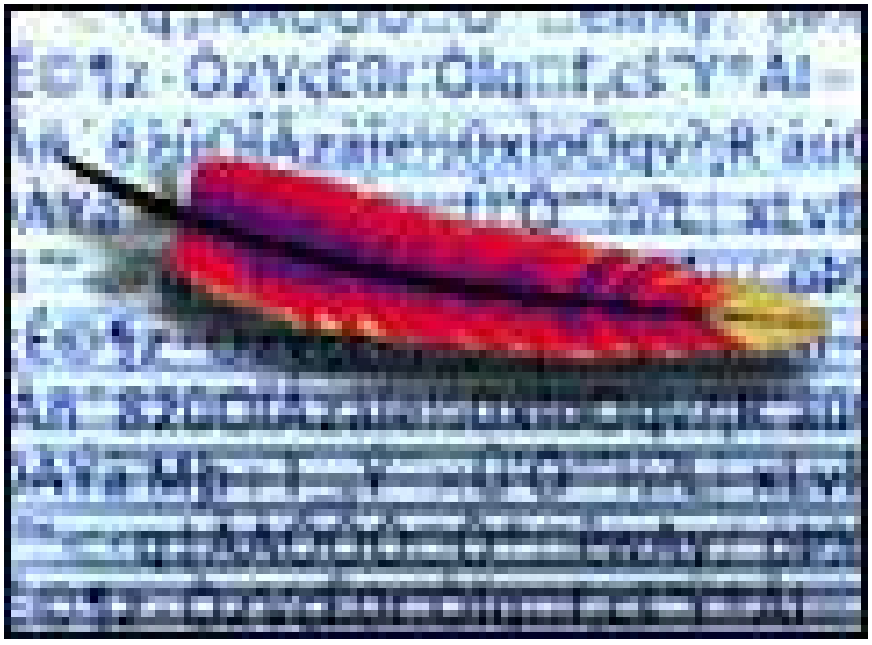
Τμήμα Τηλεπληροφορικής & Διοίκησης Τ.Ε.Ι. Ηπείρου Άρτας



ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΔΙΑΧΕΙΡΙΣΗ ΤΟΥ APACHE SERVER ΣΕ ΠΕΡΙΒΑΛΛΟΝ WINDOWS

ΟΝΟΜΑ ΣΠΟΥΔΑΣΤΡΙΑΣ: ΣΑΒΒΑ ΚΩΝΣΤΑΝΤΙΑ

ΥΠΕΥΘΥΝΟΣ ΚΑΘΗΓΗΤΗΣ : ΚΥΡΙΟΣ ΠΑΠΑΜΩΚΟΣ ΓΕΩΡΓΙΟΣ

ΘΕΜΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ : ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΔΙΑΧΕΙΡΙΣΗ ΤΟΥ APACHE SERVER ΣΕ ΠΕΡΙΒΑΛΛΟΝ WINDOWS

ΑΡΤΑ 2005

Τμήμα Τηλεπληροφορικής & Διοίκησης

Τ.Ε.Ι. Ηπείρου Άρτας

ΕΥΧΑΡΙΣΤΙΕΣ

Στο σημείο αυτό θα ήθελα να ευχαριστήσω τον καθηγητή μου Κύριο Παπαμώκο Γεώργιο για την βοήθεια που μου έδωσε ούτως ώστε να φέρω εις πέρας την πτυχιακή μου εργασία. Επίσης θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες στην οικογένεια μου που με στήριξε καθ' όλη την διάρκεια των σπουδών μου.

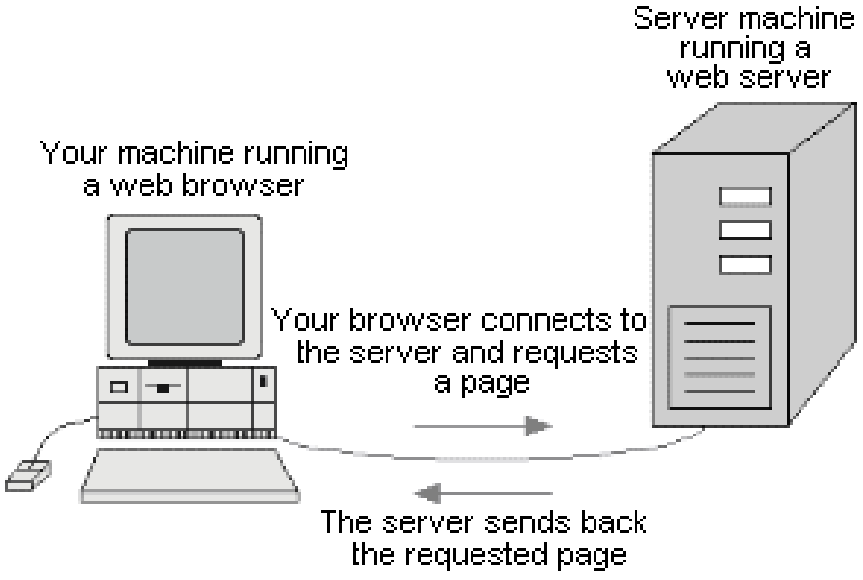
ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

1. Τι είναι και πώς λειτουργούν οι διακομιστές Web

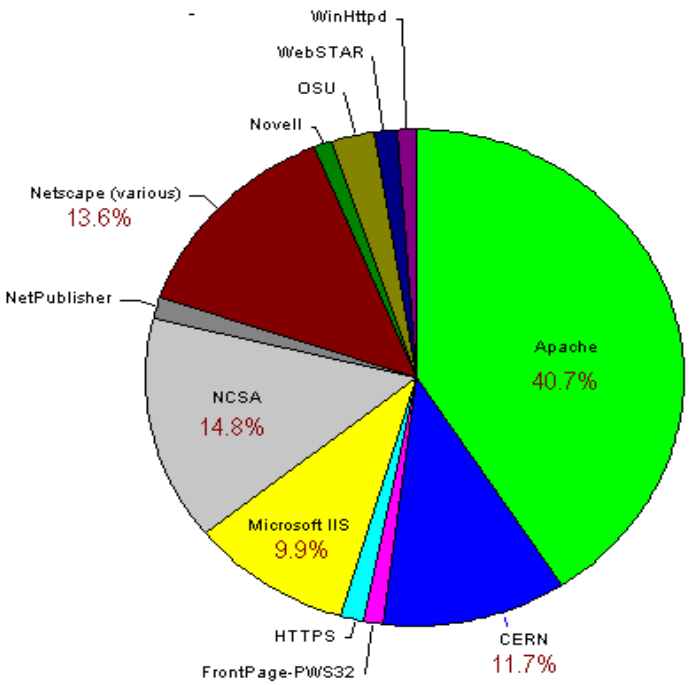
Ο διακομιστής Web (web server) είναι ένα λογισμικό το οποίο μας παρέχει ο World Wide Web.Το λογισμικό αυτό λαμβάνει αιτήσεις από έναν «πελάτη» όπως για παράδειγμα ο δημοφιλής Internet Explorer. Αφού πάρει μία αίτηση την επεξεργάζεται και επιστρέφει στον «πελάτη» δεδομένα τα οποία αντλεί από κάποια κατάλληλα διαμορφωμένη σελίδα. Τα δεδομένα αυτά μπορεί να είναι κείμενο, εικόνες γραφικών video clips και άλλα πολυμέσα. Ο «πελάτης» ή αλλιώς Browser στη συνέχεια αφού λάβει τα δεδομένα τα προβάλλει στον χρήστη δίνοντας τους την κατάλληλη μορφή. Όταν ένας χρήστης εκτελεί μια ενέργεια, όπως το να πατήσει ένα σύνδεσμο ή να καταθέσει μία φόρμα, αποστέλλεται στον διακομιστή ένα μήνυμα το οποίο μεταφέρει την ενέργεια που έγινε, μαζί με τα σχετικά δεδομένα, για παράδειγμα το όνομα μέσα σ' ένα πεδίο κειμένου το οποίο πληκτρολογήθηκε από τον χρήστη.
Η επικοινωνία ανάμεσα στον διακομιστή και στον πελάτη γίνετε με ένα πρωτόκολλο γνωστό σαν Hypertext Transfer Protocol (HTTP) το οποίο καθορίζει τους κανόνες επικοινωνίας έτσι ώστε να μπορεί οποιοσδήποτε browser να συνδεθεί σε οποιονδήποτε web server. Τα περισσότερα κείμενα που μεταδίδονται από τους web server είναι διαμορφωμένα με την γνωστή Hypertext Markup Language (HTML) την οποία οι browsers μεταφράζουν για να προβάλουν το τελικό αποτέλεσμα στον χρήστη.



