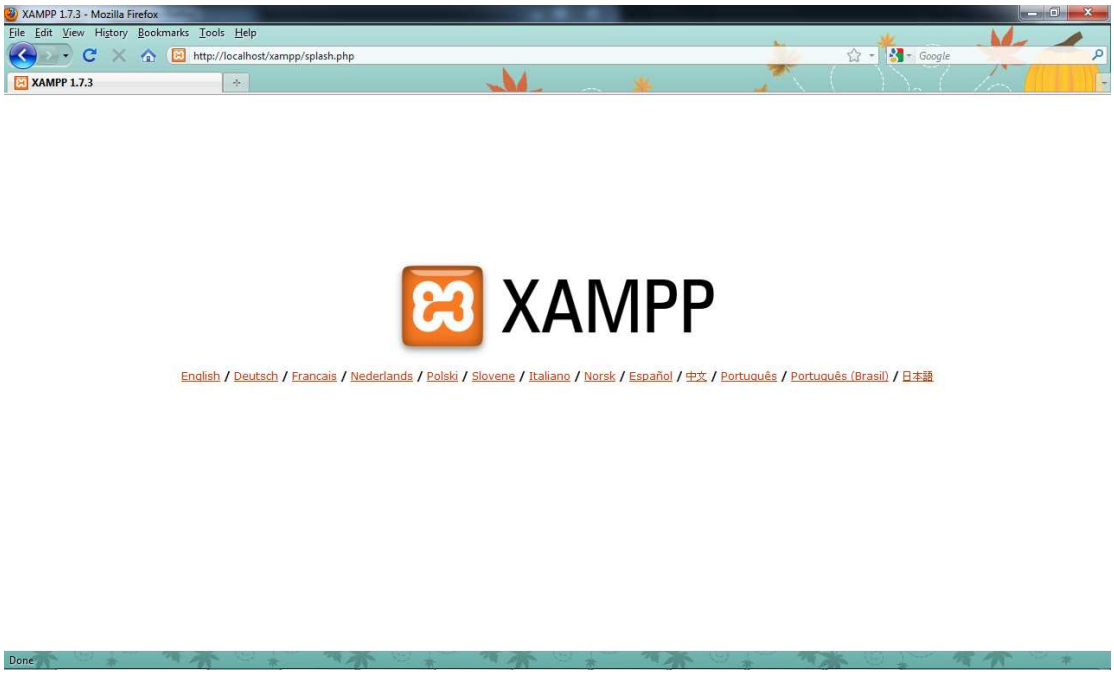


Εικόνα 8: Τέλος εγκατάστασης XAMPP

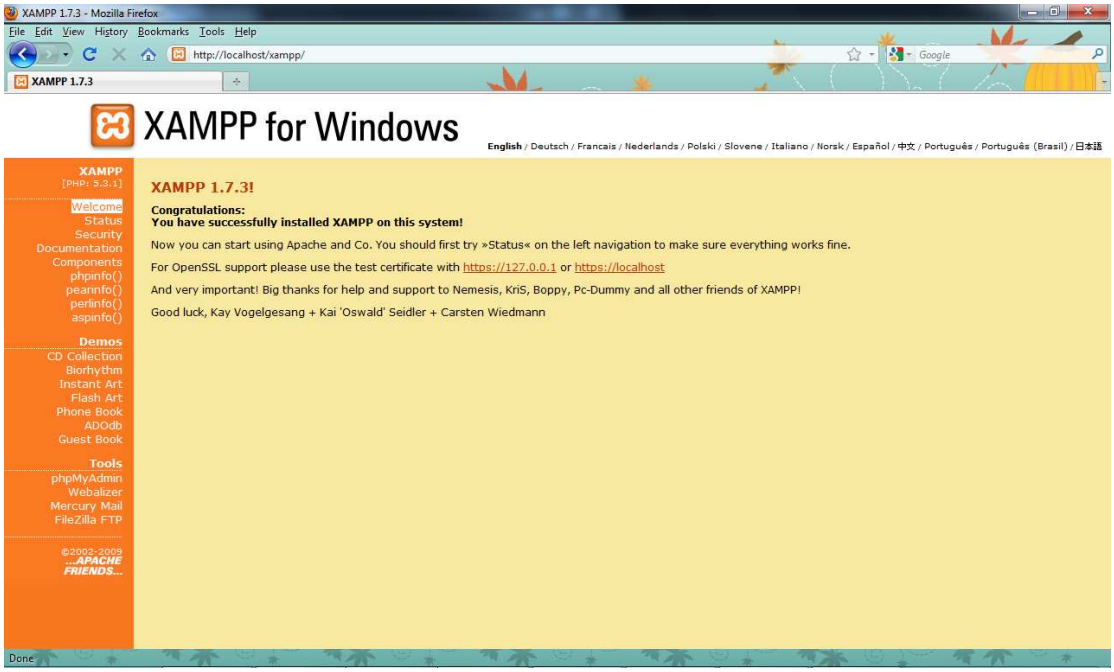


Εικόνα 9: XAMPP Control Panel

Επόμενο βήμα είναι να ανοίξουμε έναν browser και να πληκτρολογήσουμε http://localhost/ ή http://127.0.0.1 για να μας εμφανιστεί η πρώτη σελίδα του XAMPP ώστε να κάνουμε τις σωστές ρυθμίσεις. Στο παράθυρο που μας ανοίγει επιλέγουμε για γλώσσα τα Αγγλικά για να συνεχίσουμε.

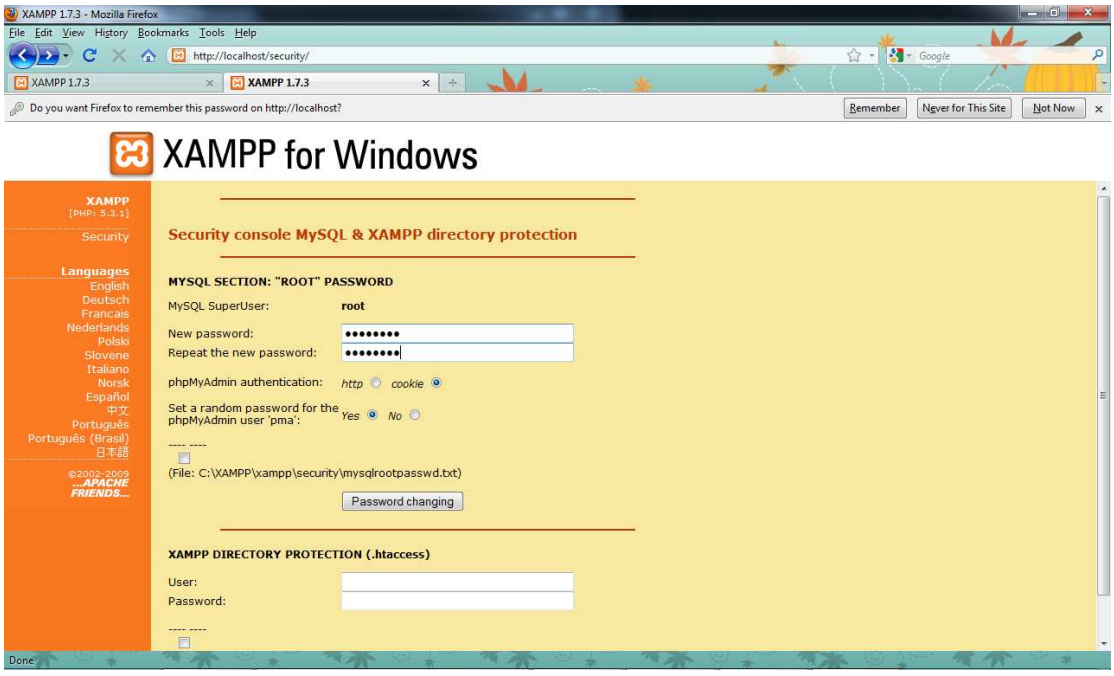


Εικόνα 10: Ρυθμίσεις XAMPP. Επιλογή γλώσσας.



Εικόνα 11: Μήνυμα καλωσορίσματος XAMPP

Στην παραπάνω εικόνα βλέπουμε το μήνυμα καλωσορίσματος και την επιτυχή εγκατάσταση του XAMPP και από το αριστερό μενού επιλέγουμε το SECURITY για να θέσουμε τους κωδικούς για την MySQL, το phpmyadmin και για την προστασία του XAMPP directory.



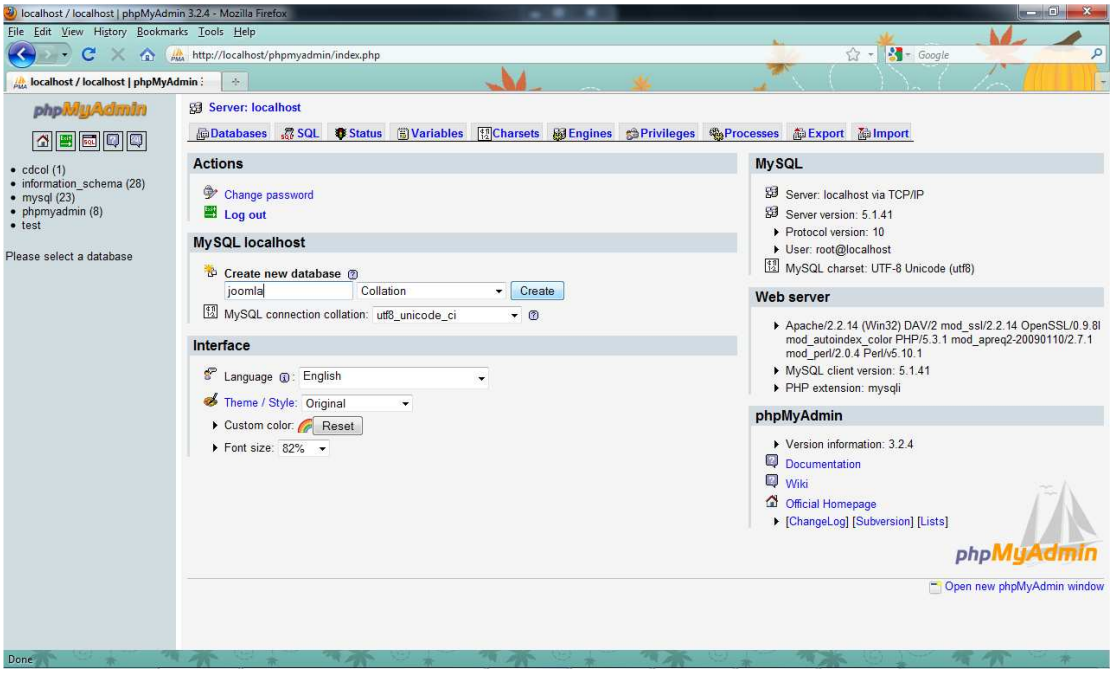
Εικόνα 12: Ορισμός κωδικών της MySQL και του XAMPP directory

Αφού δώσουμε και USER και PASSWORD για την προστασία του XAMPP παίρνουμε το παρακάτω μήνυμα ασφαλείας:

SUCCESS: The XAMPP directory is protected now! All personal data was saved in the following file:
C:\XAMPP\xampp\security\xampp.users
C:\XAMPP\xampp\htdocs\xampp\.htaccess

Εικόνα 13: Μήνυμα επιτυχής αποθήκευσης κωδικών του XAMPP directory

Το επόμενο βήμα είναι να δημιουργήσουμε τη βάση δεδομένων. Από το μενού της αριστερής στήλης του κέντρου διαχείρισης του XAMPP, επιλέγουμε τα Tools το phpMyAdmin. Αφού κάνουμε Login, βλέπουμε ότι στο πλαίσιο MySQL localhost, υπάρχει το "Create new database". Πληκτρολογούμε το επιθυμητό όνομα (εδώ "joomla") και στο πεδίο "Collation" επιλέγουμε utf8_unicode_ci και κατόπιν κάνουμε κλικ στο create. Έτοιμη η βάση μας!

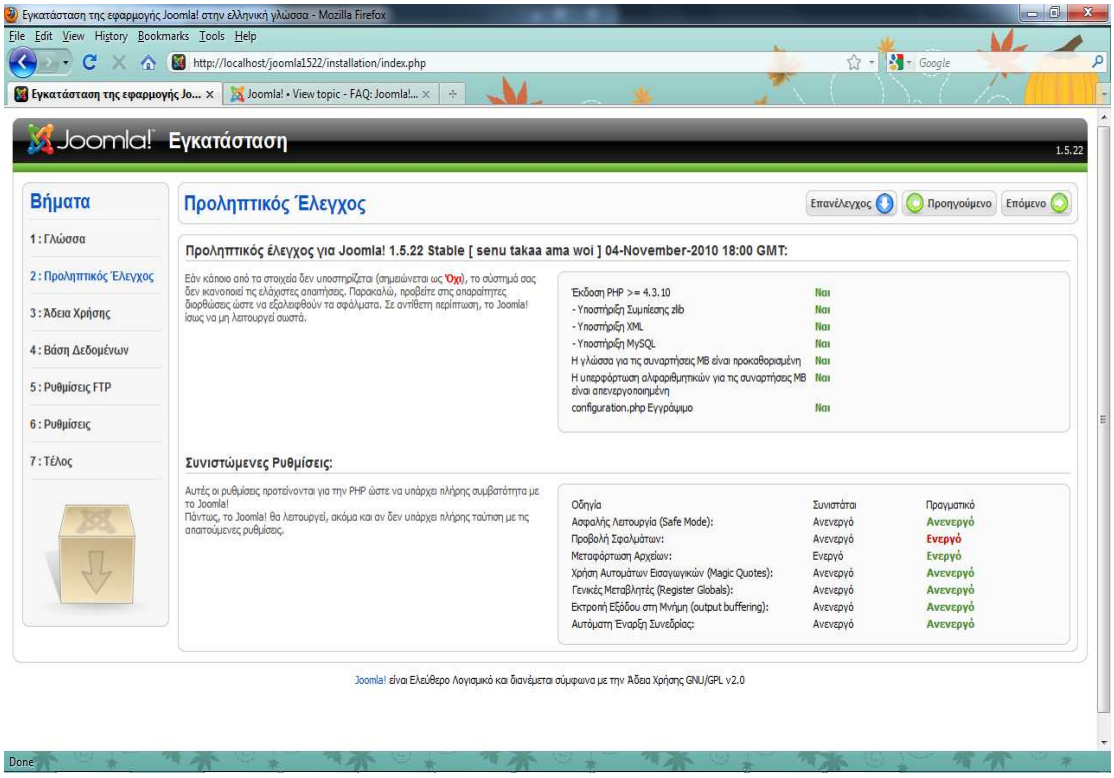


Εικόνα 14: Δημιουργία βάσης δεδομένων μέσω του phpMyAdmin

5.3 Εγκατάσταση του Joomla!

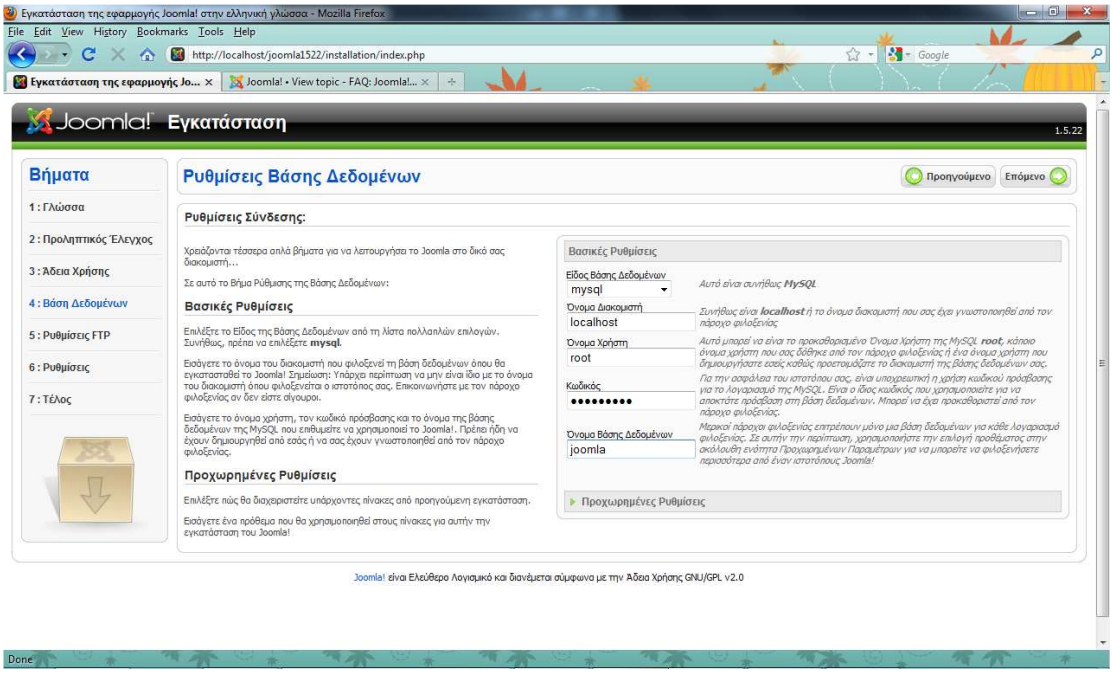
Από τη σελίδα http://www.joomla.gr/ κατεβάσαμε την έκδοση του joomla! Joomla_1.5.22-Stable-Full_Package. Το επόμενο βήμα είναι να πάμε στον φάκελο htdocs που βρίσκεται στο C\XAMPP\xampp\htdocs και να δημιουργήσουμε έναν φάκελο στον οποίο και θα αποθηκεύσουμε τα αρχεία του zip file που κατεβάσαμε.

Στη συνέχεια ανοίγουμε έναν browser και πληκτρολογούμε http://localhost/joomla1522 . Στο παράθυρο που μας εμφανίζεται επιλέγουμε τη γλώσσα που θέλουμε να χρησιμοποιήσουμε για την εγκατάσταση του joomla! και μεταβαίνουμε στην επόμενη σελίδα που γίνεται ένας προληπτικός έλεγχος για όλα όσα χρειάζεται το Joomla για να λειτουργεί σωστά (για παράδειγμα η PHP).



Εικόνα 15: Εγκατάσταση Joomla! Προληπτικός έλεγχος

Αφού διαβάσουμε και την άδεια χρήσης, θα πρέπει να ρυθμίσουμε τη βάση δεδομένων όπως εμφανίζεται και στην παρακάτω εικόνα.

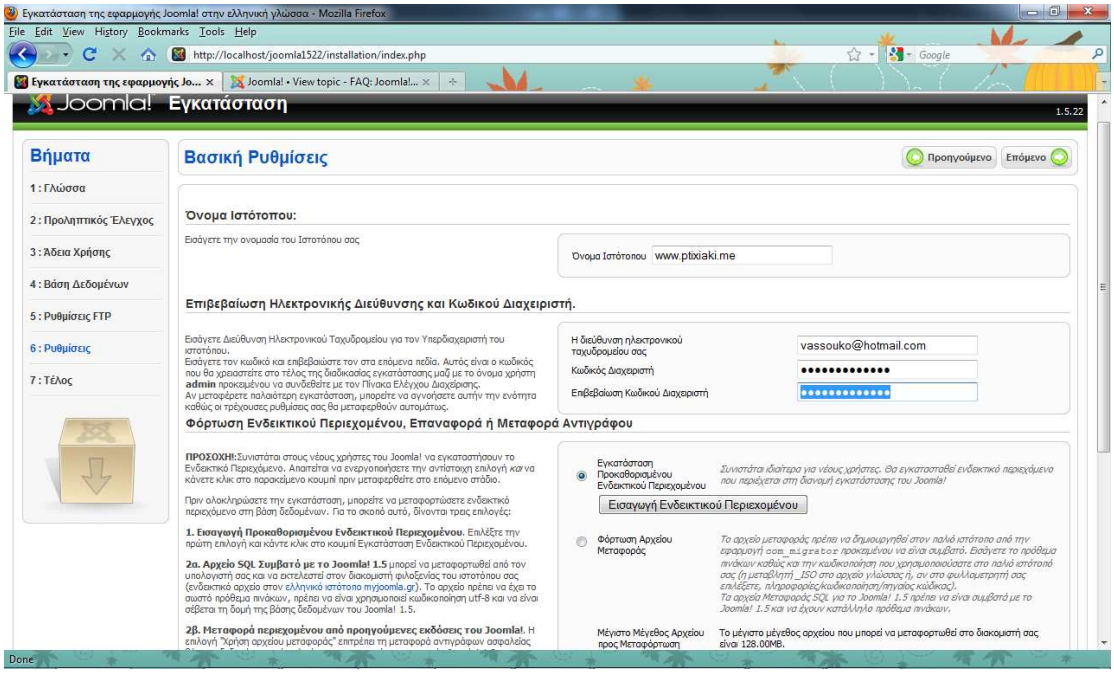


Εικόνα 16: Ρυθμίσεις Βάσης Δεδομένων

- ✓ Είδος βάσης δεδομένων: MySQL
- ✓ Όνομα διακομιστή: localhost
- ✓ Όνομα χρήστη: root

- ✓ Κωδικός: Πληκτρολογούμε τον κωδικό που είχαμε ορίσει πιο πριν για την MySQL
- ✓ Όνομα βάσης δεδομένων: Joomla

Κατόπιν κάνουμε κλικ στο επόμενο και μας ανοίγει ένα παράθυρο στο οποίο δίνουμε την ονομασία του site μας (www.ptixiaki.me), το mail του διαχειριστή και τον κωδικό που επιθυμεί. Επίσης εάν θέλουμε να μας φτιάξει μια ενδεικτική αρχική σελίδα από μόνο του το joomla κάνουμε κλικ στο «Εισαγωγή Ενδεικτικού Περιεχομένου».



Εικόνα 17: Βασικές ρυθμίσεις Joomla!

Εφόσον η εγκατάστασή μας έχει ολοκληρωθεί, στην τελευταία οθόνη εγκατάστασης μας εμφανίζεται ένα μήνυμα που μας ενημερώνει ότι πρέπει να διαγράψουμε τον φάκελο instalation, ο οποίος βρίσκεται στον φάκελο εγκατάστασης του Joomla!, με το όνομα «joomla1522». Μόλις διαγράψουμε τον φάκελο instalation, θα μπορέσουμε να δούμε το site μας.

Για την κατασκευή του ιστοτόπου που θα χρησιμοποιήσουμε για έλεγχο ασφάλειας θα χρησιμοποιήσουμε το VirtueMart από την παρακάτω ιστοσελίδα: http://virtuemart.net/ .