1.1 Εισαγωγή στον Apache Server

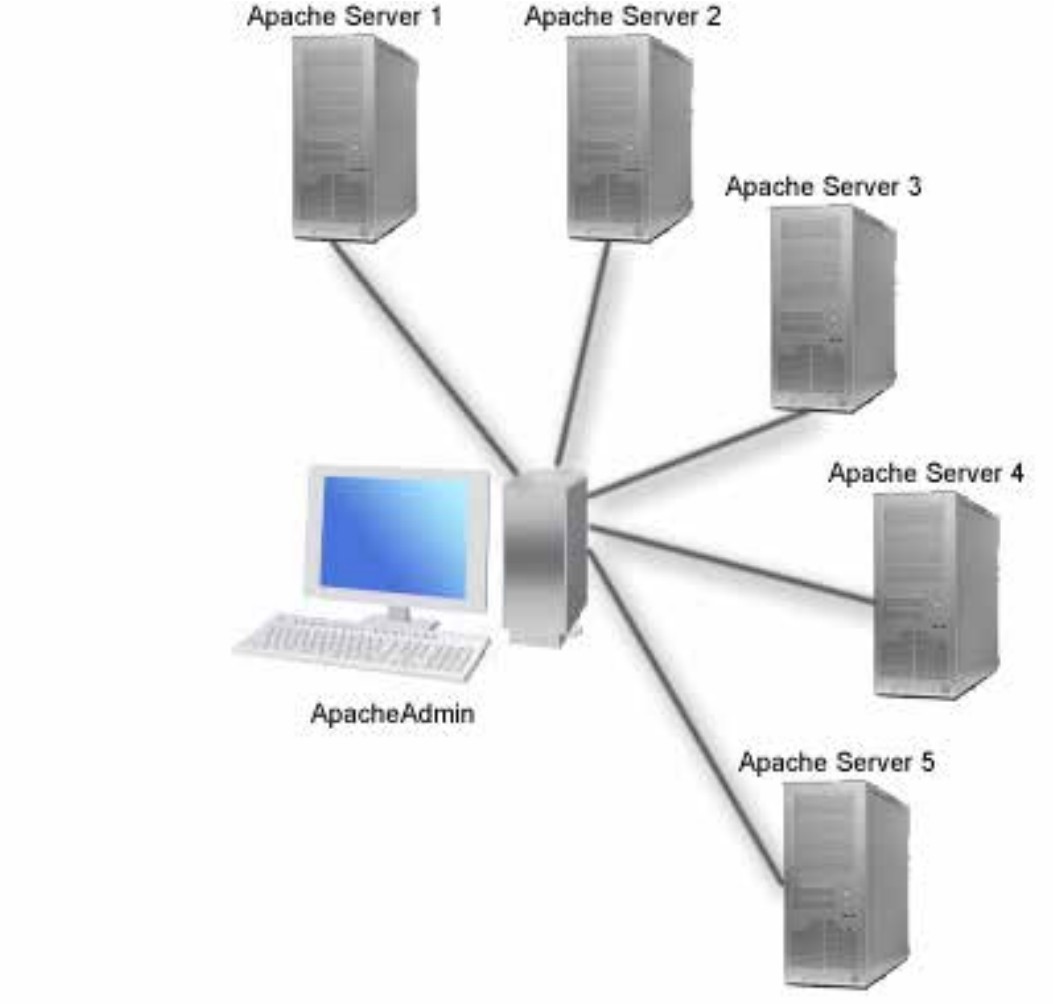
Ο Apache server δημιουργήθηκε στον σωστό χώρο την σωστή στιγμή αφού δεν υπήρχε τίποτε άλλο όμοιο με αυτό. Οι άνθρωποι που δημιουργούσαν τις ιστοσελίδες χρειάζονταν ορισμένα χαρακτηριστικά γνωρίσματα και κάποιους σταθερούς τρόπους για την διόρθωση των λαθών. Έτσι γεννήθηκε ο Apache που είναι λογισμικό από τους χρήστες για τους χρήστες. Το μοντέλο ανοικτού προτύπου ήταν ιδανικό γι' αυτό το έργο (Apache server) γιατί ειδικά τις πρώτες μέρες του Ιστού(Web) τα πράγματα κινούνταν πολύ γρήγορα έτσι καμία εταιρεία δεν μπορούσε να συμβαδίσει. Οι άνθρωποι όμως δεν ανέχονταν να περιμένουν έναν μηχανικό να αποφασίσει ότι έπρεπε να δημιουργηθεί ένα προϊόν. Χρειάζονταν ένα χαρακτηριστικό γνώρισμα επειγόντως έτσι έπρεπε να το δημιουργήσουν οι ίδιοι.

Ο Apache Web Server είναι ο δημοφιλέστερος web server αφού λειτουργεί σε όλα τα γνωστά λειτουργικά συστήματα Windows, Netware καθώς επίσης και στις περισσότερες εκδόσεις τον Unix/Linux συστημάτων.