ΚΕΦΑΛΑΙΟ 6

COMPONENT VIRTUEMART

6.1 Περιγραφή του VirtueMart

Το VirtueMart [7] είναι μια open source e-commerce λύση, σχεδιασμένο σαν extension για το Mambo ή το Joomla! CMS. Είναι γραμμένο σε PHP, απαιτεί τη χρήση της MySQL και είναι μια αξιόπιστη λύση για ηλεκτρονικά καταστήματα κατασκευασμένα με Joomla. Λόγω του ισχυρού εργαλείου διαχείρισης που έχει, δίνει στον διαχειριστή την δυνατότητα να χειρίζεται απεριόριστο αριθμό προϊόντων και διαφορετικών τιμών ανά προϊόν, παραγγελιών, κατηγοριών, εκπτώσεων, πελατών και ομάδων shopper και έχει ποικιλία στον τρόπο πληρωμών. Χρησιμοποιείται για μεσαίων κατηγοριών ηλεκτρονικών καταστημάτων.

6.1.1 Γενικά χαρακτηριστικά

Τα γενικά χαρακτηριστικά[8] του VirtueMart είναι τα παρακάτω:

- Προδιαγραφές για με SSL - Secure Sockets Layer (https) Encryption (128-bit),
- Εύκολα προσαρμόσιμοι τρόποι φορολόγησης: 1) Φορολόγηση αποστολής, 2) Φορολόγηση καταστήματος, και 3) Ευρωπαϊκό πρότυπο φορολόγησης (φορολογία καταστήματος για οποιοδήποτε πελάτη που προέρχεται από την Ευρωπαϊκή ένωση),
- Οι αγοραστές μπορούν να διαχειριστούν τους λογαριασμούς τους (απαιτείται εγγραφή),
- Διαχείριση διευθύνσεων αποστολής (οι πελάτες μπορούν να εισαγάγουν τις διευθύνσεις αποστολής τους),
- Ιστορικό παραγγελιών: Ο αγοραστής μπορεί να δει όλες τις προηγούμενες παραγγελίες του λεπτομερώς,
- Email επιβεβαίωσης παραγγελίας (παραμετροποιήσιμο) στέλνεται στον αγοραστή και στον ιδιοκτήτη του καταστήματος,

[7] http://en.wikipedia.org/wiki/VirtueMart
[8] http://virtuemart.net/component/content/69?task=view

- Πολλαπλά νομίσματα (ο πελάτης επιλέγει να κάνει την περιήγησή του και τις αγορές του με όποιο νόμισμα επιθυμεί),
- Πολλαπλές γλώσσες.

6.1.2 Κατάλογος προϊόντων

- Ισχυρή και άνετη διεπαφή για τον διαχειριστή καταστήματος,
- Προδιαγραφές για απεριόριστο αριθμό προϊόντων και κατηγοριών,
- Μπορεί να χρησιμοποιηθεί ως κατάστημα ή απλά σας κατάλογος παρουσίασης προϊόντων χωρίς τιμές,
- Γρήγορη μηχανή αναζήτησης προϊόντων, κατηγοριών και κατασκευαστών ή προμηθευτών,
- Εκτιμήσεις προϊόντων και reviews από τους αγοραστές,
- Προσθήκη συγκεκριμένων ιδιοτήτων σε επιλεγμένα προϊόντα όπως “special”,
- Διαθεσιμότητα προϊόντων, παρουσιάστε τον χρόνο παράδοσης του προϊόντος,
- Περιοχή downloads για ψηφιακά προϊόντα και προγράμματα,
- Ενημέρωση εγγεγραμμένων πελατών για προϊόντα που θα είναι ξανά διαθέσιμα.

6.1.3 Χαρακτηριστικά διαχείρισης

- Πολλαπλές εικόνες και αρχεία ανά προϊόν, σαν προσπέκτους, για την καλύτερη περιγραφή του προϊόντος,
- Ιδιότητες προϊόντων (όπως το μέγεθος ή το χρώμα) μπορούν να προστεθούν στο προϊόν,
- Τύποι προϊόντων για ταξινόμηση (όπως «αυτοκίνητο», «μοτοσικλέτα» ή το «μουσική»),
- Ομάδες αγοραστών για τους πελάτες (επιτρέπει τα διαφορετικές επίπεδα τιμών και τις επιλογές πληρωμής – π.χ. Χονδρική, λιανική),
- Διαφορετικές τιμές ανά προϊόν (ανάλογα με τον τύπο πελάτη ή τη διαθεσιμότητα),
- Ποικιλότροπη παρουσίαση τιμών (μορφοποίηση αριθμού & νομίσματος με ή χωρίς το ΦΠΑ),
- Κέντρο στατιστικών/ελέγχου καταστήματος με ανάλυση νέων πελατών, νέων παραγγελιών και πολλά άλλα,
- Διαχείριση ποσότητας αποθεμάτων για τα προϊόντα,
- Διαχείριση παραγγελίας με ιστορικό παραγγελίας, ενημέρωση πελάτη και δυνατότητα επεξεργασίας παραγγελίας,
- Κύρια έκθεση: πουλημένα προϊόντα, μηνιαίος/ετήσιος τζίρος,
- Διαχείριση κατάστασης παραγγελίας,
- Διαχείριση διαφορετικών νομισμάτων, χωρών και κρατών.

6.1.4 Τρόποι πληρωμής

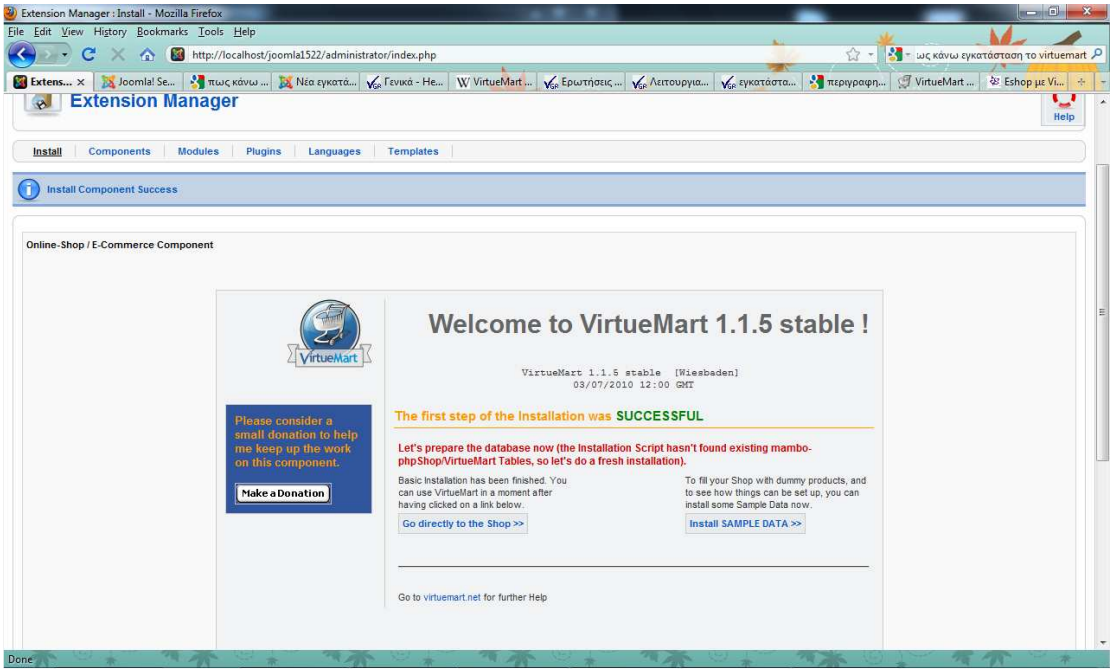
- Προδιαγραφές για ONLINE συναλλαγή με πιστωτική κάρτα,

- Προκαθορισμένες υπηρεσίες πληρωμής όπως το authorize.net ®, PayPal, 2Checkout, eWay, Worldpay, PayMate και NoChex,
- Αντικαταβολή, κατάθεση στην τράπεζα και παραλαβή από το κατάστημα μας.

6.2 Εγκατάσταση του VirtueMart

Η εγκατάσταση του VirtueMart γίνεται όπως σε όλα τα Joomla components. Από τη επίσημη σελίδα http://virtuemart.net/downloads , θα επιλέξουμε την έκδοση Complete Package for Joomla! 1.5

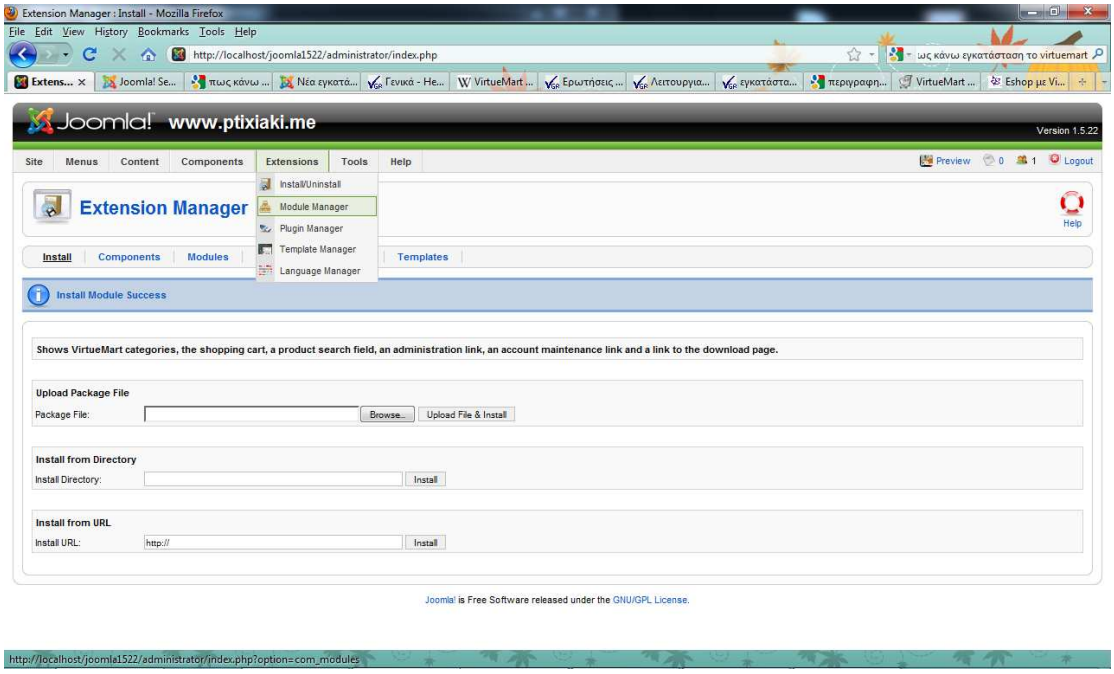
Μετά την αποθήκευσή του, κάνουμε login σαν administrator στο Joomla και από την επιλογή “Extensions” επιλέγουμε το “Install/Uninstall”, στην περιοχή Upload Package File πατάμε το “browse” για να βρούμε το επιθυμητό αρχείο (com_virtuemart_1.1.5.j15) και πατάμε το “Upload File & Install”. Το VirtueMart εγκαταστάθηκε και είναι έτοιμο.



Εικόνα 18: Σελίδα Διαχείρισης Joomla! Επιτυχής εγκατάσταση του VirtueMart

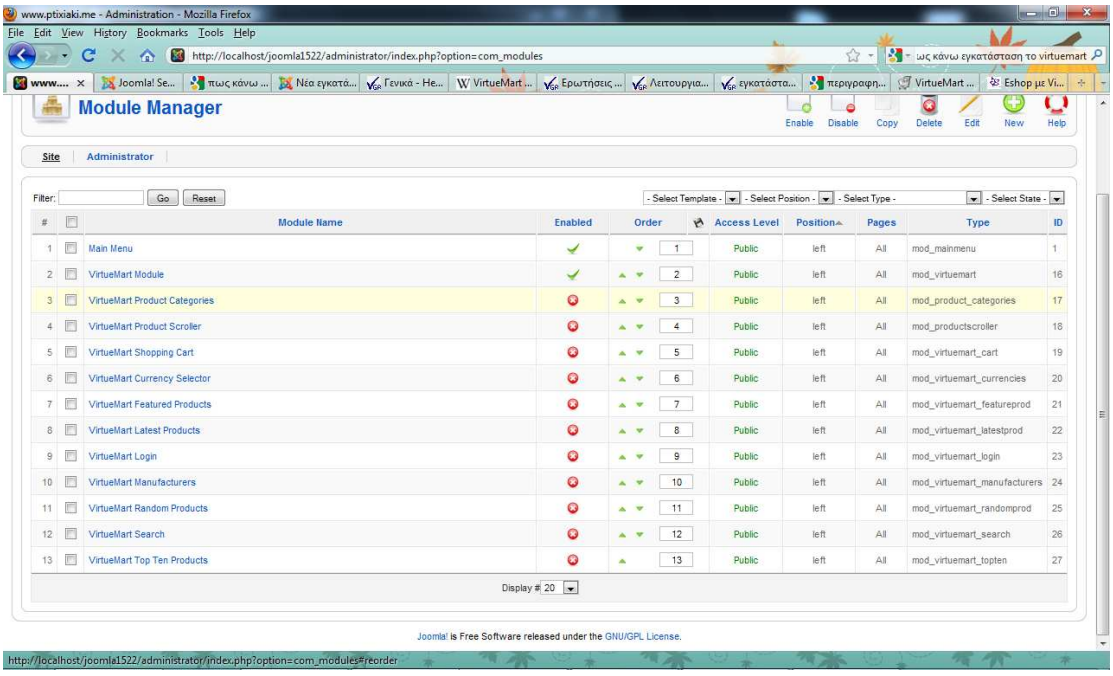
Στο παράθυρο που μας εμφανίζει μπορούμε να επιλέξουμε εάν επιθυμούμε να εγκαταστήσουμε δείγμα δεδομένων για να δούμε πως λειτουργεί το component ή να πάμε κατευθείαν στο e-κατάστημα. Εμείς επιλέγουμε να μην γίνει καμία εγκατάσταση δεδομένων.

Το επόμενο βήμα είναι να εγκαταστήσουμε το κυρίως module (mod_virtuemart_1.1.5.j15) του VirtueMart, με τα ίδια ακριβώς βήματα που εγκαταστήσαμε και προηγουμένως το component. Εφόσον η εγκατάσταση του είναι επιτυχής, πηγαίνουμε στην επιλογή “Extensions”, επιλέγουμε το “Module Manager” και ενεργοποιούμε το module που μόλις εγκαταστήσαμε, ώστε να έχουμε πρόσβαση στο ηλεκτρονικό μας κατάστημα.



Εικόνα 19: Εγκατάσταση του κυριότερου module του VirtueMart (mod_virtuemart_1.1.5.j15)

Με τον ίδιο τρόπο εγκαθιστούμε και ενεργοποιούμε και τα υπόλοιπα modules του VirtueMart (search module, frontpage categories module, XHTML product categories module).

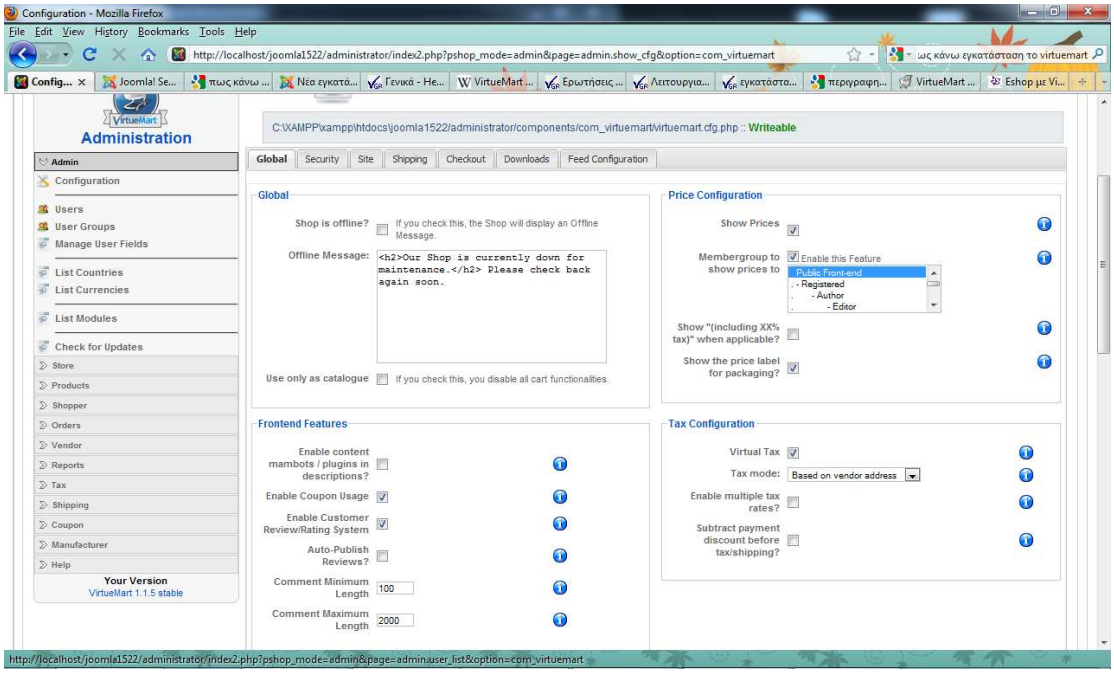


Εικόνα 20: VirtueMart Modules

6.3 Ρυθμίσεις Διαχείρισης του VirtueMart

Μόλις έχουμε υλοποιήσει όλα τα βήματα εγκατάστασης, μπορούμε από το top menu να έχουμε πρόσβαση στο πιο σημαντικό τμήμα του VirtueMart, το configuration

panel. Επιλέγοντας Components -> VirtueMart εμφανίζεται ένα νέο παράθυρο, στο οποίο επιλέγουμε το Configuration.



Εικόνα 21: Το Configuration Panel του VirtueMart, Global Settings

Στην παραπάνω εικόνα υπάρχουν επτά καρτέλες (Global, Security, Site, Shipping, Checkout, Downloads και Feed Configuration) με τις αντίστοιχες ρυθμίσεις του εγκατεστημένου component. Πιο αναλυτικά θα δούμε μόνο τις ρυθμίσεις που αφορούν την ασφάλεια στο κατάστημά μας.

Στην καρτέλα **Global** υπάρχουν οι γενικές ρυθμίσεις για το e-κατάστημα, από τις οποίες μας ενδιαφέρει:

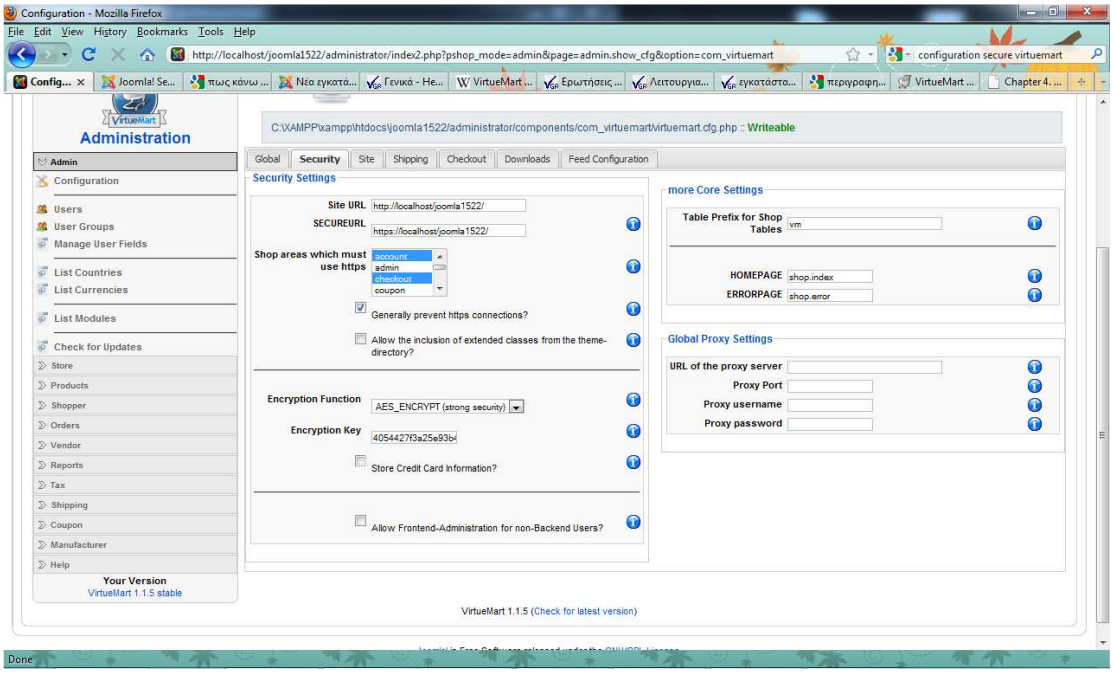
- Στο πλαίσιο του User Registration Settings μας ενδιαφέρει η ρύθμιση: **"Show the "Remember me" checkbox on login"**. Με αυτή την επιλογή, επιτρέπεται να ρυθμιστεί ένα cookie στον browser του πελάτη, έτσι ώστε να μην είναι αναγκαίο να κάνει login κάθε φορά που επισκέπτεται το site μας. Τέτοιου είδους cookies μπορεί να αποτελέσουν κίνδυνο για την ασφάλεια του site, ιδίως όταν χρησιμοποιούνται κοινόχρηστοι υπολογιστές (π.χ. σε κάποιο internet καφέ). Επιλέγουμε λοιπόν να μην αποθηκεύονται τα cookies των χρηστών.

Η επόμενη καρτέλα, είναι η καρτέλα **Security** στην οποία και ρυθμίζουμε την ασφάλεια στο component VirtueMart. Πιο αναλυτικά:

- **SECUREURL**: αν έχουμε επιλέξει το ηλεκτρονικό μας κατάστημα να χρησιμοποιεί SSL πιστοποιητικό ασφαλείας, θα πρέπει να πληκτρολογήσουμε το URL του site μας αρχίζοντας με https:// και τελειώνοντας με κάθετο (https://www.ptixiaki.me/). Σ' αυτό το πεδίο πρέπει να προσέξουμε πολύ, καθώς αν η διεύθυνση δεν είναι υπαρκτή, οι πελάτες θα λάβουν στις οθόνες τους σφάλμα 404.
- **Shop areas which must use https**: Κάποιες από τις περιοχές του site, όπως αυτή του login αναγκαστικά θα χρησιμοποιήσουν SECUREURL σύνδεση.

Έτσι εδώ επιλέγουμε τις ενότητες που εμείς θεωρούμε ότι πρέπει να χρησιμοποιούν το SECUREURL, και αυτές εξ'ορισμού είναι: "account" (Account Maintenance) και "checkout" (the complete Checkout).

- **Encryption Function**: Συνιστάται η επιλογή AES_ENCRYPT. Επιλέγουμε την MySQL λειτουργία, η οποία χρησιμοποιείται για να κωδικοποιήσει και να κρυπτογραφήσει σημαντικά στοιχεία στους πίνακες δεδομένων. Η AES κρυπτογράφηση είναι πολύ πιο ασφαλή, διότι στην πραγματικότητα κρυπτογραφεί τα δεδομένα, δεν τα κωδικοποιεί μόνο. Η AES κρυπτογράφηση είναι διαθέσιμη για MySQL > = 4.0.2.
- **Encryption Key**: Το μυστικό κλειδί για την κρυπτογράφηση των δεδομένων του λογαριασμού πληρωμών, όπως αριθμούς πιστωτικών καρτών και την αποθήκευση τους κρυπτογραφημένα στην βάση δεδομένων.
- **Store Credit Card Information?:** Επιτρέπει να απενεργοποιήσετε εντελώς την αποθήκευση των δεδομένων πιστωτικών καρτών, το απενεργοποιούμε.
- **Table Prefix for Shop Tables:** Αυτό είναι ένα πειραματικό χαρακτηριστικό που επιτρέπει πολλαπλά καταστήματα σε μια εγκατάσταση Joomla.
- **HOMEPAGE:** Αυτή είναι η σελίδα που θα φορτωθεί στο frontend από προεπιλογή. Συμπληρώνουμε shop.index.
- **ERRORPAGE:** Αυτή είναι η προεπιλεγμένη σελίδα για την εμφάνιση VirtueMart μηνύματα λάθους. Συμπληρώνουμε shop.error.



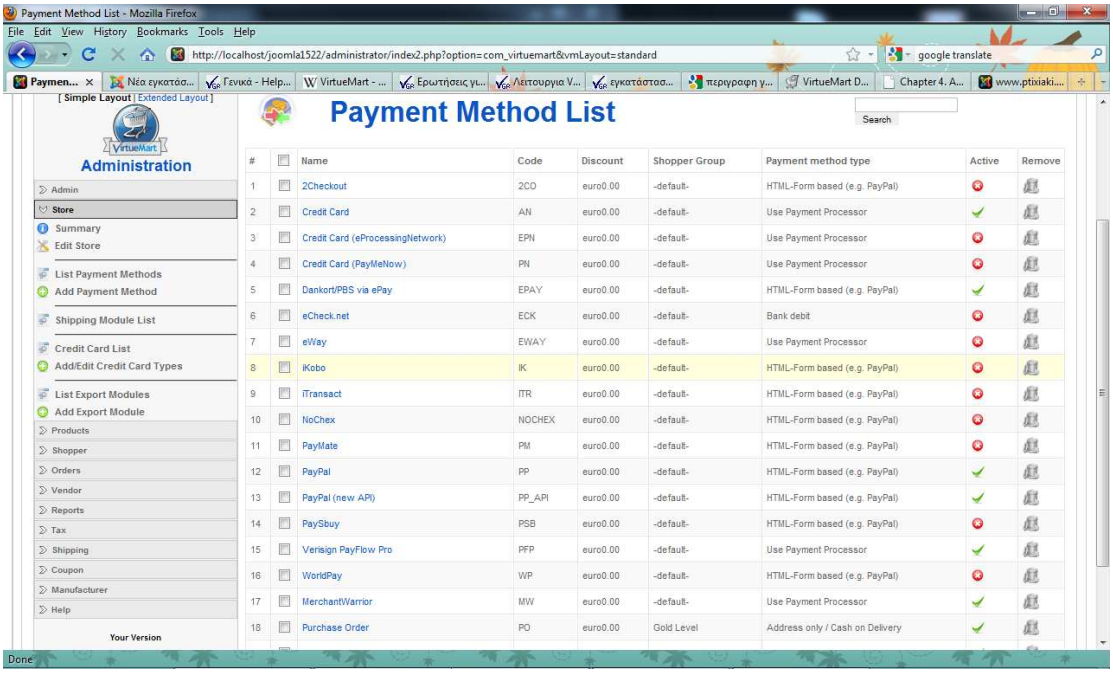
Εικόνα 22: Configuration Panel - VirtueMart - Security Settings

6.4 Ρυθμίσεις πληρωμών του ηλεκτρονικού καταστήματος

Αφού έχουμε ρυθμίσει τις βασικές πληροφορίες του καταστήματος μας, θα πρέπει να εφαρμόσουμε και ρυθμίσεις που αφορούν στο εμπόριο που θα διεξάγεται από την ιστοσελίδα μας.

Ας δούμε τους τρόπους πληρωμής που υποστηρίζει το VirtueMart. Από το αριστερό μενού “Store” επιλέγουμε “List Payment Methods” και βλέπουμε μια λίστα με όλες τις υποστηριζόμενες μεθόδους πληρωμής.

Παρατηρούμε ότι η λίστα των μεθόδων πληρωμής που μας προσφέρει το component που επιλέξαμε να εγκαταστήσουμε είναι αρκετά μεγάλη, δίνοντας τη δυνατότητα στον ιδιοκτήτη του ηλεκτρονικού καταστήματος να επιλέξει όποιες μεθόδους εξυπηρετούν με τον καλύτερο τρόπο την επιχείρησή του.



Εικόνα 23: Μέθοδοι πληρωμής που υποστηρίζει το VirtueMart

Ας δούμε λίγο πιο αναλυτικά ορισμένες από αυτές τις μεθόδους:

- **PayPal** [9]: Είναι μια online υπηρεσία μεταφοράς χρηματικών ποσών που χρησιμοποιείται ευρέως για ασφαλείς διαδικτυακές συναλλαγές. Η εταιρία που το λειτουργεί, είναι συγχρόνως και ιδιοκτήτρια του eBay, γι' αυτό και οι περισσότερες συναλλαγές στο eBay εξοφλούνται μέσω PayPal. Η λειτουργία του είναι όπως αυτή ενός τραπεζικού λογαριασμού, ο χρήστης δηλαδή μπορεί να στείλει ή να δεχτεί από κάποιον άλλο χρήστη χρήματα ή να βάλει χρήματα στο λογαριασμό του μέσω κάρτας. Παρέχει 100% ασφάλεια καθώς ο μόνος γνώστης των στοιχείων του εκάστοτε χρήστη είναι το ίδιο το PayPal,ο παραλήπτης παραλαμβάνει τα χρήματα και όχι τον αριθμό της κάρτας και όλες οι σελίδες που χρησιμοποιούνται για τις συναλλαγές είναι κρυπτογραφημένες.
- **Authorize.net**[10]: Διαχειρίζεται τις online πληρωμές μέσω πιστωτικών καρτών όπως ακριβώς ένα κλασσικό μηχανάκι χρέωσης πιστωτικών καρτών.
- **2Checkout** [11] : Λειτουργεί σαν εξουσιοδοτημένος μεταπωλητής και περιλαμβάνει μεθόδους όπως οικονομικές αναφορές, πρόληψη από απάτες,

[9] http://en.wikipedia.org/wiki/Paypal
[10] http://www.authorize.net/company/whatwedo/

παρακολούθηση θυγατρικών εταιριών, εξυπηρέτηση πελατών και παρακολούθηση πωλήσεων. Ο ιδιοκτήτης του e-καταστήματος συμπράττει σύμβαση με την 2CO, εισάγει τα προϊόντα του στη βάση δεδομένων των προϊόντων της 2CO και αυτόματα δημιουργούνται σύνδεσμοι στον διαδικτυακό του ιστότοπο. Όταν κάποιος αγοραστής επιλέξει να πληρώσει το προϊόν που επέλεξε, η 2CO χειρίζεται την πώληση σε ασφαλές περιβάλλον και καταθέτει το ποσό που αναλογεί στον ιδιοκτήτη στο λογαριασμό του. Visa, American Express, PayPal, UCB, Master Card, Discover είναι τύποι πιστωτικών καρτών που δέχεται.

- **eWay** [12] : Προσφέρει μια κορυφαία λύση getaway πληρωμών για την επεξεργασία πληρωμών σε πραγματικό χρόνο. Με την eWay γίνονται ασφαλής online αποπληρωμές με πιστωτικές κάρτες αλλά και μέσω mail.
- **WorldPay**[13]: Είναι ένα τμήμα της Royal Bank of Scotland, το οποίο παρέχει υπηρεσίες πληρωμών για παραγγελίες μέσω mail καθώς και διαδικτυακές συναλλαγές. Οι πελάτες είναι κυρίως πολυεθνικές, αλλά και πολλά κανάλια λιανικής πώλησης. Η RBS WorldPay ξεκίνησε ως πάροχος ηλεκτρονικής πληρωμής που ονομάστηκε Streamline το 1989, αλλά έχει επεκταθεί σε Mail Order/Telephone Order, "ανεπιτήρητες" πληρωμές και διακίνηση ασφαλών πληρωμών μέσω του Διαδικτύου, μέσω συγχωνεύσεων και εξαγορών πολλών άλλων εταιρειών.
- **Paymate**[14]: Είναι ένας online πάροχος πληρωμών από την Αυστραλία που άρχισε να λειτουργεί τον Οκτώβριο του 2001. Η υπηρεσία είναι παρόμοια με το PayPal των ΗΠΑ, με τη διαφορά ότι οι πιστώσεις των κεφαλαίων χρεώνονται απευθείας στον τραπεζικό λογαριασμό του δικαιούχου. Το Paymate δίνει τη δυνατότητα στους πωλητές της Αυστραλίας να δέχονται πληρωμές σε δολάρια Αυστραλίας, δολάρια, λίρες, ευρώ και NZD ενώ οι πωλητές στη Νέα Ζηλανδία μπορεί να δεχθούν πληρωμές σε NZD.
- **Nochex**[15]: Είναι άλλος ένας online πάροχος πληρωμών, ο οποίος έχει βάση το Ηνωμένο Βασίλειο και εξειδικεύεται στην παροχή των επιχειρήσεων σε απευθείας σύνδεση με τις υπηρεσίες πληρωμών και στην άμεση μεταφορά χρημάτων μεταξύ ιδιωτών μέσω του Διαδικτύου. Προσφέρει τρεις τύπους λογαριασμού: προσωπικό, πωλητή και έμπορου και στον καθένα τύπο έχει και όριο για συναλλαγές.
- **Cash On Delivery**: Στη συγκεκριμένη συναλλαγή (η γνωστή σε όλους μας αντικαταβολή), η πληρωμή για ένα αγαθό δεν γίνεται διαδικτυακά αλλά χέρι-χέρι. Σε περίπτωση που ο αγοραστής δεν καταβάλλει το αντίστοιχο ποσό του κόστους του αγαθού τότε το προϊόν επιστρέφεται στον πωλητή.

Εφόσον είδαμε, επιγραμματικά, μερικούς από τους τρόπους πληρωμής που υποστηρίζει το VirtueMart, ας συνεχίσουμε με τις ρυθμίσεις που απαιτούνται σε αυτό το στάδιο. Παρατηρούμε πως υπάρχουν δύο καρτέλες: στην πρώτη επιλέγουμε κάποια από τις έτοιμες μεθόδους (όπως και αυτές που αναφέραμε πιο πάνω), και στη δεύτερη καρτέλα έχουμε τη δυνατότητα να επιλέξουμε να δημιουργήσουμε εμείς μια καινούργια μέθοδο.

---

[11] http://www.2checkout.com/how
[12] http://www.eway.com.au/about-eway-payment-service-provider/eway-explained/
[13] http://en.wikipedia.org/wiki/WorldPay
[14] http://en.wikipedia.org/wiki/Paymate
[15] http://en.wikipedia.org/wiki/Nochex

Στην πρώτη περίπτωση θα πρέπει να επιλέξουμε τις μεθόδους που επιθυμούμε και κατόπιν να συνδεθούμε με τους payment processors. Στη δεύτερη περίπτωση, υπεύθυνος είναι ο διαχειριστής για την εγκατάσταση και την ενεργοποίηση του Plugin που απαιτείται (συνήθως δίνονται από τον πάροχο κάποιο username και password που πρέπει να γραφτεί στο Payment Method Editor).

Ας επιλέξουμε να δίνεται η δυνατότητα στο χρήστη να μπορεί να πληρώσει μέσω PayPal, πιστωτικών καρτών και με αντικαταβολή. Μένει να «τοποθετήσουμε» στα ηλεκτρονικά «ράφια» τα προϊόντα μας και το ηλεκτρονικό μας κατάστημα είναι έτοιμο…ασφαλές όμως είναι; Στα επόμενα κεφάλαια θα ασχοληθούμε με τις κοινές επιθέσεις και τους τρόπους θωράκισης του e-καταστήματός μας.

ΚΕΦΑΛΑΙΟ 7

ΤΑ ΕΠΙΚΙΝΔΥΝΑ ΣΗΜΕΙΑ ΓΙΑ ΤΗΝ ΑΣΦΑΛΕΙΑ ΤΗΣ ΙΣΤΟΣΕΛΙΔΑΣ ΜΑΣ

Στις μέρες μας οι διαδικτυακές συναλλαγές έχουν αυξηθεί πάρα πολύ, με ανάλογη αύξηση και στον αριθμό και στον τύπο των επιθέσεων που δέχεται η ασφάλεια των ηλεκτρονικών συστημάτων πληρωμής. Κάποιες επιθέσεις έχουν στόχο το λογισμικό των καλαθιών αγοράς και κάποιες άλλες χρησιμοποιούν ευπάθειες που είναι κοινές σε οποιαδήποτε δικτυακή εφαρμογή, όπως τη χρήση της SQL ή τη συγγραφή τμημάτων κώδικα και την παράθεσή τους σε διάφορα σημεία του site. Ας εξετάσουμε κάποιες κοινές ευπάθειες ασφαλείας.

7.1 Υπερχειλίσεις Μνήμης (Buffer Overflows)

Οι χάκερ χρησιμοποιούν πολλαπλά μέσα για να δημιουργήσουν δυσλειτουργίες σε μια εφαρμογή. Η υπερχείλιση μνήμης[16] είναι ένα από τα δημοφιλέστερα μέσα να το κάνουν. Στην υπερχείλιση, ένας χάκερ επιβαρύνει το διακομιστή με την προσθήκη ενός ελαττώματος σε κάποια διαδικτυακή φόρμα και την αποστολή επιπλέον πληροφοριών. Αν γίνει υπερχείλιση των ορίων, μπορεί να προκαλέσει μέχρι και κατάρρευση του συστήματος. Τα Buffer Overflows έχουν γίνει ένα πολύ δημοφιλές εργαλείο hacking που χρησιμοποιείται από χάκερ σήμερα.

Πώς ένας χάκερ εκτελεί το σύνολο της πράξης; Πολλές ιστοσελίδες εμφανίζουν φόρμες πρέπει να συμπληρωθούν από τους επισκέπτες της ιστοσελίδας. Για παράδειγμα, μια περιοχή ηλεκτρονικού εμπορίου ζητά από τους πελάτες να συμπληρώσουν ένα έντυπο με προσωπικές πληροφορίες κατά τη διάρκεια της εγγραφής. Κάθε ειδικό πεδίο στη φόρμα εγγραφής δέχεται ένα μέγιστο αριθμό χαρακτήρων (αναφέρεται στον πηγαίο κώδικα HTML).

Ένας έξυπνος χάκερ μπορεί να διατυπώσει ορισμένες μετατροπές στη ρύθμιση του πηγαίου κώδικα και να επιτρέψει το πρόγραμμα περιήγησης να αυξήσει το όριο των χαρακτήρων που δέχεται. Στη συνέχεια πάει πίσω στη φόρμα, πληκτρολογεί μεγαλύτερο χαρακτήρα στο συγκεκριμένο πεδίο και την υποβάλλει. Κατά την υποβολή της φόρμας, η εφαρμογή μπλοκάρει και δεν μπορεί να αντεπεξέλθει, καθώς η αίτηση δεν είχε ως σκοπό να αποδεχθεί τέτοιο μεγάλο χαρακτήρα. Έτσι, αναγκάζει σε τμήματα υπερχείλιση της μνήμης του. Επίσης, μπορεί να έχει ως αποτέλεσμα την συντριβή του συστήματος.

Ποιες είναι οι έσχατες συνέπειες της πράξης αυτής;

Η υπερχείλιση μνήμης είναι ο καλύτερος τρόπος για να συντρίψει το σύστημα ή να κάνει μια εφαρμογή και να εκτελούνται εντολές για λογαριασμό του χάκερ. Αυτή η

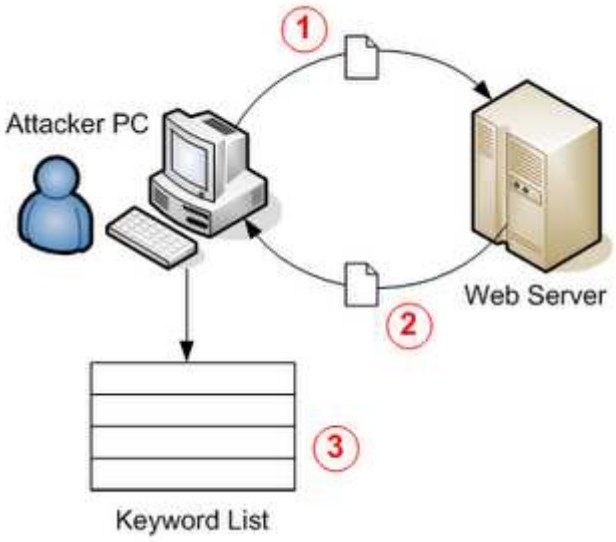
[16] http://www.hacker4lease.com/attack-methods/buffer-overflow/

τεχνική μπορεί να κάνει ένα διακομιστή δυσλειτουργικό, και να σταματήσει η λειτουργία της ιστοσελίδας. Επίσης μπορεί να χρησιμοποιηθεί για να εκτελέσει ο κακόβουλος χρήστης δικό του κώδικα, με ό,τι αυτό συνεπάγεται.

Μια δυνατή προστασία από τέτοιου είδους επιθέσεις είναι να θωρακίζουμε το σύστημα με εξελιγμένες εκδόσεις των καθιερωμένων πρωτοκόλλων. Αυτό βοηθά στη μείωση του κινδύνου προσβολής από αυτές τις επιθέσεις. Οι εφαρμογές που έχουν φόρμες συμπληρώματος από τους χρήστες πρέπει να είναι σε θέση να διαχειριστούν μια υπερχείλιση των εισροών, είτε με τη χρήση προσωρινού χώρος για να πετάξει τα δεδομένα που αποτελούν πλεόνασμα, ή να απορρίπτουν την υποβολή των εισροών πλεονάσματος με συστηματικό τρόπο.

7.2 SQL injections

Μια από τις πιο συνηθισμένες και συγχρόνως και πιο επικίνδυνες επιθέσεις για μια ιστοσελίδα, είναι οι επιθέσεις SQL Injection[17], γιατί στην ουσία επιτρέπει στους εισβολείς του συστήματος να κλέψουν ζωτικής σημασίας δεδομένα που βρίσκονται αποθηκευμένα στη βάση (όπως για παράδειγμα ένας εισβολέας να κάνει επίθεση στον server και να καταφέρει να κλέψει τους κωδικούς πρόσβασης – βλέπε εικόνα 24).



Εικόνα 24: Παράδειγμα υποκλοπής κωδικών

Στις ιστοσελίδες δίνεται η δυνατότητα στους χρήστες μέσω κάποιον φορμών να εισάγουν δεδομένα (όπως για την εγγραφή τους, το login, αναζήτηση, καροτσάκια αγορών κ.α.). Μέσω αυτών των κελιών λοιπόν, μπορεί κάποιος κακόβουλος χρήστης να εισάγει τμήματα κώδικα SQL και αν δεν είναι σωστά θωρακισμένη η βάση μας να μπορέσει να εξαγάγει πληροφορίες ζωτικής σημασίας (στατιστικά στοιχεία της εταιρίας, πληροφορίες πληρωμής και άλλες πληροφορίες χρηστών).

[17] http://www.hacker4lease.com/attack-methods/sql-injection/

Ένας hacker που κάνει χρήση της SQL Injection PHP, ASP,. NET, Java μπορεί να αποκτήσει τον πλήρη πρόσβαση στη βάση δεδομένων της ιστοσελίδας και να πάρει κάθε είδους πληροφορίες που συλλέγονται.

Ένα απλό παράδειγμα για SQL Injection[18] είναι το παρακάτω, στο οποίο script υπάρχει ένα SQL ερώτημα με τη σύνδεση κωδικοποιημένων strings μαζί με ένα string που εισάγεται από το χρήστη. Από τον χρήστη έχει ζητηθεί να πληκτρολογήσει το όνομα μιας πόλης (Redmond).

APPLICATION CODE

```
var shipcity;
ShipCity = Request.form (“Shipcity”)
var sql = “SELECT * FROM OrdersTable
     WHERE ShipCity = ‘” + Shipcity + “’”;
```

GOOD USER

```
Inputs Redmond in the form
Query to back-end is:
SELECT * FROM OrdersTable WHERE ShipCity = ‘Redmond’
```

;Ομως ένας κακόβουλος χρήστης αν γράψει τον παρακάτω κώδικα (και εφόσον είναι συντακτικά σωστός) και εκτελεστεί από τον διακομιστή, θα περάσει και στον SQL server μας.

Το ερωτηματικό “;” στον παρακάτω κώδικα υποδηλώνει το τέλος ενός ερωτήματος και την έναρξη ενός άλλου, η διπλή παύλα “- -“ σημαίνει ότι το υπόλοιπο της τρέχουσας γραμμής είναι σχόλιο και θα πρέπει να αγνοηθεί. Όταν λοιπόν, ο SQL Server επεξεργαστεί αυτή τη δήλωση, θα επιλέξει πρώτα όλα τα αρχεία από τον πίνακα OrdersTable, όπου το πεδίο ShipCity είναι Redmond. Στη συνέχεια, ο SQL Server θα φέρει όλο τον πίνακα OrdersTable και θα τον εμφανίσει στον εισβολέα.