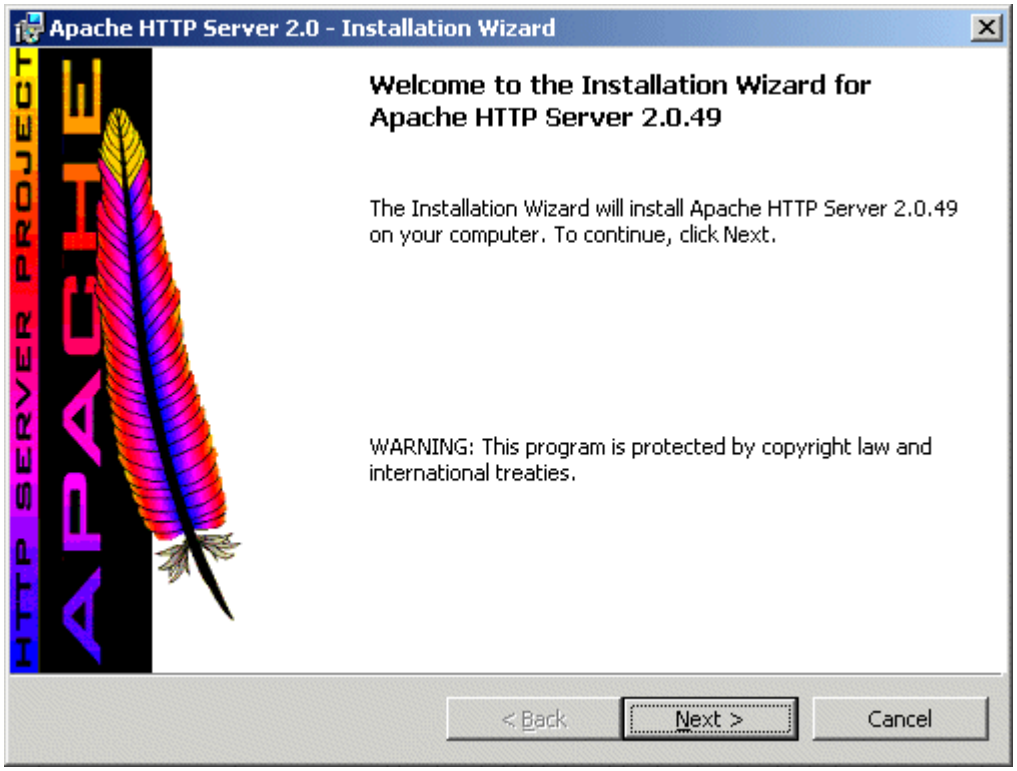
ΚΕΦΑΛΑΙΟ 2

ΕΓΚΑΤΑΣΤΑΣΗ



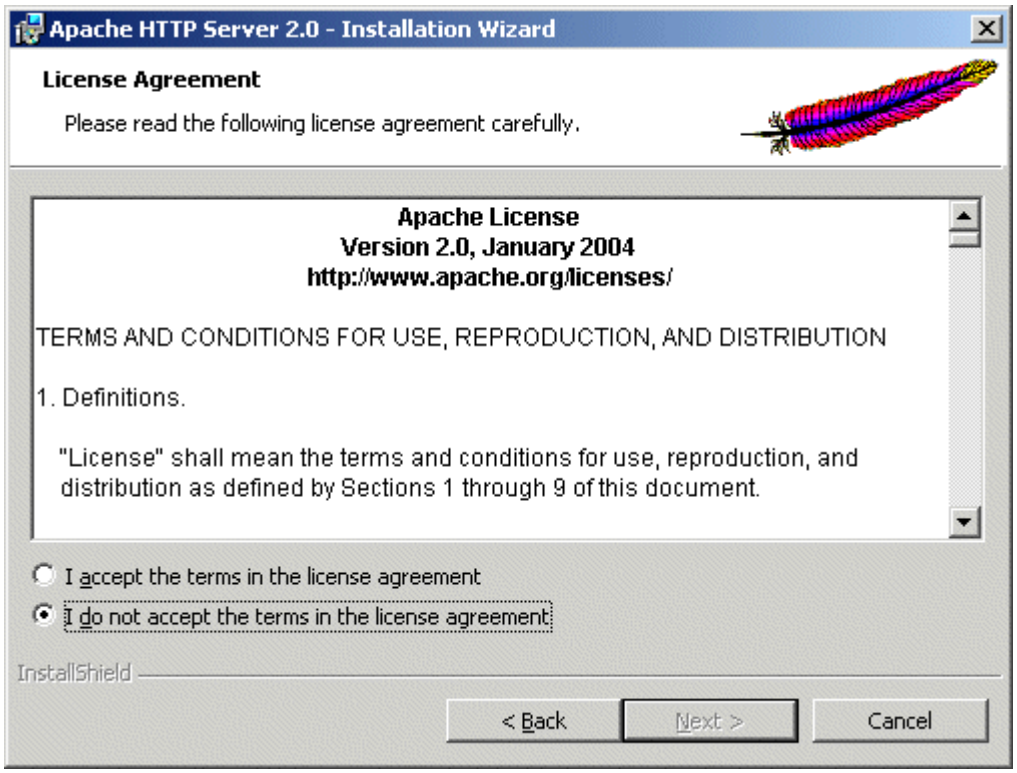
2. Εγκατάσταση του Apache Server

Πιο κάτω θα δούμε την διαδικασία εγκατάστασης του Apache σε περιβάλλον Windows. Πριν όμως αρχίσει η εγκατάσταση θα πρέπει να είμαστε σίγουροι ότι δεν τρέχει κάποιος άλλος Web Server στο σύστημα μας. Για τον λόγο αυτό θα ήταν καλό να καταργήσουμε την εγκατάσταση των υπαρχόντων servers αν υπάρχουν ή να τους απενεργοποιήσουμε με οποιοδήποτε άλλο τρόπο. Όπως είδαμε και πριν ο Apache είναι Open Source (ανοικτού κώδικα), έτσι μπορούμε να τον κατεβάσουμε από την διεύθυνση http://http.apache.org/download.cgi για όλα τα λειτουργικά συστήματα.
Όταν η λήψη του αρχείου ολοκληρωθεί κάνουμε διπλό κλικ πάνω του για να ξεκινήσουμε τον οδηγό εγκατάστασης (Installation Wizard) . Ανοίγει ένα παράθυρο που μας καλωσορίζει στην εγκατάσταση του Apache (εικόνα 1) όπου επιλέγουμε Next.