MALICIOUS USER

```
Inputs the following in the form:
Redmond’ DROP TABLE OrderTable –
Query to the back-end is:
SELECT * FROM OrdersTable WHERE ShipCity = ‘Redmond’
DROP TABLE OrdersTable--’
```

7.3 Phishing

Όπως το ίδιο το όνομά του υπονοεί -παραλλαγή του αγγλικού «fishing» (ψάρεμα), το Phishing [19] αναφέρεται στην προσπάθεια απόσπασης προσωπικών στοιχείων, οικονομικού συνήθως χαρακτήρα που αφορούν τραπεζικούς λογαριασμούς και πιστωτικές κάρτες, χρησιμοποιώντας ως δόλωμα κάποιο ψεύτικο πρόσχημα.

Το Phishing επιχειρείται συνήθως με τη αποστολή κάποιου spam email (*), το οποίο ισχυρίζεται –ψευδώς- ότι αποστέλλεται από κάποια υπαρκτή και νόμιμη

[18] www.netlib.com/files/SQL_SecurityWebcast.ppt
[19] http://www.forthnet.gr/templates/viewcontentTmCh.aspx?c=10009043

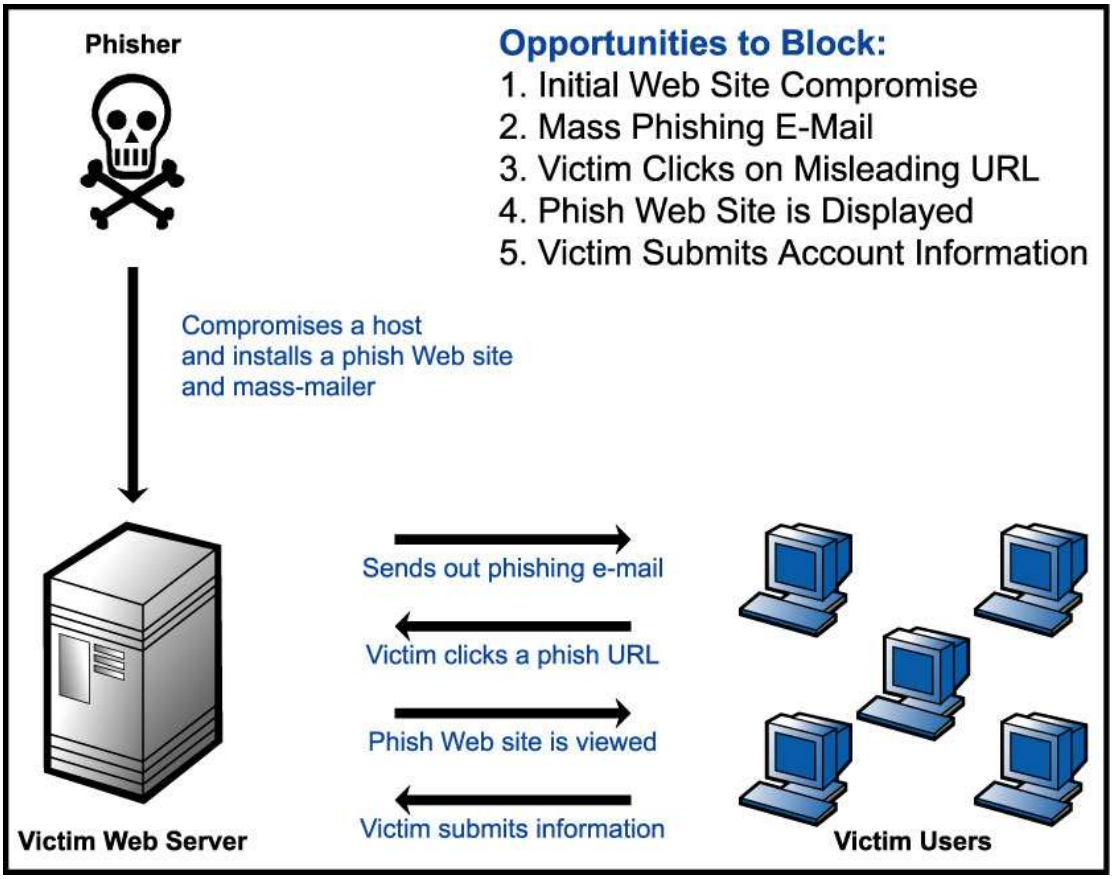
εταιρεία (τράπεζα, ηλεκτρονικό κατάστημα, υπηρεσία ηλεκτρονικών πληρωμών κλπ.), σε μία προσπάθεια να παραπλανήσει τον παραλήπτη και να του αποσπάσει απόρρητα προσωπικά και οικονομικά δεδομένα. Στη συνέχεια, τα στοιχεία αυτά θα χρησιμοποιηθούν από τους εγκέφαλους της απάτης για την πραγματοποίηση μη εξουσιοδοτημένων/ παράνομων οικονομικών συναλλαγών.

Τα email αυτά ισχυρίζονται ότι ο παραλήπτης απαιτείται να ενημερώσει ή να επαληθεύσει άμεσα κάποια προσωπικά στοιχεία του για λόγους ασφαλείας, και τον οδηγούν μέσω συνδέσμων σε πλαστά web sites, τα οποία μιμούνται πολύ πειστικά τους διαδικτυακούς τόπους υπαρκτών και αξιόπιστων οργανισμών. Σε κάποιες περιπτώσεις η αντιγραφή είναι τόσο καλή που και ο ίδιος ο internet browser «ξεγελιέται» και δείχνει στην γραμμή θέματος την αναμενόμενη διεύθυνση και όχι την πραγματική διεύθυνση της πλαστής διαδικτυακής τοποθεσίας.

Σε μία προσπάθεια να μειώσουν τον χρόνο αντίδρασης του ανυποψίαστου παραλήπτη, ορισμένα μηνύματα απειλούν ότι εάν δεν προβεί στις απαιτούμενες ενέργειες (ενημέρωση, επαλήθευση στοιχείων) εντός του υποδεικνυόμενου –σύντομου- χρονικού διαστήματος ο λογαριασμός του θα μπλοκαριστεί και δεν θα μπορεί να πραγματοποιήσει περαιτέρω συναλλαγές. Σκοπός τους είναι να εξαναγκάσουν τον παραλήπτη να αποκαλύψει τις πληροφορίες που του ζητείται χωρίς καν να προλάβει να εξετάσει την γνησιότητα του μηνύματος. Έτσι οι Phishers χρησιμοποιούν αυτές τις πληροφορίες, προσποιούνται την ταυτότητα των θυμάτων και έχουν πρόσβαση σε όλες τις κινήσεις του χρήστη και ενεργούν εις βάρος του.

Χρειάζεται ιδιαίτερη προσοχή ώστε ο παραλήπτης ενός τέτοιου μηνύματος να αποφύγει την εξαπάτηση μέσω Phishing. Τα email που αποστέλλονται μοιάζουν αρκετά επίσημα και οι πλαστές σελίδες είναι τις περισσότερες φορές πανομοιότυπες με τις πραγματικές, αφού δημιουργούνται με αντιγραφή του HTML κώδικά τους.

Μια ανάλογη περίσταση, παρουσιάζεται στην εικόνα 25. Ένας phisher δημιουργεί έναν ιστότοπο πανομοιότυπο αυτού που θέλει να οικειοποιηθεί και χρησιμοποιώντας μαζικά e-mail «ψαρεύει» χρήστες, αποστέλλοντας τους μια URL διεύθυνση μέσω του ηλεκτρονικού τους ταχυδρομείου. Ο χρήστης-θύμα επιλέγοντας το link που του έχει αποσταλεί, μεταβαίνει σε μια ηλεκτρονική τοποθεσία που συνήθως παρουσιάζεται -πολύ πειστικά- ως γνήσια. Ο χρήστης-θύμα εισάγει τα προσωπικά του στοιχεία, τα οποία αποστέλλονται στον υπολογιστή του phisher.



Εικόνα 25: Παράδειγμα Phishing

7.4 Hidden Manipulation-Κρυφή Χειραγώγηση-Παραποίηση Τιμών

Η κρυφή χειραγώγηση (Hidden Manipulation[20]) πρόκειται για μια συνηθισμένη, επίσης, ευπάθεια η οποία εμφανίζεται στα καρότσια αγορών και στις διαδικασίες πληρωμής. Όταν αναφερόμαστε στην συγκεκριμένη ευπάθεια, στην πιο γνωστή μορφή της, η συνολική πληρωτέα τιμή των αγαθών αποθηκεύεται σε έναν κρυμμένο πεδίο HTML μιας δυναμικά φτιαγμένης ιστοσελίδας. Έτσι ένας επιτιθέμενος χρησιμοποιώντας έναν απλό Netscape HTML Editor ή το Achilles του δίνεται η δυνατότητα να τροποποιήσει το πληρωτέο ποσό, καθώς οι πληροφορίες από τον browser του χρήστη οδηγούνται στον server του δικτύου.

Μετά την κρυφή χειραγώγηση, η τελική πληρωτέα τιμή μπορεί να τροποποιηθεί από τον επιτιθέμενο σε μια αξία της επιλογής του. Αυτές οι πληροφορίες στέλνονται τελικά στην έξοδο πληρωμής με την οποία ο «ηλεκτρονικός» έμπορος συνεργάζεται. Εάν ο αριθμός των συναλλαγών είναι πολύ μεγάλος, η παραποίηση τιμών μπορεί να περάσει απαρατήρητη, ή μπορεί να ανακαλυφθεί πάρα πολύ αργά. Οι επαναλαμβανόμενες επιθέσεις αυτής της φύσης θα μπορούσαν ενδεχομένως να «ακρωτηριάσουν» τη βιωσιμότητα του ηλεκτρονικού εμπόρου.

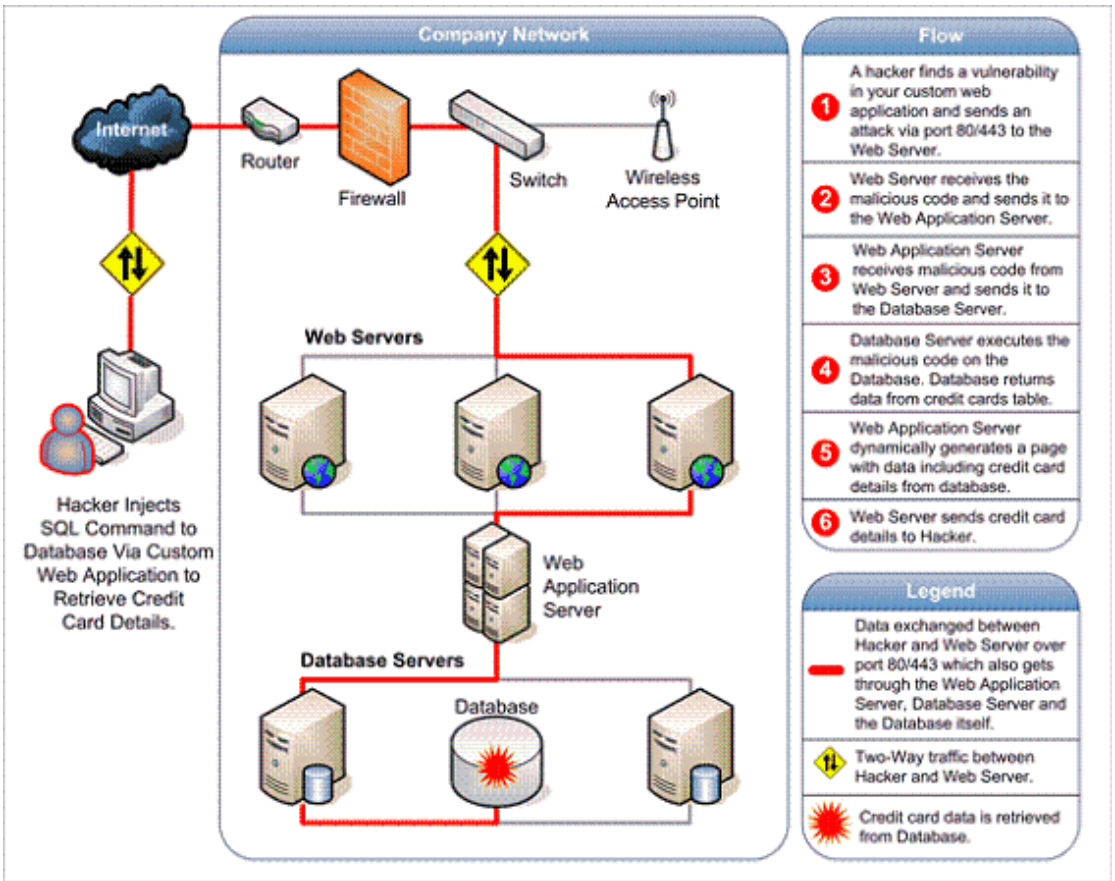
[20] http://www.hacker4lease.com/attack-methods/hidden-manipulation/

Οι επιθέσεις τέτοιου τύπου κάνουν τους ιδιοκτήτες ηλεκτρονικών καταστημάτων να ανησυχούν ιδιαιτέρως και να χρησιμοποιούν πολλές τεχνικές ασφαλείας όπως η χρησιμοποίηση ηχητικού λογισμικού κατά τω ιών, τοίχου προστασίας και τα πιο πρόσφατα λογισμικά εντοπισμού εισβολέων, μέχρι και στην απόκρυψη υψηλού κινδύνου πεδίων. Ακόμη όμως και σε αυτή την περίπτωση, αν ο επιτιθέμενος έχει καλή γνώση προγραμματισμού μπορεί να αποκαλύψει τα πεδία και τα δεδομένα τους και να τα εκμεταλλευτεί.

Επιθέσεις παραποίησης τιμών, μπορούν και παραποιούν πληροφορίες ζωτικής σημασίας για τα ηλεκτρονικά καταστήματα και οι ιδιοκτήτες να έρθουν αντιμέτωποι με τεράστιες ζημίες (συνήθως οικονομικές). Μια σειρά τέτοιων επιθέσεων εξαφανίζει την εμπιστοσύνη και την αξιοπιστία στους πελάτες.

7.5 Cross Site Scripting (XSS)

Η cross-site scripting(ή XSS[21]) επίθεση είναι μια από τις πιο συχνές web επιθέσεις που γίνονται στο application-layer. Συχνοί στόχοι είναι τα πεδία που χρησιμοποιεί ο χρήστης της σελίδας (π.χ. φόρμες που συμπληρώνει), χειραγωγεί δηλαδή τα scripts της εφαρμογής για να εκτελεστούν με τον τρόπο που επιθυμεί ο κακόβουλος χρήστης. Ένας τέτοιος χειρισμός μπορεί να ενσωματώσει ένα script, το οποίο θα εκτελείται κάθε φορά που φορτώνεται η σελίδα ή κάθε φορά που θα τρέχει ένα σχετικό event.



Εικόνα 26: Επίθεση hacker με σκοπό να κλέψει στοιχεία πιστωτικών καρτών

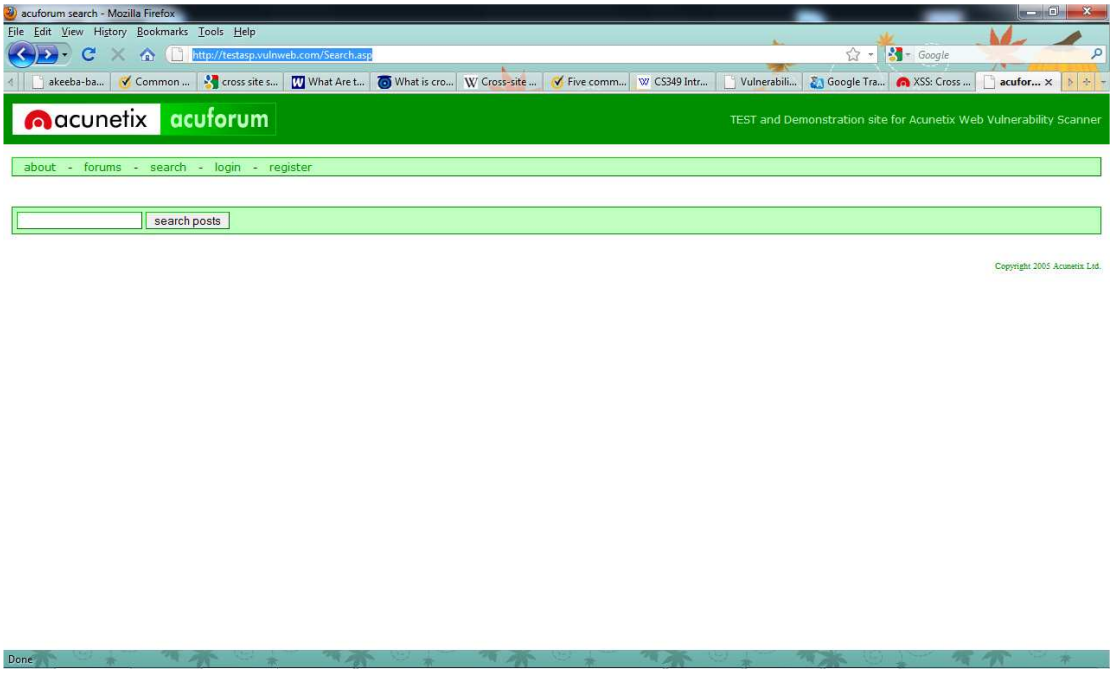
[21] http://www.acunetix.com/websitesecurity/xss.htm

Ένα βασικό παράδειγμα είναι όταν ένας κακόβουλος χρήστης παρεμβάλλει ένα script σε μια νόμιμη διεύθυνση URL ενός ηλεκτρονικού καταστήματος, ανακατευθύνοντας τον χρήστη σε ένα ψεύτικο site το οποίο όμως είναι πανομοιότυπο με το αρχικό. Η κακόβουλη σελίδα χρησιμοποιώντας κάποιο script και κλέβει το cookie του χρήστη την ώρα που σερφάρει ανυποψίαστος στη σελίδα. Παρόλο που δεν έχει γίνει επίθεση στο e-κατάστημα, ο εισβολέας έχει καταφέρει να παγιδεύσει τον χρήστη. Συνηθισμένο τέχνασμα τέτοιων επιθέσεων είναι η εμφάνιση ενός URL πανομοιότυπο με το κανονικό, έτσι ο χρήστης να ξεγελαστεί (αφού η διεύθυνση θα του φαίνεται γνωστή) και θα πέσει στην παγίδα του εισβολέα.

Στην συγκεκριμένη επίθεση μπορεί να μην κινδυνεύει το ίδιο το site μας στο back-end, αλλά σίγουρα κινδυνεύει να χαθεί η εμπιστοσύνη, η αξιοπιστία και οι πωλήσεις μας, καθώς οι πελάτες θα σταματήσουν τις αγορές και θα στραφούν προς άλλα ηλεκτρονικά καταστήματα που θα τους προσφέρουν την ασφάλεια των δεδομένων τους (και των χρημάτων τους).

Ας δούμε ένα παράδειγμα παρεμβολής κώδικα σε μηχανή αναζήτησης. Το παράδειγμα είναι φτιαγμένο από την http://www.acunetix.com και είναι μεταφερμένο όπως ακριβώς το παρουσιάζει. Το παράδειγμα δεν είναι hacking tutorial, αλλά ένας βασικός τρόπος για να δείξει πως η XSS μπορεί να χρησιμοποιηθεί για τον έλεγχο και την τροποποίηση της λειτουργικότητας σε μια ιστοσελίδα, τον επανασχεδιασμό της και της διαδικασίες παραγωγής της.

Τοποθετούμε τον ακόλουθο σύνδεσμο στον περιηγητή μας: http://testasp.vulnweb.com/Search.asp και βλέπουμε μια απλή ιστοσελίδα με μια φόρμα αναζήτησης.

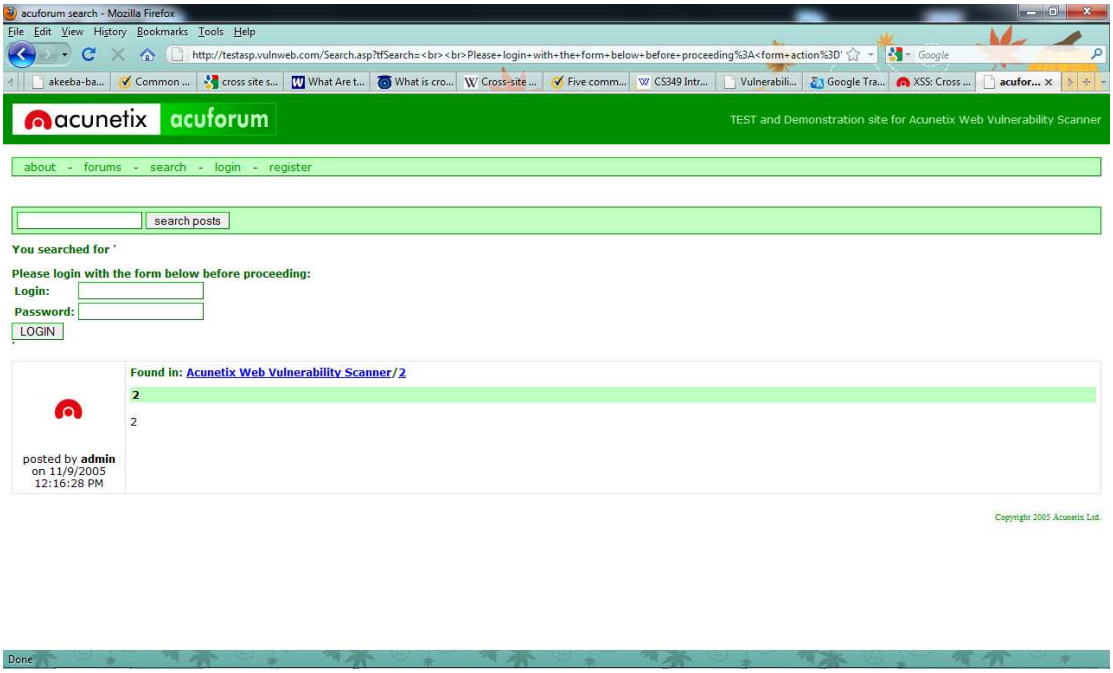


Εικόνα 27: Αρχική σελίδα δοκιμής παρεμβολής κώδικα

Στο κελί που μας επιτρέπει την αναζήτηση τοποθετούμε τον παρακάτω κώδικα:

```
<br><br>Please login with the form below before
proceeding:<form
action="destination.asp"><table><tr><td>Login:</td><
td><input type=text length=20
name=login></td></tr><tr><td>Password:</td><td><inpu
t type=text length=20
name=password></td></tr></table><input type=submit
value=LOGIN></form>
```

Κάνουμε κλικ στο search posts και παίρνουμε την παρακάτω σελίδα:



Εικόνα 28: Η σελίδα μετά τον κακόβουλο κώδικα

Μέσω XSS στη σελίδα, δημιουργήσαμε μια ψεύτικη φόρμα σύνδεσης χρήστη, από την οποία μπορούμε να συλλέξουμε τα στοιχεία του χρήστη. Ο κώδικας αυτός περιέχει ένα τμήμα που αναφέρεται σε “destination.asp”, ώστε να επιτρέψει στον χάκερ να μπορεί να αποφασίσει αν η πλαστή φόρμα σύνδεσης θα στείλει το login του χρήστη για να ανακτηθεί και να χρησιμοποιηθεί κακόβουλα.

Ο επιτιθέμενος θα μπορούσε να στείλει σε ανυποψίαστους χρήστες τη φόρμα με την μορφή συνδέσμου:
“http://testasp.vulnweb.com/Search.asp?tfSearch=%3Cbr%3E%3Cbr%3EPlease+login+with+the+form+below+before+proceeding%3A%3Cform+action%3D%22test.asp%22%3E%3Ctable%3E%3Ctr%3E%3Ctd%3ELogin%3A%3C%2Ftd%3E%3Ctd%3E%3Cinput+type%3Dtext+length%3D20+name%3Dlogin%3E%3C%2Ftd%3E%3C%2Ftr%3E%3Ctr%3E%3Ctd%3EPassword%3A%3C%2Ftd%3E%3Ctd%3E%3Cinput+type%3Dtext+length%3D20+name%3Dpassword%3E%3C%2Ftd%3E%3C%2Ftr%3E%3C%2Ftable%3E%3Cinput+type%3Dsubmit+value%3DLOGIN%3E%3C%2Fform%3E”

Ο παραπάνω κώδικας θα δημιουργήσει το ίδιο παράθυρο με πριν, αποδεικνύοντας μας πως οι επιθέσεις XSS μπορούν να χρησιμοποιηθούν με πολλούς διαφορετικούς τρόπους. Μετά από μια τέτοια επίθεση και αφού ο εισβολέας αποκτήσει τα στοιχεία σύνδεσης του χρήστη, μπορεί να αναγκάσει τον περιηγητή μας να επιστρέψει στην αρχική σελίδα και το θύμα να μην καταλάβει ότι εξαπατήθηκε.

Για παράδειγμα, αυτό μπορεί να χρησιμοποιηθεί και σε spam e-mail. Ο επιτιθέμενος να αποστέλλει ένα μήνυμα ηλεκτρονικού ταχυδρομείου στο οποίο θα ισχυρίζεται πως ένας ιστότοπος, στον οποίο το θύμα είναι επικυρωμένος χρήστης, υποψιάζεται πως κάποιος άλλος χρήστης έχει εισέλθει με τον λογαριασμό του χρήστη-θύμα και να ζητάει από τον χρήστη να κάνει κλικ στον σύνδεσμο για να δώσει τα στοιχεία του για επαλήθευση.

7.6 Packet Sniffer (Παρακολούθηση πακέτων)

Packet sniffer[22] ή απλώς sniffer, επίσης αποκαλούμενο network monitor ή network analyzer, είναι λογισμικό με δυνατότητα παρακολούθησης των πακέτων ενός δικτύου. Όταν γίνει αντιληπτό κάποιο πακέτο το οποίο ικανοποιεί συγκεκριμένα κριτήρια, καταγράφεται σε ένα αρχείο.

Εικόνα 29: Packet Sniffer

Για πολύ καιρό οι μηχανικοί δικτύων, διαχειριστές συστημάτων και επαγγελματίες στον τομέα της ασφάλειας, αλλά και crackers, κάνουν χρήση ανάλογων εργαλείων. Χρησιμοποιείται νόμιμα από τους πρώτους για καταγραφή και διορθώσεις στην κίνηση (traffic) του δικτύου.

Οι περισσότεροι προσωπικοί υπολογιστές συνδέονται σε ένα Τοπικό δίκτυο (Local Area Network - LAN), που σημαίνει ότι μοιράζονται μία σύνδεση με άλλους υπολογιστές. Αν το δίκτυο δεν χρησιμοποιεί switch (μεταγωγέας, είναι μια συσκευή που φιλτράρει και ξαναστέλνει τα πακέτα ανάμεσα στους τομείς ενός LAN) η κίνηση που προορίζεται για έναν τομέα μεταδίδεται σε κάθε μηχάνημα του δικτύου. Επακόλουθα, κάθε υπολογιστής στην πραγματικότητα βλέπει τα δεδομένα που προέρχονται από ή προορίζονται για τους γειτονικούς υπολογιστές, αλλά τα αγνοεί.

[22] http://el.wikipedia.org/wiki/Packet_sniffer

Το sniffer[23] αναγκάζει τον υπολογιστή, συγκεκριμένα την Network Interface Card (κάρτα δικτύου-NIC), να αρχίσει να προσέχει και αυτά τα πακέτα, τα οποία προορίζονται για άλλους υπολογιστές. Για να το καταφέρει αυτό θέτει τη NIC σε ειδική λειτουργία, γνωστή ως promiscuous mode. Όταν η NIC βρίσκεται σε αυτή τη λειτουργία, μια κατάσταση που συνήθως απαιτεί δικαιώματα ανώτερου χρήστη (root), ένα μηχάνημα μπορεί να βλέπει όλα τα δεδομένα που μεταδίδονται στον τομέα του.

Υπάρχουν πολλές δυνατότητες, που καθορίζουν την τύχη των πακέτων:

- Τα πακέτα μετριούνται. Με αυτό τον τρόπο, προσθέτοντας στη συνέχεια το συνολικό μέγεθός τους για μία ορισμένη χρονική περίοδο (συμπεριλαμβάνοντας τις επικεφαλίδες των πακέτων), εξάγεται μια καλή ένδειξη για το πόσο φορτωμένο είναι το δίκτυο. Το πρόγραμμα μπορεί να παρέχει γραφικές απεικονίσεις της σχετικής κίνησης του δικτύου.
- Τα πακέτα μπορούν να εξετασθούν λεπτομερώς. Είναι δυνατόν να γίνει σύλληψη συγκεκριμένων πακέτων, ώστε να διαγνωσθεί και να αντιμετωπιστεί ένα πρόβλημα.

Είναι διαθέσιμο για πολλές πλατφόρμες και εμπορικές και open-source. Κάποιοι απλοί κώδικες είναι στην πραγματικότητα πολύ εύκολο να υλοποιηθούν σε C ή Perl, χρησιμοποιώντας μια γραμμή εντολών και μεταφέρουν τα καταγεγραμμένα δεδομένα στην οθόνη. Τα πιο πολύπλοκα projects χρησιμοποιούν ένα GUI, στατιστικά γραφήματα κυκλοφορίας και προσφέρουν αρκετές επιλογές διαμόρφωσης.

Σε ένα LAN υπάρχουν χιλιάδες πακέτα που ανταλλάσσονται με πολλαπλές μηχανές κάθε λεπτό, και η προσφορά για κάθε εισβολέα είναι άφθονη. Οτιδήποτε που διαβιβάζεται στο plaintext μέσω του δικτύου θα είναι ευάλωτο (όπως κωδικοί πρόσβασης, ιστοσελίδες, ερωτήσεις βάσεων δεδομένων και μηνύματα). Ένα sniffer μπορεί εύκολα να προσαρμοστεί για να συλλάβει συγκεκριμένα πακέτα κυκλοφορίας όπως κρυπτογραφημένες πληροφορίες κωδικών πρόσβασης ή e-mail. Μόλις ξεκινήσει η κίνηση των πακέτων, αυτά καταγράφονται, και έτσι οι εισβολείς μπορούν να εξαγάγουν γρήγορα τις πληροφορίες που χρειάζονται (κωδικούς πρόσβασης, κείμενα μηνυμάτων) χωρίς να αντιληφθούν οι χρήστες ότι ήταν σε κίνδυνο καθώς τα sniffers δεν προκαλούν ζημία ή η διαταραχή στο περιβάλλον του δικτύου.

7.7 DoS Attack (DDoS Attack)

Μια denial-of-service επίθεση (DoS attack[24]) ή distributed denial-of-service attack (DDoS attack) είναι μια προσπάθεια να αποτραπεί η χρήση ενός συστήματος στους χρήστες που προορίζεται, προσωρινά ή επ'αόριστον. Οι δράστες των επιθέσεων DoS συνήθως έχουν ως στόχο τοποθεσίες ή οι υπηρεσίες που φιλοξενούνται σε υψηλού προφίλ web servers όπως οι τράπεζες, πύλες πληρωμής με πιστωτική κάρτα, ακόμα και διακομιστές ονομάτων root. Σε αυτές τις επιθέσεις δεν γίνονται προσπάθειες παραβίασης ή κλοπής στοιχείων.

[23] http://www.symantec.com/connect/articles/sniffers-what-they-are-and-how-protect-yourself

[24] http://www.us-cert.gov/cas/tips/ST04-015.html

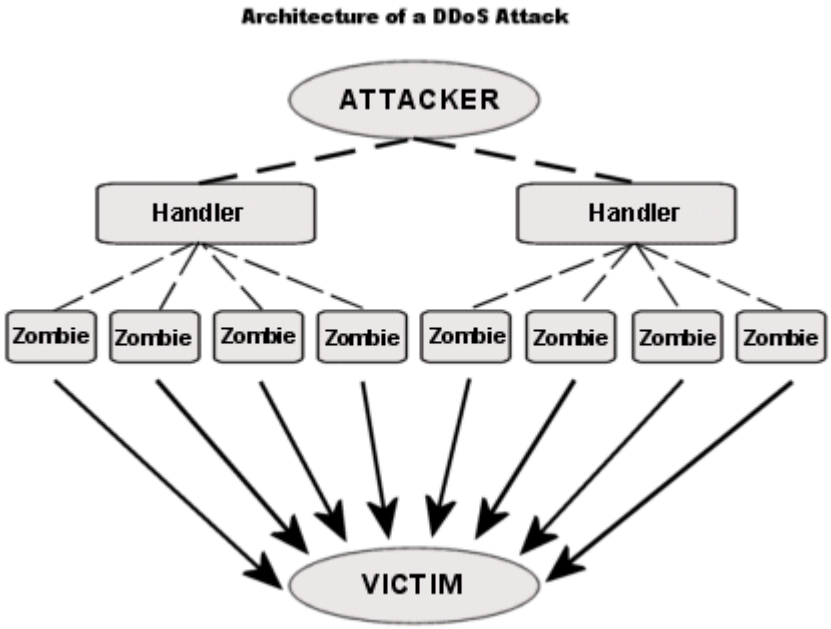
Μια κοινή μέθοδος DoS επίθεσης, είναι να υπερφορτώνει το σύστημα με εξωτερικές αιτήσεις επικοινωνίας, με αποτέλεσμα να μην μπορεί να ανταποκριθεί στις αιτήσεις τον κανονικών χρηστών ή να ανταποκρίνεται πάρα πολύ αργά, τόσο ώστε να είναι αποδοτικά μη διαθέσιμο.

Υπάρχουν δύο γενικές μορφές επιθέσεων DoS:

- ✓ Επιθέσεις που προκαλούν κατάρρευση του συστήματος
- ✓ Επιθέσεις που «πλημμυρίζουν» το σύστημα (αργή ανταπόκριση)

Μια DoS attack μπορεί να πραγματοποιηθεί με πολλούς τρόπους. Οι πέντε πιο βασικές είναι οι εξής:

1. κατανάλωση των υπολογιστικών πόρων (χώρο στο δίσκο, χρόνο επεξεργασίας, εύρος ζώνης)
2. παρεμβολή πληροφοριών ρύθμισης (πληροφορίες δρομολόγησης)
3. παρεμβολή πληροφοριών για την κατάσταση (όπως reset σε TCP sessions)
4. διατάραξη των φυσικών στοιχείων του δικτύου
5. παρεμπόδιση των μέσων επικοινωνίας χρηστών και θύματος



Εικόνα 30: Η αρχιτεκτονική μιας DDoS επίθεσης

7.8 Cross-Site Request Forgery (CSRF)

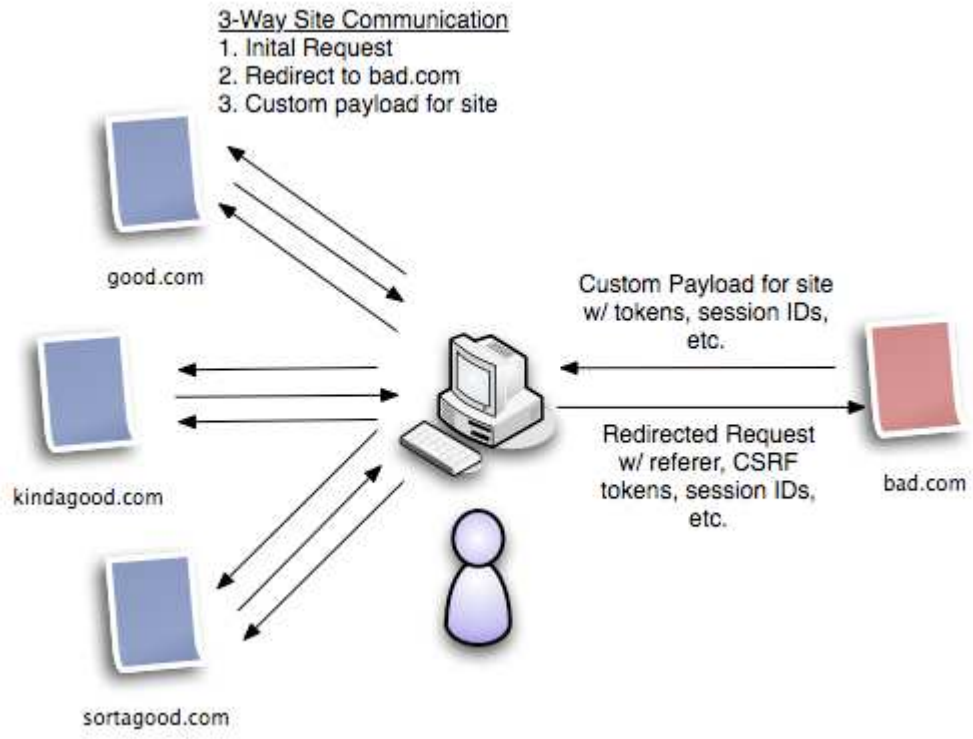
Μια επίθεση Cross Site Request Forgery (CSRF [25]), εκμεταλλεύεται τους ανυποψίαστους χρήστες να εκτελούν κακόβουλες δραστηριότητες (εν αγνοία τους), εφόσον έχουν επικυρωθεί σαν χρήστες. Στην χειρότερη περίπτωση, το θύμα θα μπορούσε να είναι ο διαχειριστής, με αποτέλεσμα να τεθεί σε κίνδυνο όλη η διαδικτυακή εφαρμογή.

[25] http://alko.web.id/blog/other/cross-site-request-forgery.html

Ας δούμε μερικές από τις πιο κοινές επιθέσεις που επιτυγχάνονται με Cross-Site Request Forgery:

- ✓ μπορεί να εξαναγκάσει τον χρήστη-θύμα να δημοσιεύσει ένα υβριστικό σχόλιο ή ένα κακόβουλο link στο blog της ιστοσελίδας μας
- ✓ μπορεί να αλλάξει κωδικούς, e-mail, διαπιστευτήρια του login, ουσιαστική τελική πρόσβαση (αν το θύμα είναι ο διαχειριστής)
- ✓ να υποβάλλει το e-mail του θύματος για να κάνει sign up σε άλλο site
- ✓ να κάνει μια αγορά και να χρησιμοποιήσει τη διεύθυνση των χάκερς

Οι δυνατότητες των CSRF επιθέσεων είναι τόσο ισχυρές, που αναγκάζουν τις τράπεζες, τους οικονομικούς brokers, οι υπηρεσίες bill pay και βασικά οποιοδήποτε θεσμικό όργανο που δίνει πιστοποιήσεις χρήστη για χρήματα, χρειάζεται να προσεγγίζουν κάθε μέρα με ιδιαίτερη προσοχή και εποπτεία. Σε ένα blog post, η SECCOM Labs απέδειξε πόσο εύκολα μια επίθεση CSRF θα μπορούσε να προκαλέσει καθεστώς τραπεζών.



Εικόνα 31: Dynamic CSRF Attack

Οι επιθέσεις CSRF στοχεύουν στις λειτουργίες που μπορούν να προκαλέσουν αλλαγές στον server, όπως επίσης και για να έχουν πρόσβαση σε ευαίσθητα δεδομένα. Μερικές φορές, είναι δυνατόν από μόνο του κάποιο ευάλωτο site να αποθηκεύσει μια τέτοια επίθεση. Οι εν λόγω ευπάθειες ονομάζονται CSRF flaws. Πως μπορεί να συμβεί αυτό; Επιτυγχάνεται με μια απλή αποθήκευση ενός IMG ή IFRAME tag σε κάποιο πεδίο που δέχεται HTML ή με μια πιο σύνθετη Cross-Site Scripting. Αν μπορέσει να αποθηκεύσει μια CSRF επίθεση στον ιστότοπο, τότε η σοβαρότητα της επίθεσης αυξάνεται. Αυτό συμβαίνει γιατί είναι πιο πιθανό το θύμα να επισκεφτεί κάποια σελίδα στο διαδίκτυο, στην οποία πολύ πιθανόν να έχει ήδη πιστοποιηθεί σαν χρήστης.

Η επίθεση CSRF είναι γνωστή επίσης από μια σειρά ονομάτων όπως XSRF, "Sea Surf", Session Riding, Cross-Site Reference Forgery, Hostile Linking. Η Microsoft αναφέρεται σε αυτό το είδος της επίθεσης ως One-Click επίθεση.

Αφού εξετάσαμε τα τρωτά σημεία στην ιστοσελίδα μας, θα μελετήσουμε τους τρόπους με τους οποίους μπορούμε να την θωρακίσουμε από τους κακόβουλους χρήστες.

ΚΕΦΑΛΑΙΟ 8

ΠΡΩΤΟΚΟΛΛΑ ΑΣΦΑΛΕΙΑΣ ΔΙΚΤΥΟΥ

Σε μια ηλεκτρονική επικοινωνία, η εμπιστοσύνη μεταξύ των συναλλασσόμενων μερών είναι πολύ σημαντική. Η τεχνολογία παρέχει προηγμένες λύσεις στο θέμα αυτό. Στην περίπτωση του ηλεκτρονικού καταστήματος, όπου οι συναλλαγές πραγματοποιούνται μέσω ανοιχτών δικτύων, η ασφάλεια είναι επιτακτική ανάγκη.

Για τη ασφάλεια σε επίπεδο δικτύου υπάρχουν διάφορες τεχνικές και μηχανισμοί που επιτρέπουν τη υγιείς ηλεκτρονικές συναλλαγές, τις οποίες θα εξετάσουμε σε αυτό το κεφάλαιο.

8.1 Secure HTTP (S-HTTP)

Το HTTPS (Secure HTTP)[26] χρησιμοποιείται στην επιστήμη των υπολογιστών για να δηλώσει μία ασφαλή http σύνδεση. Ένας σύνδεσμος (URL) που αρχίζει με το πρόθεμα https υποδηλώνει ότι θα χρησιμοποιηθεί κανονικά το πρωτόκολλο HTTP, αλλά η σύνδεση θα γίνει σε διαφορετική πόρτα (443 αντί 80) και τα δεδομένα θα ανταλλάσσονται κρυπτογραφημένα.

Το σύστημα αυτό σχεδιάστηκε αρχικά από την εταιρία Netscape Communications Corporation για να χρησιμοποιηθεί σε sites όπου απαιτείται αυθεντικοποίηση χρηστών και κρυπτογραφημένη επικοινωνία. Σήμερα χρησιμοποιείται ευρέως στο διαδίκτυο όπου χρειάζεται αυξημένη ασφάλεια διότι διακινούνται ευαίσθητες πληροφορίες (π.χ. αριθμοί πιστωτικών καρτών, passwords κοκ).

Το HTTPS δεν είναι ξεχωριστό πρωτόκολλο όπως μερικοί νομίζουν, αλλά αναφέρεται στον συνδυασμό του απλού HTTP πρωτοκόλλου και των δυνατοτήτων κρυπτογράφησης που παρέχει το πρωτόκολλο Secure Sockets Layer (SSL). Η κρυπτογράφηση που χρησιμοποιείται διασφαλίζει ότι τα κρυπτογραφημένα δεδομένα δεν θα μπορούν να υποκλαπούν από άλλους κακόβουλους χρήστες ή από επιθέσεις man-in-the-middle.

Για να χρησιμοποιηθεί το HTTPS σε έναν server, θα πρέπει ο διαχειριστής του να εκδώσει ένα πιστοποιητικό δημοσίου κλειδιού. Σε servers που χρησιμοποιούν το λειτουργικό σύστημα UNIX αυτό μπορεί να γίνει μέσω του προγράμματος OpenSSL. Στην συνέχεια το πιστοποιητικό αυτό θα πρέπει να υπογραφεί από μία αρχή πιστοποίησης (certificate authority), η οποία πιστοποιεί ότι ο εκδότης του πιστοποιητικού είναι νομότυπος και ότι το πιστοποιητικό είναι έγκυρο. Με τον τρόπο αυτό οι χρήστες μπορούν να δουν την υπογραφή της αρχής πιστοποίησης και να βεβαιωθούν ότι το πιστοποιητικό είναι έγκυρο και ότι κανένας κακόβουλος χρήστης δεν το έχει πλαστογραφήσει.

[26] http://el.wikipedia.org/wiki/HTTPS

Όπως αναφέρθηκε προηγουμένως, το HTTPS χρησιμοποιείται κυρίως όταν απαιτείται μεταφορά ευαίσθητων προσωπικών δεδομένων. Πολλοί χρήστες πιστωτικών καρτών θεωρούν ότι το HTTPS προστατεύει ολοκληρωτικά τον αριθμό της πιστωτικής τους κάρτας από κατάχρηση. Αυτό όμως δεν ισχύει γιατί το HTTPS χρησιμοποιεί την κρυπτογράφηση για να μεταδώσει τον αριθμό από τον υπολογιστή του πελάτη προς τον server. Η μετάδοση είναι ασφαλής και τα δεδομένα φτάνουν στον server χωρίς κανείς να μπορέσει να τα υποκλέψει. Παρόλα αυτά υπάρχει το ενδεχόμενο διάφοροι χάκερ να έχουν επιτεθεί στον server και από εκεί να έχουν υποκλέψει τα ευαίσθητα προσωπικά δεδομένα.