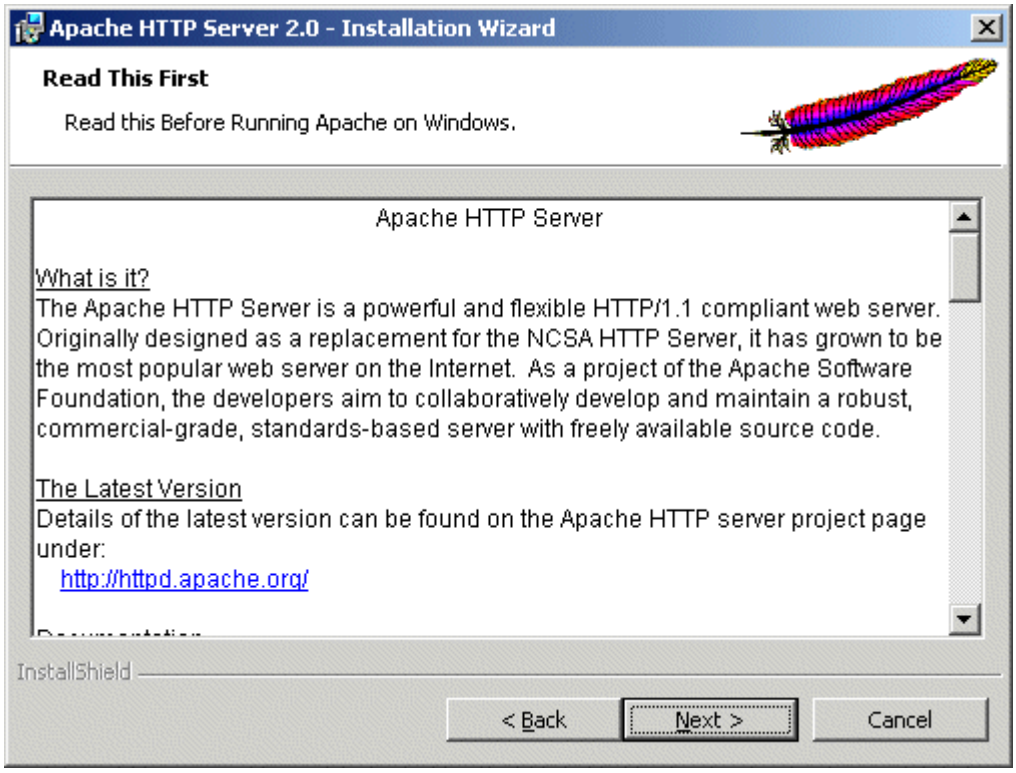
Εικόνα 1

Στην επόμενη καρτέλα θα διαβάσουμε και θα μας ζητηθεί να αποδεκτούμε την συμφωνία άδειας χρήσης του Apache (εικόνα 2). Στην ουσία η συμφωνία άδειας χρήσης αναφέρει ότι μπορούμε να κάνουμε ότι θέλουμε με το λογισμικό αλλά σε καμία περίπτωση δεν μπορούμε να ισχυριστούμε ότι εμείς το γράψαμε.



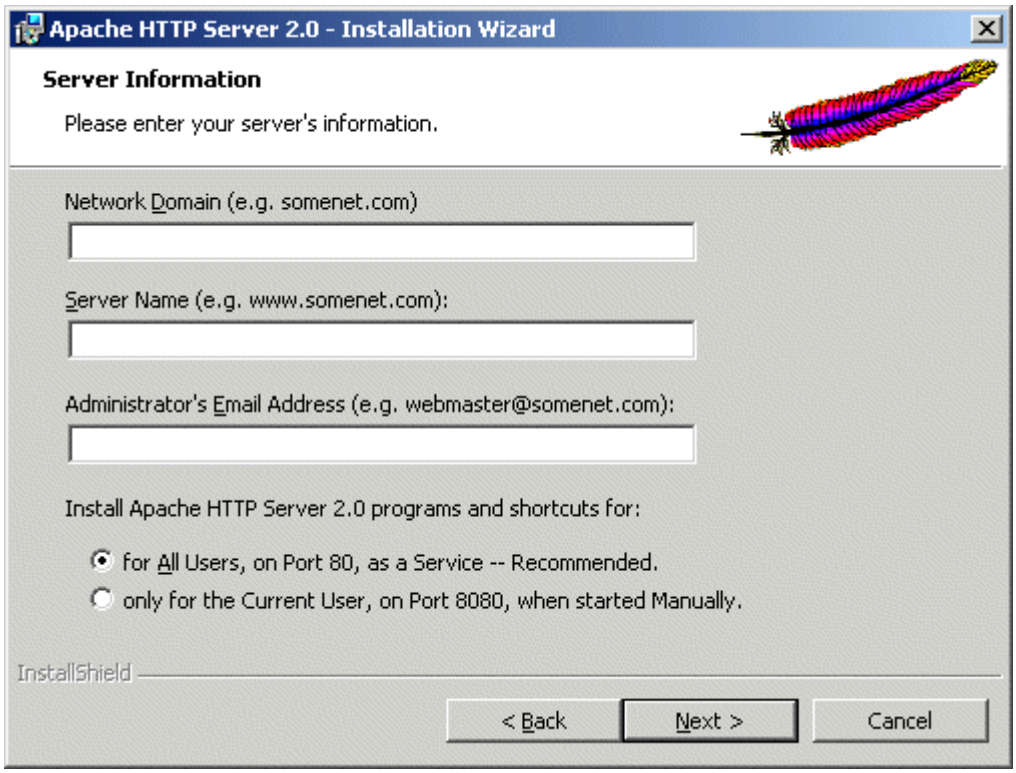
Εικόνα 2

Στην συνέχεια αφού αποδεχθούμε τους όρους της συμφωνίας άδειας χρήσης, ο οδηγός μας παρουσιάζει μία σύντομη εισαγωγή στο Apache (εικόνα 3).



Εικόνα 3

Ακολούθως ζητά να παρέχουμε βασικές πληροφορίες για τον υπολογιστή μας, όπως μπορούμε να δούμε στην εικόνα 4. Οι πληροφορίες αυτές είναι η πλήρης διεύθυνση δικτύου για τον server, το όνομα domain και όνομα του server (π.χ www.something.com), καθώς και η διεύθυνση ηλεκτρονικού ταχυδρομείου του επόπτη (administrator) του server. Οι client υπολογιστές για να προσπελάσουν τον server μας θα χρησιμοποιούν το όνομα του server και η διεύθυνση ηλεκτρονικού ταχυδρομείου του επόπτη θα προστίθεται στα μηνύματα σφάλματος έτσι ώστε οι επισκέπτες να ξέρουν πως να επικοινωνήσουν μαζί μας όταν προκύπτουν προβλήματα.