8.2 Ασφαλείς Ηλεκτρονικές Συναλλαγές (Secure Electronics Transaction-SET)

Το Secure Electronics Transaction (SET)[27] ήταν ένα πρότυπο πρωτόκολλο για τη διασφάλιση των συναλλαγών με πιστωτικές κάρτες μέσω του διαδικτύου. Το SET δεν ήταν κάποιο σύστημα πληρωμών, αλλά είναι ένα σύνολο από πρωτόκολλα ασφαλείας που επιτρέπουν στους χρήστες να μπορούν να χρησιμοποιούν, ως τρόπο πληρωμής για τις διαδικτυακές τους αγορές, τις πιστωτικές τους κάρτες με ασφάλεια. Το πρωτόκολλο αυτό αναπτύχθηκε από την MasterCard και τη Visa.

Η διαδικασία περιλαμβάνει ένα αριθμό ελέγχων ασφαλείας που πραγματοποιούνται με τη χρήση ψηφιακών πιστοποιητικών, τα οποία χορηγούνται στους εμπλεκόμενους αγοραστές, εμπόρους και τράπεζες. Στηρίζεται στην κρυπτογραφία και τα ψηφιακά πιστοποιητικά για να εξασφαλίσει την εμπιστευτικότητα και την ασφάλεια μηνυμάτων. Είναι το μόνο πρωτόκολλο συναλλαγής διαδικτύου που παρέχει την ασφάλεια μέσω επικύρωσης. Διώχνει τον κίνδυνο της αλλοίωσης των συναλλασσόμενων πληροφοριών, αφού τις κρυπτογραφεί και με χρήση των ψηφιακών πιστοποιητικών ελέγχει την ταυτότητα των εμπλεκόμενων της συναλλαγής.

8.3 Ψηφιακές Υπογραφές – Digital Signatures

Η Ψηφιακή Υπογραφή[28] είναι ένα μαθηματικό σύστημα που χρησιμοποιείται για την απόδειξη της γνησιότητας ενός ψηφιακού μηνύματος ή εγγράφου. Μια έγκυρη

[27] http://en.wikipedia.org/wiki/Secure_Electronic_Transaction
[28] http://el.wikipedia.org/wiki/Ψηφιακή_υπογραφή

ψηφιακή υπογραφή δίνει στον παραλήπτη την πιστοποίηση ότι το μήνυμα που δημιουργήθηκε ανήκει στον αποστολέα που το υπέγραψε ψηφιακά και ότι δεν αλλοιώθηκε-παραποιήθηκε κατά την μεταφορά.

Οι ψηφιακές υπογραφές [29] χρησιμοποιούν την κρυπτογραφία του δημόσιου κλειδιού. Ο χρήστης διαθέτει ένα δημόσιο κλειδί και ένα ιδιωτικό, τα οποία έχουν κάποιο μαθηματικό συσχετισμό. Ακόμη κι αν κάποιος γνωρίζει κάποιο από τα δυο κλειδιά, είναι αδύνατον να βρει το άλλο. Το ιδιωτικό κλειδί χρησιμοποιείται από τον αποστολέα για την δημιουργία της ψηφιακής υπογραφής του, ενώ το δημόσιο χρησιμοποιείται από τον παραλήπτη για την επαλήθευση της ψηφιακής υπογραφής του αποστολέα.

Στη διαδικασία της δημιουργίας και επαλήθευσης της υπογραφής εμπλέκεται και η έννοια της συνάρτησης κατακερματισμού (ή κατατεμαχισμού -one way hash). Με την εφαρμογή της συνάρτησης κατακερματισμού, από ένα μήνυμα ανεξαρτήτου του μεγέθους του, παράγεται η «σύνοψή του», η οποία είναι μία σειρά από bits συγκεκριμένου μεγέθους (π.χ. 128 ή 160 bits). Η σύνοψη του μηνύματος (fingerprint ή message digest) είναι μία ψηφιακή αναπαράσταση του μηνύματος, είναι μοναδική για το μήνυμα και το αντιπροσωπεύει.

Η συνάρτηση κατακερματισμού είναι μονόδρομη, διότι από την σύνοψη που δημιουργεί, είναι υπολογιστικά αδύνατον κάποιος να εξάγει το αρχικό μήνυμα. Η πιθανότητα δύο μηνύματα να έχουν την ίδια σύνοψη είναι εξαιρετικά μικρή. Αυτό σημαίνει ότι αν το μήνυμα του αποστολέα έχει κάποια συγκεκριμένη σύνοψη και το μήνυμα που λάβει ο παραλήπτης (χρησιμοποιώντας την ίδια συνάρτηση κατακερματισμού) παράγει διαφορετική σύνοψη, τότε το μήνυμα κατά την μετάδοσή του έχει αλλοιωθεί (μη ακεραιότητα). Οποιαδήποτε αλλαγή σε ένα μήνυμα συνεπάγεται και τη δημιουργία διαφορετικής σύνοψης.

Θεωρώντας ότι ο αποστολέας έχει ένα συγκεκριμένο ζευγάρι κλειδιών και το ιδιωτικό του κλειδί είναι στην πλήρη κατοχή του, τότε το γεγονός ότι ο αποστολέας χρησιμοποιεί το ιδιωτικό του κλειδί για να κρυπτογραφήσει το μήνυμα, πιστοποιεί στον παραλήπτη που το αποκρυπτογραφεί με το αντίστοιχο δημόσιο κλειδί (του αποστολέα) την ταυτότητα του αποστολέα (αυθεντικότητα). Η ψηφιακή υπογραφή είναι ένας τρόπος αυθεντικοποίησης του αποστολέα του μηνύματος.

Μία ψηφιακή υπογραφή μπορεί να πλαστογραφηθεί εάν ο δικαιούχος του ιδιωτικού κλειδιού δεν το έχει υπό τον πλήρη έλεγχό του (π.χ. χάσει το μέσο στο οποίο έχει αποθηκευτεί το ιδιωτικό κλειδί).

[29] http://www.eett.gr/opencms/opencms/EETT/Electronic_Communications/DigitalSignatures/IntroEsign.html

8.4 Ψηφιακά Πιστοποιητικά (Digital Certificates)

Το Ψηφιακό πιστοποιητικό[30] είναι ένα ηλεκτρονικό έγγραφο που χρησιμοποιείται για την αναγνώριση μίας οντότητας (φυσικό πρόσωπο, εξυπηρετητής, οργανισμός κοκ) και την ανάκτηση του δημοσίου κλειδιού αυτής.

Η έκδοση ενός ψηφιακού πιστοποιητικού γίνεται μετά από αίτηση του ενδιαφερομένου σε μία Αρχή Πιστοποίησης. Η Αρχή Πιστοποίησης επιβεβαιώνει την ταυτότητα του αιτούντος και εκδίδει το πιστοποιητικό, το οποίο συνοπτικά περιλαμβάνει τα εξής στοιχεία:

- Το ονοματεπώνυμο και διάφορες άλλες πληροφορίες σχετικά με τον κάτοχο του πιστοποιητικού.
- Το δημόσιο κλειδί του κατόχου του πιστοποιητικού.
- Την ημερομηνία λήξης του πιστοποιητικού.
- Το όνομα και την ψηφιακή υπογραφή της Αρχής Πιστοποίησης που το εξέδωσε.

Τα ψηφιακά πιστοποιητικά χρησιμοποιούνται ευρέως για διάφορες κρυπτογραφημένες ηλεκτρονικές συναλλαγές μέσω του διαδικτύου. Παραδείγματα τέτοιων συναλλαγών είναι: Σύνοδοι με βάση το πρωτόκολλο SSL (Client/Server SSL Certificates), κρυπτογραφημένο και υπογεγραμμένο ηλεκτρονικό ταχυδρομείο (S/MIME Certificates), υπογραφή αντικειμένων (Object-signing Certificates) κοκ. Το πιο διαδεδομένο πρότυπο ψηφιακών πιστοποιητικών είναι το X.509.

8.5 Επίπεδο Ασφαλών Συνδέσεων- Secure Sockets Layer (SSL)

8.5.1 Γενικά για το πρωτόκολλο SSL

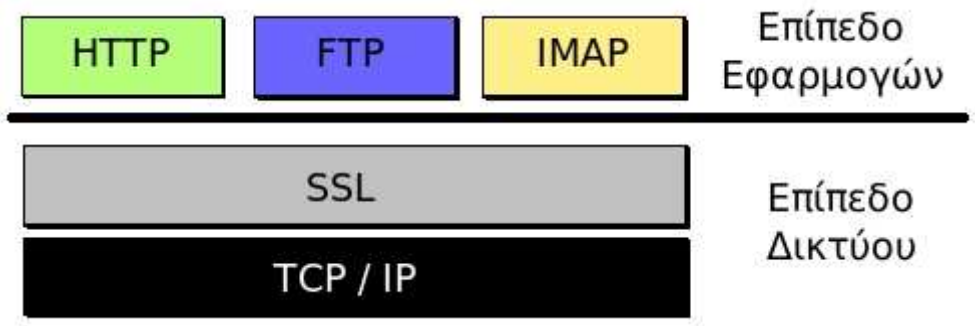
Το πρωτόκολλο SSL (Secure Sockets Layer)[31] αναπτύχθηκε από την εταιρεία Netscape και σχεδιάστηκε για να παρέχει ασφάλεια κατά την μετάδοση ευαίσθητων δεδομένων στο διαδίκτυο. Η έκδοση 3.0 του πρωτοκόλλου κυκλοφόρησε από την Netscape το 1996 και αποτέλεσε την βάση για την μετέπειτα ανάπτυξη του πρωτοκόλλου TLS (Transport Layer Security), το οποίο πλέον τείνει να

[30] http://el.wikipedia.org/wiki/Ψηφιακό_πιστοποιητικό
[31] http://el.wikipedia.org/wiki/SSL

αντικαταστήσει το SSL. Τα δύο αυτά πρωτόκολλα χρησιμοποιούνται ευρέως για ηλεκτρονικές αγορές και χρηματικές συναλλαγές μέσω του διαδικτύου.

Το SSL χρησιμοποιεί μεθόδους κρυπτογράφησης των δεδομένων που ανταλλάσσονται μεταξύ δύο συσκευών (συνηθέστερα Ηλεκτρονικών Υπολογιστών) εγκαθιδρύοντας μία ασφαλή σύνδεση μεταξύ τους μέσω του διαδικτύου. Το πρωτόκολλο αυτό χρησιμοποιεί το TCP/IP για τη μεταφορά των δεδομένων και είναι ανεξάρτητο από την εφαρμογή που χρησιμοποιεί ο τελικός χρήστης. Για τον λόγο αυτό μπορεί να παρέχει υπηρεσίες ασφαλούς μετάδοσης πληροφοριών σε πρωτόκολλα ανώτερου επιπέδου όπως για παράδειγμα το HTTP, το FTP, το telnet κοκ.

Η μετάδοση πληροφοριών μέσω του διαδικτύου γίνεται ως επί το πλείστον χρησιμοποιώντας τα πρωτόκολλα TCP/IP (Transfer Control Protocol / Internet Protocol). Το SSL λειτουργεί πριν το TCP/IP και μετά τις εφαρμογές υψηλού επιπέδου, όπως είναι για παράδειγμα το HTTP (προβολή ιστοσελίδων), το FTP (μεταφορά αρχείων) και το IMAP (email). Άρα λοιπόν αυτό που ουσιαστικά κάνει το SSL είναι να παίρνει τις πληροφορίες από τις εφαρμογές υψηλότερων επιπέδων, να τις κρυπτογραφεί και στην συνέχεια να τις μεταδίδει στο Internet προς τον Η/Υ που βρίσκεται στην απέναντι πλευρά και τις ζήτησε.



Εικόνα 32: Το SSL λειτουργεί πριν το TCP/IP και μετά τις εφαρμογές υψηλού επιπέδου

Το SSL προσφέρει συνοπτικά τις ακόλουθες υπηρεσίες:

- Πιστοποίηση του server από τον client.
- Πιστοποίηση του client από τον server.
- Εγκαθίδρυση ασφαλούς κρυπτογραφημένου διαύλου επικοινωνίας μεταξύ των δύο μερών.

Οι κρυπτογραφικοί αλγόριθμοι που υποστηρίζονται από το πρωτόκολλο είναι οι εξής: DES - Data Encryption Standard, DSA - Digital Signature Algorithm, KEA - Key Exchange Algorithm, MD5 - Message Digest, RC2/RC4, RSA, SHA-1 - Secure Hash Algorithm, SKIPJACK, Triple-DES.

Το πρωτόκολλο SSL χρησιμοποιεί έναν συνδυασμό της κρυπτογράφησης δημοσίου και συμμετρικού κλειδιού. Η κρυπτογράφηση συμμετρικού κλειδιού είναι πολύ πιο γρήγορη και αποδοτική σε σχέση με την κρυπτογράφηση δημοσίου κλειδιού, παρ' όλα αυτά όμως η δεύτερη προσφέρει καλύτερες τεχνικές πιστοποίησης. Κάθε

σύνδεση SSL ξεκινά πάντα με την ανταλλαγή μηνυμάτων από τον server και τον client έως ότου επιτευχθεί η ασφαλής σύνδεση, πράγμα που ονομάζεται χειραψία (handshake). Η χειραψία επιτρέπει στον server να αποδείξει την ταυτότητά του στον client χρησιμοποιώντας τεχνικές κρυπτογράφησης δημοσίου κλειδιού και στην συνέχεια επιτρέπει στον client και τον server να συνεργαστούν για την δημιουργία ενός συμμετρικού κλειδιού που θα χρησιμοποιηθεί στην γρήγορη κρυπτογράφηση και αποκρυπτογράφηση των δεδομένων που ανταλλάσσονται μεταξύ τους. Προαιρετικά η χειραψία επιτρέπει επίσης στον client να αποδείξει την ταυτότητά του στον server.

Αναλυτικότερα, η διαδικασία χειραψίας έχει ως εξής:

1. Αρχικά ο client στέλνει στον server την έκδοση του SSL που χρησιμοποιεί, τον επιθυμητό αλγόριθμο κρυπτογράφησης, μερικά δεδομένα που έχουν παραχθεί τυχαία και οποιαδήποτε άλλη πληροφορία χρειάζεται ο server για να ξεκινήσει μία σύνδεση SSL.
2. Ο server απαντά στέλνοντας παρόμοιες πληροφορίες με προηγουμένως συμπεριλαμβανομένου όμως και του ψηφιακού πιστοποιητικού του, το οποίο τον πιστοποιεί στον client. Προαιρετικά μπορεί να ζητήσει και το ψηφιακό πιστοποιητικό του client.
3. Ο client λαμβάνει το ψηφιακό πιστοποιητικό του server και το χρησιμοποιεί για να τον πιστοποιήσει. Εάν η πιστοποίηση αυτή δεν καταστεί δυνατή, τότε ο χρήστης ενημερώνεται με ένα μήνυμα σφάλματος και η σύνδεση SSL ακυρώνεται. Εάν η πιστοποίηση του server γίνει χωρίς προβλήματα, τότε η διαδικασία της χειραψίας συνεχίζεται στο επόμενο βήμα.
4. Ο client συνεργάζεται με τον server και αποφασίζουν τον αλγόριθμο κρυπτογράφησης που θα χρησιμοποιηθεί στην ασφαλή σύνδεση SSL. Επίσης ο client δημιουργεί το συμμετρικό κλειδί που θα χρησιμοποιηθεί στον αλγόριθμο κρυπτογράφησης και το στέλνει στον server κρυπτογραφημένο, χρησιμοποιώντας την τεχνική κρυπτογράφησης δημοσίου κλειδιού. Δηλαδή χρησιμοποιεί το δημόσιο κλειδί του server που αναγράφεται πάνω στο ψηφιακό του πιστοποιητικό για να κρυπτογραφήσει το συμμετρικό κλειδί και να του το στείλει. Στην συνέχεια ο server χρησιμοποιώντας το ιδιωτικό του κλειδί μπορεί να αποκρυπτογραφήσει το μήνυμα και να αποκτήσει το συμμετρικό κλειδί που θα χρησιμοποιηθεί για την σύνδεση.
5. Ο client στέλνει ένα μήνυμα στον server ενημερώνοντάς τον ότι είναι έτοιμος να ξεκινήσει την κρυπτογραφημένη σύνδεση.
6. Ο server στέλνει ένα μήνυμα στον client ενημερώνοντάς τον ότι και αυτός είναι έτοιμος να ξεκινήσει την κρυπτογραφημένη σύνδεση.
7. Από εδώ και πέρα η χειραψία έχει ολοκληρωθεί και τα μηνύματα που ανταλλάσσουν τα δύο μηχανήματα (client - server) είναι κρυπτογραφημένα.

Η διαδικασία της χειραψίας φαίνεται πιο παραστατικά στο σχήμα που ακολουθεί.

Εικόνα 33: Η διαδικασία της χειραψίας των δύο συσκευών σύμφωνα με το πρωτόκολλο SSL

Η χρήση του πρωτοκόλλου SSL αυξάνει τα διακινούμενα πακέτα μεταξύ των δύο μηχανών και καθυστερεί την μετάδοση των πληροφοριών επειδή χρησιμοποιεί μεθόδους κρυπτογράφησης και αποκρυπτογράφησης.

Ειδικότερα οι διάφορες καθυστερήσεις εντοπίζονται στα εξής σημεία:

- Στην αρχική διαδικασία χειραψίας όπου κανονίζονται οι λεπτομέρειες της σύνδεσης και ανταλλάσσονται τα κλειδιά της συνόδου.
- Στην διαδικασία κρυπτογράφησης και αποκρυπτογράφησης που γίνεται στους δύο υπολογιστές με αποτέλεσμα να δαπανώνται υπολογιστικοί πόροι και χρόνος.
- Στην καθυστέρηση μετάδοσης των κρυπτογραφημένων δεδομένων αφού αυτά αποτελούνται από περισσότερα bytes σε σχέση με την αρχική μη κρυπτογραφημένη πληροφορία.

Λόγω αυτών των επιβαρύνσεων που εισάγει το πρωτόκολλο SSL, χρησιμοποιείται πλέον μονάχα σε περιπτώσεις όπου πραγματικά χρειάζεται ασφαλής σύνδεση (π.χ μετάδοση κωδικών χρήστη ή αριθμών πιστωτικών καρτών μέσω του διαδικτύου) και όχι σε περιπτώσεις απλής επίσκεψης σε μία ιστοσελίδα.

8.5.2 Αντοχή του SSL σε γνωστές επιθέσεις

Dictionary Attack

Αυτό το είδος της επίθεσης λειτουργεί όταν ένα μέρος του μη κρυπτογραφημένου κειμένου είναι στην κατοχή του ανέντιμων προσώπων. Το μέρος αυτό κρυπτογραφείται με χρήση κάθε πιθανού κλειδιού και έπειτα ερευνάται ολόκληρο το κρυπτογραφημένο μήνυμα μέχρι να βρεθεί κομμάτι του που να ταιριάζει με κάποιο από τα προϋπολογισμένα. Σε περίπτωση που η έρευνα έχει επιτυχία, τότε το κλειδί που χρησιμοποιήθηκε για την κρυπτογράφηση ολόκληρου του μηνύματος έχει βρεθεί.

Το SSL δεν απειλείται από αυτήν την επίθεση αφού τα κλειδιά των αλγορίθμων του είναι πολύ μεγάλα των 128 bit. Ακόμα και οι αλγόριθμοι σε εξαγόμενα προϊόντα, υποστηρίζουν 128 bit κλειδιά και παρ' όλο που τα 88 bit αυτών μεταδίδονται ανασφάλιστα, ο υπολογισμός 240 διαφορετικών ακολουθιών κάνει την επίθεση αδύνατο να επιτύχει.

Brute Force Attack

Η επίθεση αυτή πραγματοποιείται με την χρήση όλων των πιθανών κλειδιών για την αποκρυπτογράφηση των μηνυμάτων. Όσο πιο μεγάλα σε μήκος είναι τα χρησιμοποιούμενα κλειδιά, τόσο πιο πολλά είναι τα πιθανά κλειδιά. Τέτοια επίθεση σε αλγορίθμους που χρησιμοποιούν κλειδιά των 128 bits είναι τελείως ανούσια. Μόνο ο DES56 bit cipher είναι ευαίσθητος σε αυτήν την επίθεση, αλλά η χρήση του δεν συνιστάται.

Replay Attack

Όταν ένας τρίτος καταγράφει την ανταλλαγή μηνυμάτων μεταξύ client και server και προσπαθεί να ξανά χρησιμοποιήσει τα μηνύματα του client για να αποκτήσει πρόσβαση στον server, έχουμε την επίθεση replay attack. Όμως το SSL κάνει χρήση του connection-id, το οποίο παράγεται από τον server με τυχαίο τρόπο και διαφέρει για κάθε σύνδεση. Έτσι δεν είναι δυνατόν πότε να υπάρχουν δυο ίδια connection-id και το σύνολο των είδη χρησιμοποιημένων μηνυμάτων δεν γίνονται δεκτά από τον server. Το connection-id έχει μέγεθος 128 bit για πρόσθετη ασφάλεια.

Man-In-The-Middle-Attack

Η επίθεση Man-In-The-Middle συμβαίνει όταν ένας τρίτος είναι σε θέση να παρεμβάλλεται στην επικοινωνία μεταξύ του server και του client. Αφού επεξεργαστεί τα μηνύματα του client και τροποποιήσει όπως αυτός επιθυμεί, τα προωθεί στον server. Ομοίως πράττει για τα μηνύματα που προέρχονται από τον server. Δηλαδή, προσποιείται στον client ότι είναι ο server και αντίστροφα.

Το SSL υποχρεώνει τον server να αποδεικνύει την ταυτότητα του με την χρήση έγκυρου πιστοποιητικού του οποίου η τροποποίηση είναι αδύνατον. Μην ξεχνάμε την δυνατότητα επικοινωνίας των κλειδιών υπογεγραμμένα.

8.6 Ηλεκτρονική Ανταλλαγή Δεδομένων-Electronic Data Interchange (EDI)

Η ηλεκτρονική ανταλλαγή δεδομένων (EDI) αποτελεί τεχνολογία που υποστηρίζει την πραγματοποίηση της μεταβίβασης δεδομένων μεταξύ επιχειρήσεων ελαχιστοποιώντας τα λάθη, βελτιώνοντας την ροή των χρημάτων και την ποιότητα των παρεχόμενων υπηρεσιών, μειώνοντας τα αποθέματα και επιταχύνοντας καθολικά την διαδικασία των συναλλαγών. Το EDI συχνά αναφέρεται και ως "εμπόριο χωρίς έγγραφα" γιατί συνδυάζει τις δυνατότητες των υπολογιστών και των τηλεπικοινωνιακών δικτύων με στόχο την αντικατάσταση των έντυπων παραστατικών / εγγράφων από στις εμπορικές συναλλαγές. Η τεχνική EDI αποτελεί αποδοτικό μέσο επικοινωνίας μεταξύ συνεργαζομένων οργανισμών[32].