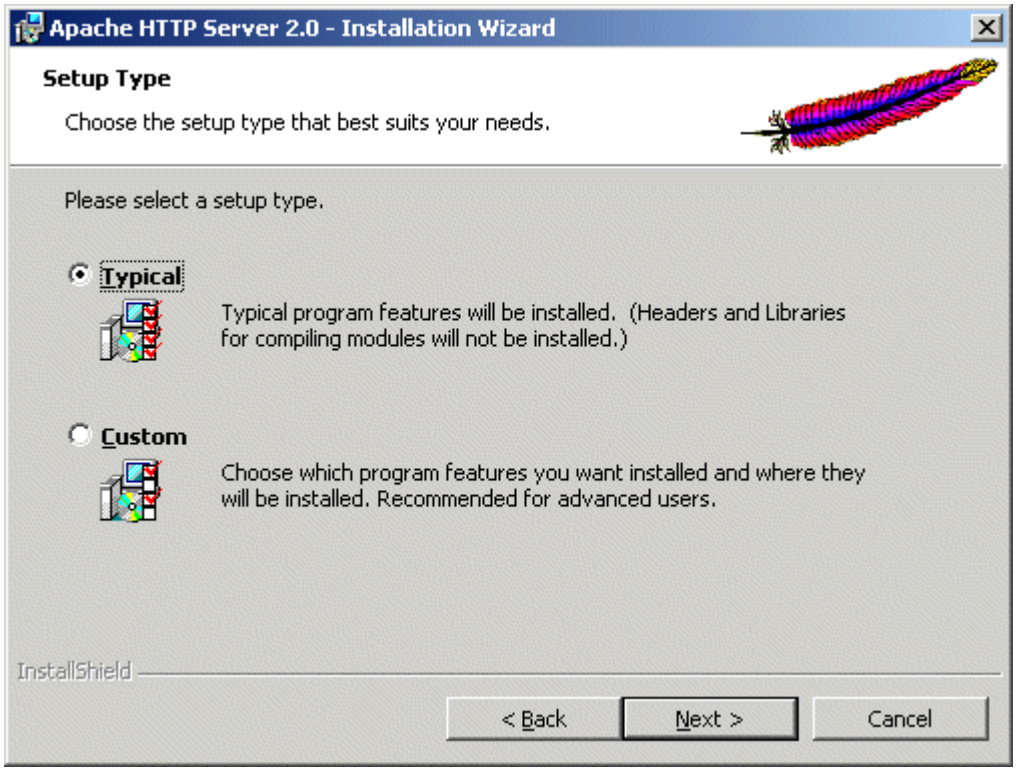
Εικόνα 4

Το πρόγραμμα εγκατάστασης τώρα ζητάει να επιλέξουμε εάν θα εγκαταστήσουμε το Apache σαν μία υπηρεσία (service) στο σύστημα ή εάν η εκκίνηση του θα γίνετε χειροκίνητα. Όταν γίνει εγκατάσταση του Apache σαν υπηρεσία ο server εκκινεί κάθε φορά που εκκινούν τα Windows και ο έλεγχος του μπορεί να γίνετε με τα γνωστά εργαλεία διαχείρισης υπηρεσιών που παρέχουν τα ίδια τα Windows. Αν έχουμε σκοπό να τρέχουμε τον Apache σε ένα περιβάλλον παραγωγής ή σε οποιοδήποτε άλλο περιβάλλον το οποίο απαιτεί την συνεχή λειτουργία του server τότε επιλέγουμε την μέθοδο εγκατάστασης σαν μία υπηρεσία.

Αν η εγκατάσταση του Apache γίνει για τον τρέχοντα χρήστη τότε θα πρέπει να ακολουθείτε η χειροκίνητη διαδικασία εκκίνησης και να ορίζουμε σαν προεπιλεγμένη θύρα (default port) την οποία το Apache ακροάζεται για αιτήσεις, την 8080. Επιλέγουμε την μέθοδο αυτή αν χρησιμοποιούμε το Apache για σκοπούς δοκιμών ή αν έχουμε ήδη έναν Web Server ο οποίος τρέχει στην θύρα 80.

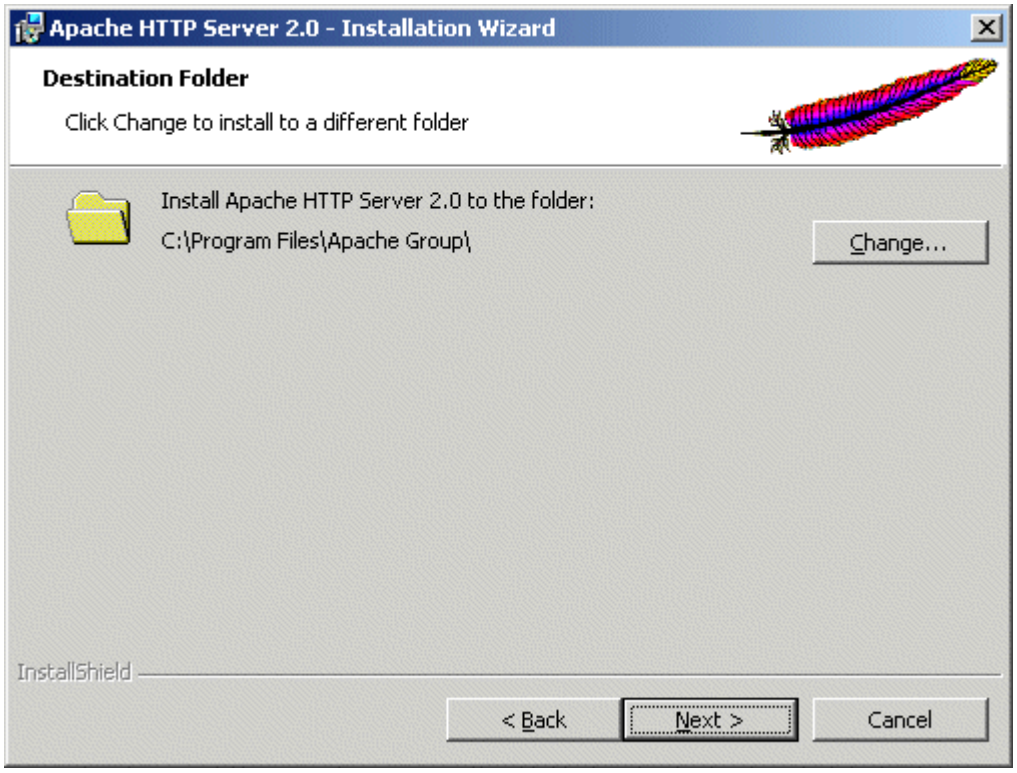
Η επόμενη οθόνη μας επιτρέπει να επιλέξουμε τον τύπο της εγκατάστασης όπως μπορούμε να δούμε στην (εικόνα 5). Με την τυπική (Typical) εγκατάσταση εγκαθίστανται τα δυαδικά αρχεία του Apache και τα αρχεία πληροφοριών τεκμηρίωσης, αλλά δεν εγκαθίστανται τα header αρχεία και οι βιβλιοθήκες. Η προτιμώμενη επιλογή είναι αυτή εκτός αν σκοπεύουμε να μεταγλωττίσουμε δικές μας ρουτίνες.

Η εξειδικευμένη (Custom) εγκατάσταση μας επιτρέπει να επιλέξουμε εάν θα εγκαταστήσουμε τα header αρχεία ή τα αρχεία πληροφοριών τεκμηρίωσης.



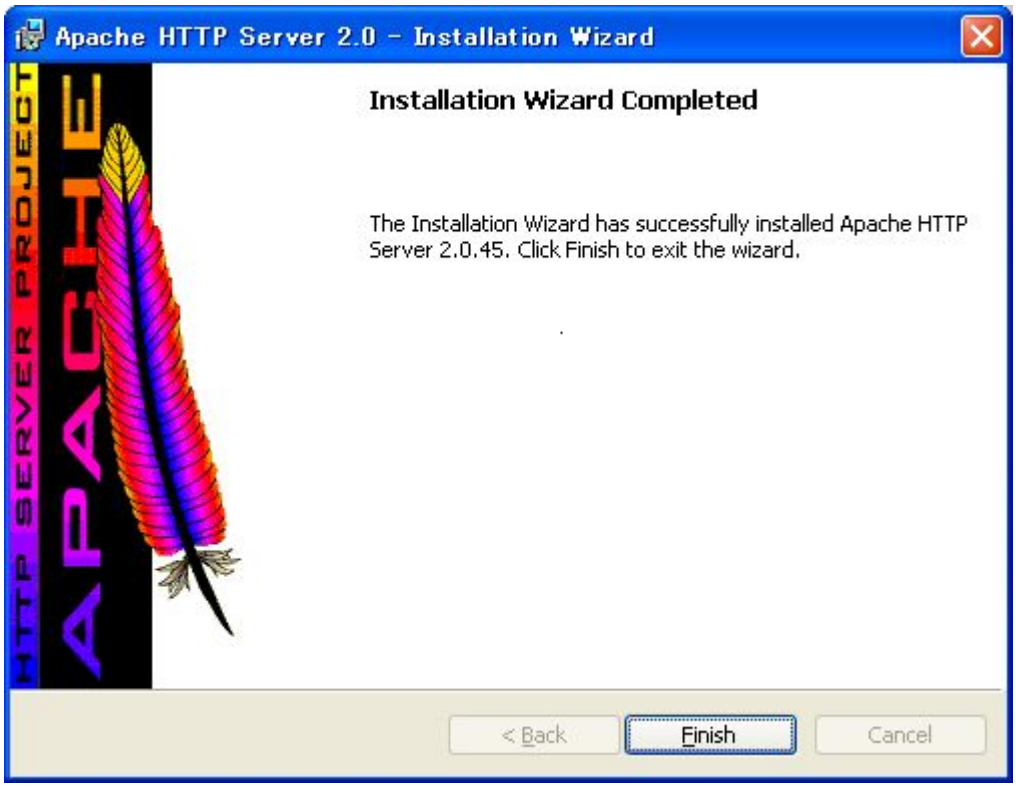
Εικόνα 5

Αφού επιλέξουμε τον κατάλογο εγκατάστασης (προτεινόμενος κατάλογος είναι ο : c:\Program Files\Apache Group), το πρόγραμμα θα προχωρήσει στην καθαυτό διαδικασία εγκατάστασης των αρχείων (εικόνα 6).



Εικόνα 6

Μόλις τελειώσει η εγκατάσταση και αν όλα πάνε καλά εμφανίζεται το τελευταίο παράθυρο το οποίο λέει ότι η εγκατάσταση έχει ολοκληρωθεί (εικόνα 7).



Εικόνα 7

Με το τέλος της εγκατάστασης κάτω δεξιά στο Taskbar βλέπουμε ένα καινούργιο εικονίδιο.

Τρέχοντας η εφαρμογή αυτή μας δείχνει την κατάσταση του server μας και μας δίνει την δυνατότητα να σταματήσουμε ή να αρχίσουμε την υπηρεσία. Για να διαπιστώσουμε την σωστή λειτουργία του Server μας δεν έχουμε παρά να ανοίξουμε τον internet explorer στην διεύθυνση http://localhost, ο explorer θα βγάλει μία σελίδα που θα μας ενημερώνει για την επιτυχία τις εγκατάστασης του Webserver μας.

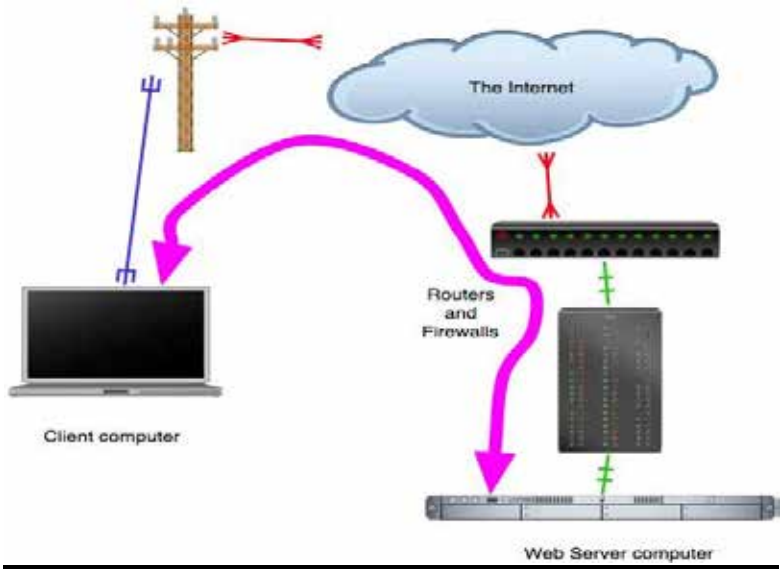
ΚΕΦΑΛΑΙΟ 3

ΔΙΑΧΕΙΡΙΣΗ

ΚΑΙ

ΔΙΑΜΟΡΦΩΣΗ

ΤΟΥ APACHE



3. Διαχείριση του Apache Server

Το εκτελέσιμο αρχείο για τον Apache Server ονομάζεται apache.exe. Δέχεται αρκετές επιλογές στην γραμμή εντολής, τις οποίες μπορούμε να δούμε στον πίνακα 1. Μπορούμε να εμφανίσουμε μία πλήρη λίστα των υποστηριζόμενων επιλογών εισάγοντας την εντολή apache.exe –h.

Πίνακας 1 Οι επιλογές του httpd

Επιλογή	Σημασία
-D	Μας επιτρέπει να περάσουμε μία παράμετρο η οποία μπορεί να χρησιμοποιηθεί για επεξεργασία από την ενότητα <IfDefine>
-I	Εμφανίζει μία λίστα των μεταγλωττισμένων modules
-v	Εμφανίζει τον αριθμό έκδοσης και τον χρόνο μεταγλώττισης του server
-f	Μας επιτρέπει να περάσουμε σαν παράμετρο την θέση του αρχείου httpd.conf, εάν είναι διαφορετική από την θέση που ίσχυε κατά τον χρόνο μεταγλώττισης

Στο Apache μπορούμε να στέλνουμε σήματα χρησιμοποιώντας το εκτελέσιμο αρχείο apache.exe:

- **apache.exe –k restart** – Επανεκκίνηση του Apache
- **apache.exe –k graceful** – Επανεκκίνηση χωρίς αρνητικές επιπτώσεις
- **apache.exe –k stop** – Τερματισμός της λειτουργίας του Apache

Μπορούμε να χρησιμοποιήσουμε τις συντομεύσεις που δημιούργησε γι'αυτές τις εντολές στο μενού Start ο οδηγός εγκατάστασης του Apache. Εάν εγκαταστήσαμε το Apache σαν μία υπηρεσία, μπορούμε να εκκινούμε ή να τερματίζουμε την λειτουργία του χρησιμοποιώντας τα εργαλεία που διαθέτουν τα ίδια τα Windows : από το Control Panel επιλέγουμε Administrative Tasks και κατόπιν κάνουμε κλικ στο εικονίδιο Services. Πιο κάτω θα δούμε πιο αναλυτικά τις εντολές που χρησιμοποιούνται για την διαχείριση του Apache είτε αυτό εγκαταστάθηκε σαν υπηρεσία είτε σαν εφαρμογή κονσόλας.