Η χρήση του EDI, προϋποθέτει και από τις δύο συναλλασσόμενες πλευρές, την εγκατάσταση κάποιου λογισμικού. Επίσης κάνει αναφορά στην «οικογένεια προτύπων», συμπεριλαμβανομένης της σειράς X12.

Για να διασφαλιστεί ότι τα EDI έγγραφα μεταβαίνουν με ασφάλεια, το διαδίκτυο εκτός από VAN παρόχους, χρησιμοποιεί και τα δικά του πρωτόκολλα ασφαλείας. Τα πιο δημοφιλή από αυτά είναι τα: File Transfer Protocol Secure (FTPS), Hyper Text Transport Protocol Secure (HTTPS) και AS2[33].

[32] http://www.logistics.tuc.gr/XEXO%20Technical-material/contents_xexo/OPA%5CE-COMMERSE%5Cfiles%5C7.0%20%CE%9A%CE%B5%CF%86%CE%AC%CE%BB%CE%B1%CE%B9%CE%BF%207.htm
[33] http://en.wikipedia.org/wiki/Electronic_Data_Interchange

ΚΕΦΑΛΑΙΟ 9

ΡΥΘΜΙΣΕΙΣ ΑΣΦΑΛΕΙΑΣ ΣΥΣΤΗΜΑΤΟΣ

Η ασφάλεια στο διαδίκτυο είναι ένα δύσκολο εγχείρημα αλλά όχι ακατόρθωτο, καθώς εξελίσσεται τόσο ραγδαία, όσο και οι τρόποι επιθέσεις. Όλες οι μέθοδοι ασφαλείας βελτιώνονται, αλλά συγκεκριμένος τρόπος να μας βγάλει από αυτή τη μάχη με τους «κακούς» του διαδικτύου δεν υπάρχει. Ωστόσο υπάρχουν κάποιες ρυθμίσεις-αρχές που μπορούν να μας βοηθήσουν να παραμείνουμε ασφαλής.

9.1 SQL Injections στο Joomla!

Ένα πολύ σημαντικό μέρος του Joomla! CMS είναι η βάση δεδομένων, καθώς εκεί βρίσκονται το περιεχόμενο, τα στοιχεία των χρηστών, οι ρυθμίσεις μας και πολλά ακόμη. Το να αποκτήσει πρόσβαση ένας κακόβουλος χρήστης, σε ένα τόσο ζωτικό κομμάτι του site, αυτομάτως σημαίνει πως μπορεί να συγκεντρώσει σημαντικές πληροφορίες (όπως τα ονόματα και τους κωδικούς των χρηστών).

Όταν γίνει μια αίτηση στην ιστοσελίδα μας, στην ουσία γίνεται ένα ερώτημα στη βάση δεδομένων. Η βάση δεν μπορεί να διαχωρίσει τον «καλό» κώδικα από τον «κακό» και θα εκτελέσει το ερώτημα κανονικά. Για να διασφαλίσουμε λοιπόν το ηλεκτρονικό μας κατάστημα και να μην επιτρέψουμε την εκτέλεση ακατάλληλων αιτημάτων θα πρέπει να κάνουμε κάποιες ρυθμίσεις ασφαλείας.

9.1.1 Ρυθμίσεις ασφαλείας στο Joomla!

Όταν δημιουργείται κάποιος πίνακας στο Joomla! το πρόθεμα του είναι “jos_”, και αρκετά security exploits γίνονται με βάση τους πίνακες της βάσης δεδομένων που η ονομασία τους είναι της μορφής “jos_xxxxx”. Πρώτο μέλημά μας λοιπόν είναι να χρησιμοποιούμε διαφορετικό όνομα για τους πίνακες μας, έτσι ώστε να προστατευθούμε από κακόβουλα λογισμικά.

Κατά τη δημιουργία του ιστοτόπου μας θα πρέπει να αλλάζουμε το πρόθεμα στην ονομασία των πινάκων, από τη σελίδα εγκατάστασης όπου συμπληρώνουμε τις ρυθμίσεις της βάσης δεδομένων. Στην περιοχή “advanced settings”, αλλάζουμε το λεκτικό “jos_” σε κάποιο της προτίμησής μας (καλό θα ήταν να είναι της μορφής “xxx_” , τριών γραμμάτων δηλαδή και κάτω παύλας).

Στην περίπτωσή που έχει ήδη γίνει εγκατάσταση του ιστοτόπου και δεν έχει αλλαχθεί η ονομασία των πινάκων, μπορεί και πάλι να γίνει η μετονομασία αφού γίνει εξαγωγή της βάσης. Κατά την εξαγωγή της βάσης δεδομένων, γίνεται η αλλαγή του προθέματος και εκτελείται ένα νέο SQL ερώτημα, το οποίο με τη σειρά του θα δημιουργήσει τους πίνακες από την αρχή με το καινούργιο όνομα. Ο διαχειριστής

μετά με τη σειρά του, θα πρέπει πολύ προσεχτικά να διαγράψει τους πίνακες με το παλιό πρόθεμα και έτσι θα υπάρχουν μόνο οι καινούργιοι.

Και πάλι όμως δεν είναι αρκετό αυτό για να μας σώσει από μια επίθεση. Έτσι λοιπόν, ο διαχειριστής θα πρέπει να γίνει πιο συγκεκριμένος στη διατύπωση των ερωτημάτων SQL που θα διαχειρίζεται το site. Παρακάτω βλέπουμε μεθόδους που θα μπορούσε να ακολουθήσει:

- ✓ η είσοδος του χρήστη θα πρέπει να επικυρώνεται από τους διαχειριστές, δηλαδή να ελέγχεται (πάντα μέσω SQL ερωτημάτων) ο τύπος, το μήκος, η μορφή και το φάσμα, και ποτέ να μην είναι σίγουρος πως κακόβουλα δεδομένα δε θα παραβρίσκονται στα ερωτήματα που θα υποβάλλει κάποιος
- ✓ απευθείας δηλώσεις SQL να μην επιτρέπονται σε καμία περίπτωση
- ✓ να περιορίζονται οι καταχωρίσεις δεδομένων σε τύπους που εκάστοτε πρέπει (όπως για παράδειγμα κάποιος χρήστη σε πεδίο που θα πρέπει να γράψει χαρακτήρες να μην βάλει εικόνα)
- ✓ το μέγεθος δεδομένων εισόδου των χρηστών να μην είναι απεριόριστοι, για να αποτραπεί και τυχόν περίπτωση overflow ή κάποια άλλη τρέλα του συστήματος (δηλαδή εάν το μέγιστο μήκος χαρακτήρων είναι 12 να μην επιτρέπονται παραπάνω)
- ✓ το περιεχόμενο των string μεταβλητών θα πρέπει να δοκιμάζεται και να γίνονται αποδεκτές μόνο αναμενόμενες τιμές. Καταχωρήσεις που περιέχουν δυαδικά δεδομένα, ακολουθίες διαφυγής και χαρακτήρες σχολίων (“”), θα πρέπει να απορρίπτονται
- ✓ εάν σε κάποιο σημείο οι χρήστες θα πρέπει να εισάγουν έναν ακέραιο αριθμό, θα πρέπει να είναι ακέραιος, για παράδειγμα ο κώδικας θα ήταν της μορφής:

```
$sql = 'UPDATE #__mytable SET `id` = '. (int) $int;
```

- ✓ αν τα δεδομένα εισόδου που θα λάβουμε είναι σε μορφή string, καλό θα ήταν να τα αποφύγουμε, γιατί τα strings είναι και η αρχή για script injections. Για παράδειγμα στον παρακάτω κώδικα, χρησιμοποιούνται δύο functions. Η πρώτη διαφεύγει τη μεταβλητή (string) και η δεύτερη της θέτει μέσα σε εισαγωγικά. Επίσης, παρατηρούμε πως η δεύτερη παράμετρος “$db-> quote ()” είναι false.

```
$sql = 'UPDATE #__mytable SET `string` = '. $db-
>quote ($db->getEscaped ($string),false);
```

Αν βγάλουμε αυτή τη συνθήκη ή αν την μετατρέψουμε σε αληθή, τότε θα το αποφύγει. Έτσι έχουμε τον εξής κώδικα:

```
$sql = 'UPDATE #__mytable SET `string` = '. $db-
       >quote ($string);
```

9.1.2 Extra προστασία με χρήση component

Καλές οι μέθοδοι, αλλά για να είμαστε σίγουροι για την προστασία του ηλεκτρονικού μας καταστήματος από SQL επιθέσεις, καλό θα είναι να χρησιμοποιήσουμε και κάποιο component το οποίο θα μας προστατεύει και θα

αντιδρά αμέσως σε όποια απόπειρα επίθεσης στο site μας από κακόβολους χρήστες και συγχρόνως θα παίρνει μέτρα μέχρι να τον αποκλείσουμε.

Τέτοια components υπάρχουν πολλά, τα οποία μπορούμε να τα βρούμε από την επίσημη σελίδα του joomla! http://extensions.joomla.org/extensions/access-a-security/site-security. Επιλέγουμε πάντα αυτό που ταιριάζει καλύτερα στις απαιτήσεις του δικού μας ιστότοπου.

Αν και αξίζει να αναφέρω κάτι πολύ σοφό, κατά την κρίση μου, που ήταν η απάντηση από ένα διαχειριστή του forum.joomla.gr (http://forum.joomla.gr/viewtopic.php?f=50&t=8707) για στην ερώτηση:
"Θέλω να μάθω κατά πόσο είναι αξιόπιστα τα components που σχετίζονται τόσο με την ασφάλεια":
"**ΔΕΝ ΥΠΑΡΧΕΙ "plug-n Play" σε ότι αφορά την "ασφάλεια".** Θα χρειαστούν ώρες ίσως και ημέρες έως ότου παραμετροποιήσεις το "παιχνιδάκι". Η διασφάλιση δεδομένων είναι από τα βασικότερα πεδία εφαρμογής του *"Η ημιμάθεια σκοτώνει".*"

9.2 DoS Attack – Denial of Service Attack στο Joomla

Επιθέσεις τύπου Denial of Service[34] μπορεί να γίνουν ο χειρότερος εφιάλτης για τους διαχειριστές δικτύου. Αν επιτευχθούν, η επίλυσή τους γίνεται δύσκολα και μπορεί να κοστίσει ώρες παραγωγικότητας και να απογοητεύσει τους πελάτες που δεν θα μπορούν να έχουν πρόσβαση στις υπηρεσίες ή πρόσβαση εξ αποστάσεως στις εργασίες τους. Ωστόσο, υπάρχουν αρκετοί τρόποι για να αποτραπούν οι επιθέσεις αυτές…

Στην πρώτη και πιο σημαντική γραμμή άμυνας είναι να χρησιμοποιηθεί ένας traffic analyzer. Αυτά τα προϊόντα λογισμικού αποτελούνται από ένα σύνολο προγραμμάτων ηλεκτρονικών υπολογιστών, που συνεχώς αναλύουν την πηγή και την κίνηση των δεδομένων, κάνοντας αναζήτηση στα πιο κοινά σημάδια ανύπαρκτων αιτημάτων κυκλοφορίας και άλλων δεικτών που συνήθως βρίσκονται σας μέρη από μία DDoS επίθεση. Με αυτόν τον τρόπο φιλτράρονται τα δεδομένα και εμποδίζονται προτού φτάσουν στον server μας.

Στην επόμενη γραμμή άμυνας είναι η «μετακίνηση» του site, δηλαδή αν η επίθεση γίνεται σε συγκεκριμένη IP διεύθυνση (όπως συμβαίνει συχνά), θα μπορούσε να ξεφύγει από την επίθεση αλλάζοντας απλά την IP διεύθυνση.

Ένας τρίτος τρόπος είναι να καθορίσουμε τα request που μπορεί να δεχτεί ο ιστότοπος μας ανά λεπτό. Δημιουργούμε μια συνάρτηση με κανόνες (rules), σύμφωνα με τους οποίους περιορίζουμε τα request των χρηστών. Η συνάρτηση αυτή θα ελέγχει τα request και αν υπερβαίνουν τους κανόνες θα τα αποτρέπει. Επίσης θα μπορούσαμε να αποκλείσουμε τους ειδικούς χαρακτήρες '%' και '_' μέσω του Joomla!.

Μια κίνηση απελπισίας είναι το λεγόμενο "Blackholing". Μόλις ο διαχειριστής αντιληφθεί ότι το site δέχεται επίθεση DDoS, δημιουργεί ένα "Blackhole" site και κατευθύνει όλη την κίνηση των δεδομένων σε εκείνο (δηλαδή σε μια διεύθυνση που

[34] http://stopddosattack.com/
http://en.wikipedia.org/wiki/DDoS#Prevention_and_response

δεν υπάρχει). Με αυτήν την κίνηση αποφεύγεται το φαινόμενο της «πλημμύρας» στις υπόλοιπες τοποθεσίες του server ή του δικτύου.

9.3 Άλλες Μέθοδοι Ασφαλείας

9.3.1 Δημιουργία αντιγράφων ασφαλείας

Ένα από τα πιο σημαντικά πράγματα που πρέπει πάντα να κάνουμε, είναι να κρατάμε αντίγραφα ασφαλείας. Ακόμη κι αν δεν έχουμε πέσει θύματα επίθεσης, οι εσφαλμένες κινήσεις είναι ανθρώπινες. Κάποιες λάθος ενέργειες, ίσως μας στοιχίσουν το ίδιο ακριβά, όσο θα μας κόστιζε και μια επίθεση. Επίσης είναι και η λύτρωση μας από κάποια επίθεση που θα «κατέστρεφε» τον server μας.

Η πιο απλή λύση είναι να κρατάμε αντίγραφα από όλα τα αρχεία και τις βάσεις δεδομένων μας που χρησιμοποιούνται σε κάποιον τοπικό υπολογιστή ή σε κάποιον εξωτερικό δίσκο. Υπάρχουν πολλά εργαλεία που μπορούν να μας βοηθήσουν να κάνουμε backup, είτε μέσω των εργαλείων που εγκαταστήσαμε για να στήσουμε το site μας είτε μέσω εφαρμογών που διατίθενται στο διαδίκτυο. Εμείς στο site μας πήραμε αντίγραφο μέσω της εφαρμογής για το Joomla! Akeeba Backup[35].

Το Akeeba είναι μια εφαρμογή, με το οποίο μπορούμε να πάρουμε αντίγραφο της ιστοσελίδας μας και να το επαναφέρουμε σε οποιονδήποτε web server που υποστηρίζει το Joomla!. Τα αντίγραφα που δημιουργεί κρατιούνται σαν αρχείο και μπορεί να περιέχει όλα τα αρχεία της ιστοσελίδας, τα περιεχόμενα της βάσης δεδομένων καθώς και ένα σύστημα εύκολης επαναφοράς. Η εγκατάστασή του είναι ίδια με την εγκατάσταση ενός component του Joomla!.

Όλη η λειτουργία του Akeeba Backup είναι βασισμένη σε Ajax, ώστε να αποτρέπονται οι χρόνοι απόκρισης του server. Έχει δημιουργηθεί από Έλληνες προγραμματιστές με επικεφαλή το Νικόλαο Διονυσόπουλο.

[35] http://www.joomplus.gr/reviews/item/783-akeeba-backup.html

9.3.2 Αλλαγή των δικαιωμάτων των αρχείων

Σε περίπτωση που είχαμε χρησιμοποιήσει διαφορετικό λειτουργικό και όχι windows (Linux, FreeBSD κ.ά.) για να στήσουμε το site μας, θα μας απασχολούσε και το θέμα των δικαιωμάτων των αρχείων. Όλα τα στοιχεία που περιέχονται σε μια ιστοσελίδα (φάκελοι, αρχεία) έχουν κάποια δικαιώματα χρήσης. Τα δικαιώματα αυτά, καθορίζουν τι ενέργειες επιτρέπεται να κάνει ο εκάστοτε χρήστης με το εκάστοτε στοιχείο.

Υπάρχουν τρεις τύποι δικαιωμάτων:

- read
- write
- execute

Τα δικαιώματα είναι ξεχωριστά για τον κάθε χρήστη και συνήθως ορίζονται από ένα τριψήφιο αριθμό. Τον πιο αυστηρό περιορισμό τον δηλώνουν οι αριθμοί 000, σύμφωνα με τον οποίο κανένας χρήστης δεν έχει κανένα δικαίωμα για οποιαδήποτε ενέργεια. Το εντελώς αντίθετο από το 000 το παίρνουμε με τον τριψήφιο 777 που μας δίνει το ελεύθερο για όλες τις χρήσεις.

Το πρώτο ψηφίο από τον αριθμό, αντιπροσωπεύει τα δικαιώματα του ιδιοκτήτη/ δημιουργού(owner), το δεύτερο των υπόλοιπων εξουσιοδοτημένων χρηστών(group) και το τρίτο τα δικαιώματα των τρίτων(all).

Για να είμαστε και ασφαλείς αλλά και να υπάρχει και χρηστικότητα των στοιχείων του ιστοτόπου μας, οι φάκελοι θα πρέπει να οριστούν σε 755 και τα αρχεία σε 644 (εκτός κι αν επιθυμούμε κάποια διαφορετική ρύθμιση για συγκεκριμένα αρχεία η φακέλους). Η αλλαγή των δικαιωμάτων γίνεται μέσω του chmod.

Ωστόσο στο Joomla!, ανάλογα με τις ρυθμίσεις της εγκατάστασης που δώσαμε (ή μέσω του Global Configuration) καθορίζει απευθείας τα δικαιώματα των αρχείων που δημιουργούνται.

Επειδή όμως χρησιμοποιώντας έναν FTP client ή ένα host πίνακα ελέγχου όπως cPanel ή Plesk, μπορεί κάποιος να αλλάξει τα δικαιώματα, θα ήταν ασφαλέστερο τα δικαιώματα των στοιχείων να οριστούν σε 644 (χωρίς την άδεια εγγραφής). Ιδιαίτερα με το αρχείο configuration.php, στο οποίο αποθηκεύονται οι ρυθμίσεις μας από το Global Configuration, να φροντίζουμε να είναι τροποποιήσιμο μόνο όταν θέλουμε να κάνουμε κάποιες αλλαγές.

Αρχείο configuration.php

Ο πιο απλούστερος και ασφαλέστερος τρόπος για την προστασία του αρχείου configuration.php, θεωρείται να μην αποθηκεύονται κρίσιμα δεδομένα στον κατάλογο public_html. Από τον Apache.org υπάρχει μια συνεχής σύσταση να αποφεύγεται η διατήρηση τέτοιων αρχείων σε αυτόν τον κατάλογο.

Για να προστατέψουμε λοιπόν τα δεδομένα του αρχείου, το μετακινούμε σε ασφαλή κατάλογο, έξω από τον public_html και το μετονομάζουμε (π.χ. vassiliki.conf). Κατόπιν δημιουργούμε ένα νέο αρχείο configuration.php, στο οποίο βάζουμε τον παρακάτω κώδικα:

```
<?php
require ( dirname ( __FILE__ ) . '/…/vassiliki.conf' );
?>
```

Σιγουρευόμαστε ότι το νέο configuration.php δεν είναι εγγράψιμο (444) και ότι δεν θα αλλάξει το περιεχόμενό του από το com_config. Αν χρειαστεί να αλλαχθούν κάποιες ρυθμίσεις, θα τις αλλάξουμε με το χέρι στο vassiliki.conf.

Χρησιμοποιώντας αυτή τη μέθοδο, ακόμη και αν για κάποιο λόγο ο web server μεταδώσει τα περιεχόμενα των αρχείων php (λόγω κάποιας λανθασμένης ρύθμισης), κανείς δεν θα μπορεί να δει τα περιεχόμενα του πραγματικού configuration.php αρχείου.[36]

9.3.3 Χρησιμοποιώντας το αρχείο htaccess.txt

Κατά την εγκατάσταση του Apache HTTP Server δημιουργήθηκε και το αρχείο htaccess.txt, το οποίο όσο βρίσκεται σε αυτή τη μορφή δεν έχει καμία επίπτωση για τον διαδικτυακό μας τόπο. Αν το μετονομάσουμε σε .htaccess θα έχουμε ένα πολύ ισχυρό εργαλείο του Apache, το οποίο με τις κατάλληλες ρυθμίσεις μπορεί:

- να κρατήσει μακριά τους «ανεπιθύμητους επισκέπτες» ή να τους παραπέμψει αλλού
- να προστατέψει τις ιστοσελίδες και τους καταλόγους με κωδικούς πρόσβασης
- να κάνουμε την ιστοσελίδα μας φιλική σε μηχανές αναζήτησης (SEF urls)
- …. κ.ά.

κοινός, αυτό το αρχείο μπορεί να γίνει πολύ ισχυρό…

Ας δούμε μερικές τεχνικές για το πως μπορούμε να αξιοποιήσουμε το αρχείο .htaccess, ώστε παραμετροποιώντας το, να αυξήσουμε την ασφάλεια του ηλεκτρονικού μας καταστήματος.

Αποκλεισμός της IP ενός ανεπιθύμητου επισκέπτη

Σε γενικές γραμμές, μια έγκυρη IP θα πρέπει να έχει τη μορφή xxx.xxx.xxx.xxx, όπου “xxx” είναι ένας αριθμός μεταξύ 0-255. Εισάγοντας ένα τμήμα μιας ανεπιθύμητης IP, μπορούμε να αποκλείσουμε όλες τις IP που περιέχουν το εν λόγω τμήμα μέσα σε αυτό το εύρος. Αυτό όμως είναι κάτι που θέλει μεγάλη προσοχή, γιατί μπορεί να αποκλείσουμε χρήσιμες και υγιείς επισκέψεις (οι «κακές» IP συνήθως είναι Blacklisted).