3.1 Τρέχοντας τον Apache ως υπηρεσία

Ο Apache μπορεί να τρέξει ως υπηρεσία στα WINDOWS. Μπορούμε να εγκαταστήσουμε των Apache ως υπηρεσία αυτόματα κατά τη διάρκεια της εγκατάστασης. Εάν επιλέξουμε να τον εγκαταστήσουμε για όλους τους χρήστες, η εγκατάσταση θα δημιουργήσει μια υπηρεσία Apache για μας. Εάν διευκρινίζουμε να γίνει η εγκατάσταση για μας μόνο, μπορούμε χειροκίνητα να καταχωρήσουμε των Apache ως υπηρεσία μετά από την εγκατάσταση. Για να πετύχει αυτό πρέπει να είμαστε μέλος της ομάδας διοικητών για την εγκατάσταση υπηρεσιών (Administrators group for the service installation).

Το Apache έρχεται με μια χρησιμότητα αποκαλούμενη όργανο ελέγχου υπηρεσιών Apache(Apache Service Monitor). Με αυτό μπορούμε να δούμε και να διαχειριστούμε την κατάσταση όλων των εγκατεστημένων υπηρεσιών Apache σε οποιαδήποτε μηχανή στο δίκτυό μας. Για να είμαστε σε θέση να διαχειριστούμε μια υπηρεσία Apache με το όργανο ελέγχου, πρέπει πρώτα να εγκαταστήσουμε την υπηρεσία (είτε αυτόματα μέσω της εγκατάστασης είτε χειρονακτικά). Μπορούμε να εγκαταστήσουμε τον Apache ως υπηρεσία των WINDOWS με την χρήση της εξής εντολής στο subdirectory bin του Apache:

apache - k install

Εάν πρέπει να διευκρινίσουμε το όνομα της υπηρεσίας που θέλουμε να εγκαταστήσουμε, χρησιμοποιούμε την ακόλουθη εντολή. Αυτό πρέπει να γίνει εάν έχουμε εγκατεστημένες διάφορες υπηρεσίες του Apache διαφορετικές μεταξύ τους στον υπολογιστή μας.

apache - k install –n "MyServiceName"

Εάν πρέπει να δώσουμε συγκεκριμένο όνομα στα αρχεία διαμόρφωσης για τις διαφορετικές υπηρεσίες, πρέπει να χρησιμοποιήσουμε την εντολή:

apache - k install –n "MyServiceName" -f "c:\files\my.conf"

Εάν χρησιμοποιήσουμε την πρώτη εντολή χωρίς οποιεσδήποτε ειδικές παραμέτρους εκτός από την εντολή **-k install** η υπηρεσία θα κληθεί Apache2 και το αρχείο διαμόρφωσης θα είναι το conf\httpd.conf.

Η αφαίρεση μιας υπηρεσίας Apache είναι εύκολη.
Απλά χρησιμοποιούμε την εντολή:

apache –k uninstall

Αν θέλουμε να απεγκαταστήσουμε μία συγκεκριμένη υπηρεσία Apache μπορούμε να την διευκρινίσουμε ως εξής:

apache –k uninstall -n "MyServiceName"

Η κανονική έναρξη, η επανεκκίνηση και η διακοπή μιας υπηρεσίας Apache γίνονται συνήθως μέσω του οργάνου ελέγχου υπηρεσιών Apache, με τη χρησιμοποίηση των εντολών **NET START Apache2** και **NET STOP Apache2** ή μέσω της κανονικής διαχείρισης υπηρεσιών των windows.

Πριν αρχίσουμε των Apache ως υπηρεσία με οποιοδήποτε μέσο, πρέπει να εξετάσουμε το αρχείο διαμόρφωσης των υπηρεσιών με την εντολή:

apache -n "MyServiceName" -t

Μπορούμε να ελέγξουμε μια υπηρεσία Apache επίσης και από τους διακόπτες στη γραμμή εντολών της.

Για να αρχίσει μια εγκατεστημένη υπηρεσία Apache θα χρησιμοποιήσουμε αυτό:

apache –k start

Για να σταματήσουμε μια υπηρεσία Apache μέσω των διακοπτών στη γραμμή εντολών, χρησιμοποιούμε αυτό:

apache –k stop ή apache –k shutdown

Μπορούμε επίσης να επανεκκινήσουμε μια τρέχον υπηρεσία και να την αναγκάσουμε για να ξαναδιαβάσει το αρχείο διαμόρφωσής της με την εντολή:

apache –k restart

Κατά έναρξη του Apache ως υπηρεσία μπορεί να αντιμετωπίσουμε ένα μήνυμα λάθους από το διευθυντή ελέγχου υπηρεσιών των windows(Windows Service Control Manager).
Παραδείγματος χάριν, εάν προσπαθούμε να αρχίσουμε τον Apache με τη χρησιμοποίηση των υπηρεσιών applet στο Control Panel των Windows, μπορεί να λάβουμε το ακόλουθο μήνυμα:

Could not start the Apache2 service on \\COMPUTER
Error 1067; The process terminated unexpectedly.

Θα πάρουμε αυτό το γενικό λάθος εάν υπάρχει οποιοδήποτε πρόβλημα με την έναρξη της υπηρεσίας Apache. Προκειμένου να δούμε τι προκαλεί πραγματικά το πρόβλημα πρέπει να ακολουθήσουμε τις οδηγίες για το τρέξιμο του Apache στα windows.

3.2 Τρέχοντας τον Apache ως εφαρμογή κονσόλας

Ο συνιστώμενος τρόπος να χρησιμοποιηθεί ο Apache συνήθως είναι σαν υπηρεσία αλλά μερικές φορές είναι ευκολότερο να δουλέψει από τη γραμμή εντολών.

Για να τρέξει ο Apache από τη γραμμή εντολών ως εφαρμογή κονσόλων χρησιμοποιούμε την ακόλουθη εντολή:
Apache

Ο Apache θα εκτελέσει και θα συνεχίσει να τρέχει έως ότου σταματήσει αφού πιέσουμε **Control-C.**
Μπορούμε επίσης να τρέξουμε τον Apache μέσω του εικονιδίου έναρξης του Apache στην κονσόλα που τοποθετείται στο **Start Menu --> Programs --> Apache HTTP Server 2.0.xx --> Control Apache Server** κατά τη διάρκεια της εγκατάστασης.

Αυτό θα ανοίξει μία κονσόλα των windows και μέσα σε αυτή θα αρχίσει ο Apache.