Ας υποθέσουμε πως θέλουμε να αποκλείσουμε την 192.168.221 (τυχαία διεύθυνση). Για να το κατορθώσουμε να αποκλείσουμε αυτή την IP (και όλες όσες είναι μέσα στην περιοχή 192.168.221.xxx, πρέπει να προσθέσουμε τον παρακάτω κώδικα στο .htaccess:

```
## USER IP BANNING
<Limit GET POST>
   order allow,deny
   deny from 192.168.221.
```

[36] http://www.joomla.gr/tutorials/security/349--configurationphp

```
    allow from all
</Limit>
```

Αποκλεισμός ανεπιθύμητου spam traffic από Site Referrers-αποτροπή μείωσης του bandwidth του server

Ο παρακάτω κώδικας έχει σκοπό να απαγορέψει τις επισκέψεις που προέρχονται από άλλα sites. Για παράδειγμα, εάν μια ιστοσελίδα είναι ανεπιθύμητη ή εμφανίζονται πολλές καταγραφές στο web site referrer log, μπορούμε να το απαγορέψουμε, έτσι ώστε κάθε επίσκεψη και traffic που προέρχεται από αυτό είτε είναι χρήστης είτε κάποιο bot, να αποκλείεται. Κάποια spam sites συνηθίζουν αυτές τις επισκέψεις, για να καταγράφουν το URL της επίσκεψής τους στο web site referrer log και να είναι αναγνωρίσιμο από τις μηχανές αναζήτησης σαν backlink. Τα spam sites είναι ανεπιθύμητα γιατί βομβαρδίζουν τον server με επισκέψεις, οι οποίες συνήθως γίνονται με κάποιο bot και το αποτέλεσμα είναι να μειώνουν το ωφέλιμο bandwidth του server.

Αν για παράδειγμα θέλουμε να αποκλείσουμε τα παρακάτω domains ή τις IPs (και τα domains και οι IPs είναι τυχαίες):

- ✓ 94.65.5.32 (Αποκλείει μια συγκεκριμένη διεύθυνση IP)
- ✓ 215.153.42. (Αποκλείει όλες τις IPs μέσα στην περιοχή 215.153.42.xxx)
- ✓ 93.32. (Αποκλείει όλες τις IPs μέσα στην περιοχή 93.32.xxx.xxx)
- ✓ 81.158.3 (Αποκλείει όλες τις IPs μέσα στην περιοχή 81.158.3xx.xxx)

Ο κώδικάς μας λοιπόν θα έχει την παρακάτω μορφή:

```
##SITE REFERRER BANNING
RewriteEngine on
# Options +FollowSymlinks
RewriteCond %{HTTP_REFERER} verybadsite\.com [NC,OR]
RewriteCond %{HTTP_REFERER} verybadsite\. [NC,OR]
RewriteCond    %{HTTP_REFERER}    sub\.verybadsite\.com
[NC,OR]
RewriteCond %{HTTP_REFERER} 32\.173\.21\.187 [NC]
RewriteRule .* - [F]
```

Απενεργοποίηση και απαγόρευση του Hotlinking

Η απενεργοποίηση του Hotlinking είναι μια απαγόρευση για συνηθισμένα κοινά αρχεία από άλλες ιστοσελίδες, έτσι ώστε μόνο το δικό μας domain να μπορεί να αναφέρεται ή να έχει πρόσβαση σε αυτά. Για παράδειγμα, με την απενεργοποίηση του Hotlinking για τα αρχεία .jpg, οποιοδήποτε ιστοσελίδα η οποία δεν είναι μέσα στη λίστα των επιτρεπόμενων domain, θα παίρνει μια αναφορά ανύπαρκτου αρχείου εικόνας, για το αρχείο .jpg που βρίσκεται στον server μας.

Στη σύνταξη του κώδικα βάζουμε τα domain και τους τύπους αρχείων για τα οποία θα γίνεται το Hotlinking. Αν υποθέσουμε ότι θέλουμε να επιτρέψουμε κάποια Domains ή IPs για κάποιους τύπους αρχείων και να απαγορεύσουμε κάποια αρχεία

για Hotlinking από όλα τα υπόλοιπα (οι παρακάτω IPs και τα Domains είναι τυχαία), τότε ο κώδικάς μας θα έχει την παρακάτω μορφή:

```
## DISABLE HOTLINKING
RewriteEngine on
# Options +FollowSymlinks

RewriteCond %{HTTP_REFERER} !^$

RewriteCond %{HTTP_REFERER} !^http://(www\.)?webmasterslife.
gr/.*$ [NC]

RewriteCond %{HTTP_REFERER} !^http://(www\.)?forum.webmaster
slife.gr/.*$ [NC]

RewriteCond %{HTTP_REFERER} !^http://(www\.)?95.65.23.195/.*
$ [NC]

RewriteRule \.(gif|jpg|png|css|js)$ - [F]
```

Για "Επιτρεπόμενα Domains/ IPs"

- ✓ webmasterslife.gr (Επιτρέπετε σε αυτό το domain η πρόσβαση στους συγκεκριμένους τύπους αρχείων)
- ✓ forum.webmasterslife.gr (Επιτρέπετε σε αυτό το subdomain η πρόσβαση στους συγκεκριμένους τύπους αρχείων)
- ✓ 95.65.23.195 (Επιτρέπετε σε αυτό την IP η πρόσβαση στους συγκεκριμένους τύπους αρχείων)

Για τη "Λίστα τύπου Αρχείων"

- ✓ gif (Απαγόρευση του hotlinking στα .gif αρχεία σε αυτόν τον server)
- ✓ jpg (Απαγόρευση του hotlinking στα.jpg αρχεία σε αυτόν τον server)
- ✓ png (Απαγόρευση του hotlinking στα.jpg αρχεία σε αυτόν τον server)
- ✓ css (Απαγόρευση του hotlinking στα.css αρχεία σε αυτόν τον server, αντί αυτού να εμφανίζετε κενό αρχείο)
- ✓ js (Απαγόρευση του hotlinking στα.js αρχεία σε αυτόν τον server, αντί αυτού να εμφανίζετε κενό αρχείο)

Αποφυγή επιθέσεων τύπου Injection και Cross-Site Scripting

Για να αποφευχθούν επιθέσεις τύπου Global Variable Injection, Code Injection και Cross-Site Scripting(XSS), ρυθμίζουμε τη δομή των php Boolean directives χρησιμοποιώντας php_flag.

- ❖ Κώδικας πρόληψης από Code Injection
  `php_flag magic_quotes_gpc on`

- ❖ Κώδικας πρόληψης από Global Variable Injection
  `php_flag register_globals off`

- ❖ Κώδικας πρόληψης από επιθέσεις Cross-Site Scripting
  `php_flag allow_url_fopen off`

Περιορισμός πρόσβασης σε κατάλογο

Πιο πάνω αναφερθήκαμε στον αποκλεισμό της IP κάποιου ανεπιθύμητου επισκέπτη. Για την προστασία του καταλόγου του διαχειριστή του ιστοτόπου λοιπόν θα πρέπει να περιορίσουμε την πρόσβαση του καταλόγου του (με τον ίδιο τρόπο εξασφαλίζουμε την προστασία και άλλων καταλόγων που επιθυμούμε), έτσι όποιος προσπαθήσει να περιηγηθεί στους καταλόγους αυτούς με μια διαφορετική IP διεύθυνση από αυτή που έχουμε ορίσει, θα του εμφανίζει σφάλμα “403 Forbidden”. Η παρακάτω μέθοδος λειτουργεί μόνο αν υπάρχει μια στατική διεύθυνση IP.

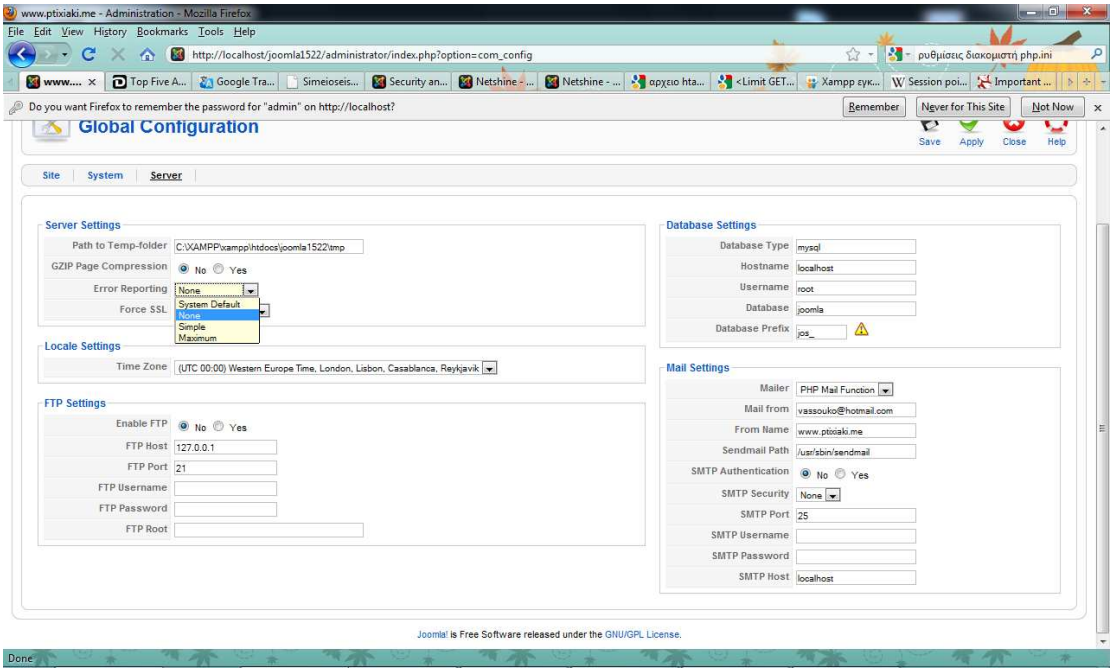
Για να προστατεύσουμε τον επιθυμητό κατάλογο, ανοίγουμε το .htaccess του (αν δεν υπάρχει δημιουργούμε ένα με ένα notepad) και προσθέτουμε τον παρακάτω κώδικα. Εμείς εδώ χρησιμοποιήσαμε την 100.100.100.100 IP ότι είναι η IP που επιτρέπουμε:

```
<Limit GET>
   Order Deny, Allow
   Deny from all
   Allow from 100.100.100.100
</Limit>
```

9.3.4 Αρχείο php.ini – Ρυθμίσεις διακομιστή

Κατά την εγκατάστασή του το Joomla! έχει κάποιες default ρυθμίσεις για τη σωστή λειτουργία του συστήματος. Μια από αυτές τις ρυθμίσεις είναι και η ρύθμιση του server “Error Reporting” που είναι default ρυθμισμένη να είναι ενεργοποιημένη. Η ρύθμιση αυτή, ενώ είναι πολύ χρήσιμη κατά την ανάπτυξη και την αποσφαλμάτωση ενός ιστοτόπου, δημιουργεί μια ευπάθεια σε κάποιες εκδόσεις της PHP και μπορεί να επιτρέψει επιθέσεις Cross-Site Scripting σε περίπτωση που υπάρχει κάποιο script που δημιουργεί σφάλμα.

Για να καταστείλουμε αυτά τα μηνύματα λάθους, μέσω της σελίδας διαχείρισης του ηλεκτρονικού μας καταστήματος, μεταβαίνουμε στο Global Configuration και κατόπιν στην καρτέλα Server. Εκεί ορίζουμε την τιμή για το “Error Reporting” να είναι “None”.



Εικόνα 34: Joomla! Global Configuration, Ρυθμίσεις του Server για την αποστολή αναφορών λαθών

Για να απενεργοποιήσουμε όμως την εμφάνιση όλων των λαθών στην PHP, πρέπει να κάνουμε και τις απαραίτητες ρυθμίσεις στο αρχείο php.ini. Οι ρυθμίσεις που πρέπει να υπάρχουν στο php.ini είναι οι εξής:

```
Display_errors = Off
Html_errors = Off
Display_startup_errors = Off
Log_errors = On
```

Με τις παραπάνω ρυθμίσεις πετυχαίνουμε τα τυχόν PHP λάθη που θα δημιουργηθούν, να συνδεθούν σε ένα αρχείο κειμένου και όχι να εμφανιστούν στο παράθυρο του browser του εκάστοτε χρήστη.

9.3.5 Το όνομα του Υπερδιαχειριστή (Super Administrator)

Κατά την εγκατάσταση ενός Joomla! 1.5.x, στο τελευταίο στάδιο, ζητείται να δοθεί ο κωδικός χρήστη και η ηλεκτρονική θυρίδα αλληλογραφίας του Υπερδιαχειριστή (Super Administrator). Το όνομα χρήστη (username) δυστυχώς δεν μπορούμε να το ορίσουμε εκείνη τι στιγμή, έχει προκαθορισθεί να είναι admin και θα μπορούσε να αποδειχθεί επικίνδυνος, σε κάποιες περιπτώσεις.

Η λύση είναι να αλλάξουμε το όνομα αυτό μόλις κάνουμε την πρώτη σύνδεσή μας στη διαχείριση του Joomla! Μέσα από το μενού Site → User Manager, επιλέγουμε να επεξεργαστούμε τον Super Administrator και δίνουμε ένα άλλο όνομα χρήστη (και σίγουρα όχι admin).[37]

[37] http://www.joomla.gr/tutorials/security/405-sa-username

ΕΠΙΛΟΓΟΣ

Στην εργασία αυτή μελετήσαμε τα διαθέσιμα CMS και επιλέξαμε το Joomla! από τα open source για να δημιουργήσουμε ένα ηλεκτρονικό κατάστημα. Στήσαμε με τη βοήθεια του εργαλείου XAMPP τον server μας. Κατόπιν με το component VirtueMart δημιουργήσαμε τον ιστότοπο του ηλεκτρονικού μας καταστήματος. Έπειτα μελετήσαμε τη συμπεριφορά του συστήματος όταν αυτό υποστεί επιθέσεις διαφόρων ειδών.

Συγκεκριμένα είδαμε πως στις επιθέσεις υπερχείλισης μνήμης, ο επιτιθέμενος εκμεταλλεύεται κάποια ελαττώματα του συστήματος που υπάρχουν στις βιβλιοθήκες του λειτουργικού συστήματος ή των εφαρμογών που χρησιμοποιούμε, ώστε να πετύχει την εκτέλεση δικών του κακόβουλων εντολών τις οποίες εισάγει ως επιπλέον πληροφορία. Διαπιστώσαμε ότι συνεχείς αναβαθμίσεις του λειτουργικού συστήματος και των εφαρμογών που χρησιμοποιούμε μπορούν να αποτρέψουν μια τέτοια επίθεση.

Η βάση δεδομένων είναι ίσως το πιο σημντικό μέρος του συστήματος, καθώς εκεί βρίσκονται το περιεχόμενο, τα ευαίσθητα στοιχεία των χρηστών, οι ρυθμίσεις μας και πολλά ακόμη. Συνήθως ο επιτιθέμενος, μέσω των φορμών που υπάρχουν στην ιστοσελίδα, παρεμβάλει SQL κώδικα με στόχο την πρόσβαση στη βάση. Για να αποφύγουμε τέτοιου είδους επιθέσεων πήραμε τα κατάλληλα μέτρα ασφαλείας μέσω ρυθμίσεων του Joomla! και με τη χρήση SQL διατυπώσαμε ερωτήματα για ελέγχονται όλα τα δεδομένα που εισάγονται μέσω των φορμών της ιστοσελίδας μας

Η προσπάθεια απόσπασης προσωπικών δεδομένων, συνήθως οικονομικου χαρακτήρα, μέσω spam email είναι συνηθισμένη τακτική ονομαζόμενη Phishing. Για να προστατέψουμε τους πελάτες μας, δυστυχώς το μόνο που μπορούσαμε να κάνουμε ήταν να τους δώσουμε συμβουλές ασφαλείας anti-phishing μέσω της πιστοποιημένης σελίδας μας.

Άλλη μια συνηθισμένη επίθεση που ασχοληθήκαμε ήταν η Cross-Site Scripting, η οποία μπορεί να ενσωματώσει ένα script, το οποίο θα εκτελείται κάθε φορά που θα φορτώνεται η σελίδα ή κάθε φορά που θα τρέχει ένα σχετικό event. Συνηθισμένο τέχνασμα είναι η εμφάνιση ενός URL πανομοιότυπου με αυτό της σελίδας που θέλει να σφετεριστεί, με σκοπό να αποπλανήσει τον χρήστη, να κλέψει τα στοιχεία που επιθυμεί μέσω του cookie του χρήστη και να τα οικειοποιηθεί. Για την πρόληψη της ασφάλειας των πελατων μας από τέτοιου είδους επιθέσεις, ρυθμίσαμε την δομή των php Boolean directives χρησιμοποιώντας php_flag κώδικα και αλλάζοντας τις απαραίτητες ρυθμίσεις του διακομιστή, μέσω της σελίδας διαχείρισης του Joomla!..

Η παρακολούθηση των πακέτων ήταν άλλο ένα θέμα για να αναπτύξουμε, καθώς με ένα λογισμικό παρακολούθησης πακέτων (packet sniffer), μπορεί ο κακόβουλος χρήστης να υποκλέψει τα δεδομένα που περιέχονται στα πακέτα δεδομένων χωρίς να κινήσει τις υποψίες των χρηστών. Για να αχρηστεύσουμε έναν sniffer χρησιμοποιήσαμε τη μέθοδο της κρυπτογράφησης μέσω ανάλογων πρωτοκόλλων.

Οι επιθέσεις DoS είναι οι επιθέσεις που κάνουν τον server μας να αργεί να ανταποκριθεί στους χρήστες και μπορεί να προκαλέσει μέχρι και κατάρρευση του συστήματος χρησιμοποιώντας υπολογιστές “zombies”. Αυτού του είδους οι επιθέσεις μπορεί να γίνουν ο χειρότερος εφιάλτης για τους διαχειριστές. Για να τις

αποτρέψουμε ήταν να χρησιμοποιήσαμε λογισμικό για ανάλυση της πηγής και της κίνησης των δεδομένων, με χρήση κανόνων καθορίσαμε τα request που μπορεί να δεχτεί ο ιστότοπος μας και αποκλείσαμε τους ειδικούς χαρακτήρες ‘%’ και ‘_’.

Επίσης για να είμαστε πιο σίγουροι για την ασφάλεια πήραμε επιπλέον μέτρα. Για αρχή αλλάξαμε τα δικαιώματα των φακέλων και των αρχείων, αλλάξαμε θέση το configuration.php αρχείο και στη θέση του δημιουργήσαμε ένα καινούργιο, ώστε να προστατέψουμε τα κρίσιμα δεδομένα που περιέχει. Μέσω του .htaccess αρχείου αποκλείσαμε ανεπιθύμητες IP διευθύνσης, ανεπιθύμητο spam traffic για την αποτροπή μείωσης του bandwidth και τέλος απενεργοποιήσαμε και απαγορεύσαμε το Hotlinking. Τέλος περιορίσαμε την πρόσβαση στον κατάλογο του διαχειριστή μέσω του .htaccess και από τη σελίδα του Back-End του Joomla! αλλάξαμε το όνομα του Super Administrator από το default “admin” σε ένα της επιλογής μας.

Για την ανάπτυξη ενός υγειούς και ασφαλούς ιστότοπου χρησιμοποιήσαμε τα πρωτόκολλα S-HTTP, SSL, SET, EDI , ψηφιακών υπογραφών και ψηφιακών πιστοποιητικών. Τέλος, με χρήση ενός component (Akeeba) δημιουργήσαμε αντίγραφα ασφαλείας, για να έχουμε backup ολόκληρου του site μας.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. “ Packt, Joomla! Web Security, 2008 (Secure your Joomla! website from common security threats with this easy-to-use guide.) “

2. “ Prentice Hall, Joomla Guide 1.5 Joomlabook, 2008 “

3. “ Wiley, Joomla Bible, 2010 “

ΠΗΓΕΣ

http://docs.joomla.org/

http://www.dclab.com/component_content_management.asp

http://docs.joomla.org/Framework

http://www.ibm.com/developerworks/aix/library/au-cmsaix/

http://www.ibm.com/developerworks/ibm/library/i-osource1/

http://el.wikipedia.org/wiki/%CE%91%CF%81%CF%87%CE%B5%CE%AF%CE%BF:Virtuemart_slogan_blue.png

http://virtuemart.net/documentation/User_Manual/Installation.chapter.html

http://virtuemart.net/documentation/User_Manual/Administrator_Tutorial.html

http://www.ngssoftware.com/papers/advanced_sql_injection.pdf

http://www.symantec.com/connect/articles/common-security-vulnerabilities-e-commerce-systems

http://www.cs3-inc.com/pk_whatisddos.html

http://thabettech.blogspot.com/

http://h10163.www1.hp.com/technology_phishing.html

http://www.windowsecurity.com/articles/Web-Applications.html

http://support.huawei.com/support/pages/kbcenter/view/product.do?actionFlag=searchManualContents&web_doc_id=SE0000483032&material_type=ProductManual&part_no=10082

http://www.vehem.fr/fr/competences/habillage-cms.php

http://tmjcss.com/page.asp?cs=2&catid=508

http://developer.practicalecommerce.com/articles/1489-Top-Five-Application-Security-Risks-for-2010

http://www.blackhat.com/presentations/bh-dc-08/Willis/Whitepaper/bh-dc-08-willis-WP.pdf

http://www.neohaxor.org/2009/08/11/dynamic-cross-site-request-forgery/

http://www.slideshare.net/guestbd1cdca/joomla-security-nuggets

http://www.islab.demokritos.gr/gr/html/ptixiakes/kostas-aris_ptyxiakh/Phtml/ssl.htm

http://www.joomlablogger.net/joomla-tips/joomla-general-tips/joomla-backup-reliable-solution/

http://www.netshinesoftware.com/security/joomla-security.html

http://forum.joomla.gr/viewtopic.php?f=54&t=8932

http://www.webmasterslife.gr/search-engine-optimization/76-%CE%A7%CF%81%CE%B7%CF%83%CE%B9%CE%BC%CE%BF%CF%80%CE%BF%CE%B9%CF%8E%CE%BD%CF%84%CE%B1%CF%82-%CF%84%CE%BF-%CE%B1%CF%81%CF%87%CE%B5%CE%AF%CE%BF-htaccess.html

http://digitalsignature.in

http://mikrospin.si

http://www.easy-servers.gr/content/view/7/6/

http://internetcorkboard.com

http://www.ibm.com/developerworks/websphere/library/techarticles/0504_mckegney/0504_mckegney.html

http://e-pcmag.gr/

http://www.easy-servers.gr/content/view/7/6/

http://www.coder.gr/article.php?story=20060707225712652

http://www.vdimitris.gr/mysql.php?seo=24

http://en.wikipedia.org/wiki/Content_management_system

http://www.google.gr/search?q=vulnerabilities&ie=utf-8&oe=utf-8&aq=t&rls=org.mozilla:el:official&client=firefox-a

http://en.wikipedia.org/wiki/Content_management_system

http://www.easy-servers.gr/content/view/9/6/