Εάν δεν εγκαταστήσαμε τον Apache ως υπηρεσία, το παράθυρο θα παραμείνει ορατό έως ότου το σταματήσουμε πιέζοντας Control-C στην κονσόλα όπου τρέχει ο Apache. Ο server θα βγει σε μερικά δευτερόλεπτα.
Εντούτοις, εάν εγκαταστήσαμε τον Apache ως υπηρεσία, το εικονίδιο αρχίζει την υπηρεσία. Εάν η υπηρεσία Apache τρέχει ήδη, το εικονίδιο δεν κάνει τίποτα.

Μπορούμε να πούμε στον Apache που τρέχει να σταματήσει με το άνοιγμα μίας άλλης κονσόλας των windows και δίνοντας την εντολή:
apache –k shutdown

Αυτό πρέπει να προτιμηθεί παρά τη συμπίεση του Control-C επειδή αυτό αφήνει τον Apache να τελειώσει οποιεσδήποτε τρέχουσες διαδικασίες και να καθαρίσει απόλυτα.

Μπορούμε επίσης να πούμε στον Apache να επανεκκίνηση. Αυτό τον αναγκάζει να ξαναδιαβάσει το αρχείο διαμόρφωσης. Οποιεσδήποτε διαδικασίες βρίσκονται υπό εξέλιξη επιτρέπεται να ολοκληρωθούν χωρίς διακοπή.
Για να επανεκκινήση ο Apache, χρησιμοποιούμε την εντολή:
apache –k restart

Εάν η κονσόλα του Apache στα windows κλείνει αμέσως ή απροσδόκητα μετά από το ξεκίνημα, ανοίγουμε την **Command Prompt από το Start Menu --> Programs**. Αλλάζουμε τον φάκελο στον οποίο εγκαταστήσαμε τον Apache, δακτυλογραφούμε την εντολή **apache** και διαβάζουμε το μήνυμα λάθους.

Κατόπιν γίνετε αλλαγή στο φάκελο log και αναθεωρούμε το αρχείο error.log για τα λάθη διαμόρφωσης. Εάν δεχτήκαμε τις προεπιλογές όταν εγκαταστήσαμε τον Apache, οι εντολές θα ήταν:

c:

cd "\Program Files\Apache Group\Apache2\bin"

apache

Στη συνέχεια περιμένουμε τον Apache να σταματήσει ή δίνουμε Control-C και εισάγουμε τα εξής:

cd ..\logs

more < error.log

Δουλεύοντας με Apache είναι σημαντικό να γνωρίζουμε πώς θα βρούμε το αρχείο διαμόρφωσης. Μπορούμε να διευκρινίσουμε ένα αρχείο διαμόρφωσης στη γραμμή εντολών με δύο τρόπους:

- το **–f** διευκρινίζει μια απόλυτη ή σχετική πορεία σε ένα ιδιαίτερο αρχείο διαμόρφωσης:

apache -f "c:\my server files\anotherconfig.conf"

ή

apache -f files \anotherconfig.conf

- το **-n** διευκρινίζει την εγκατεστημένη υπηρεσία Apache της οποίας το αρχείο διαμόρφωσης πρόκειται να χρησιμοποιηθεί:

apache -n "MyServiceName"

Εάν δεν διευκρινίζετε ένα αρχείο διαμόρφωσης με **–f** ή **-n** , ο Apache θα χρησιμοποιήσει το όνομα του αρχείου που συντάσσεται στον server, όπως conf\httpd.conf. Αυτή η ενσωματωμένη πορεία είναι σχετική με τον κατάλογο εγκατάστασης. Μπορούμε να ελέγξουμε το συνταγμένο όνομα αρχείων από μια αξία που χαρακτηρίζεται ως **SERVER_CONFIG_FILE** κατά την επίκληση του Apache με -v switch όπως:

apache –v

3.3 Εξετάζοντας την εγκατάσταση

Μετά την έναρξη του Apache (είτε σε κονσόλα παραθύρου είτε ως υπηρεσία) θα ακούει στη θύρα 80 (εκτός αν αλλάξαμε τη οδηγία Listen στα αρχεία διαμόρφωσης ή αν εγκαταστήσατε τον Apache μόνο για τον τρέχοντα χρήστη).
Για να συνδεθούμε στον server και να έχουμε πρόσβαση στη σελίδα προεπιλογής, προωθούμε έναν ξεφυλλιστή(browser) και εισάγουμε αυτό το URL:
http://localhost/

Το Apache πρέπει να αποκριθεί με μια σελίδα καλωσορίσματος και μια σύνδεση στο εγχειρίδιο του Apache. Εάν τίποτα δεν συμβεί ή παίρνουμε ένα λάθος, πρέπει να κοιτάξουμε το αρχείο error.log στο subdirectory logs. Εάν ο host μας δεν συνδέεται στο δίκτυο ή εάν έχουμε σοβαρά προβλήματα με την διαμόρφωση του dns μας (υπηρεσία ονόματος περιοχών) , μπορεί να πρέπει να χρησιμοποιήσουμε αυτό το URL:
http://127.0.0.1/

Εάν τυχαίνει να τρέχουμε τον Apache σε εναλλασσόμενη θύρα, πρέπει να βάλουμε σίγουρα το URL:
http://127.0.0.1:8080/

3.4 Εκκίνηση του Apache για Πρώτη Φορά

Πριν εκκινήσουμε το Apache θα πρέπει να βεβαιωθούμε ότι υπάρχουν οι ελάχιστες απαιτούμενες παράμετροι στο αρχείο διαμόρφωσης του Apache, httpd.conf .

3.5 Η Δομή του αρχείου Διαμόρφωσης του Apache

Το Apache διατηρεί όλες τις παραμέτρους διαμόρφωσης του σε αρχεία απλού κειμένου η οποία είναι η μόνη μορφή που κατανοεί το Apache. Το κύριο αρχείο παραμέτρων διαμόρφωσης ονομάζεται httpd.conf και μπορούμε να το ανοίξουμε και να τροποποιήσουμε χρησιμοποιώντας το Notepad ή το WordPad. Το αρχείο αυτό περιέχει ντιρεκτίβες (directives) και περιέκτες (containers), οι οποίοι μας δίνουν την δυνατότητα να προσαρμόσουμε την εγκατάσταση του Apache ανάλογα με τις ανάγκες μας. Οι ντιρεκτίβες διαμορφώνουν συγκεκριμένες ρυθμίσεις του Apache, όπως οι παράμετροι για τον έλεγχο πρόσβασης, την απόδοση και την λειτουργία στο δίκτυο. Οι περιέκτες καθορίζουν το πλαίσιο στο οποίο αναφέρονται οι ρυθμίσεις αυτές. Για παράδειγμα οι παράμετροι εξουσιοδότησης μπορούν να αναφέρονται στον server σαν σύνολο, σε έναν κατάλογο ή σε ένα μεμονωμένο αρχείο.

3.5.1 Ντιρεκτίβες

Η σύνταξη μιας ντιρεκτίβας του Apache υπόκειται στους ακόλουθους κανόνες:

- Τα ορίσματα τις ντιρεκτίβας ακολουθούν μετά από το όνομά της.
- Τα ορίσματα τις ντιρεκτίβας χωρίζονται μεταξύ τους με κενά διαστήματα.
- Ο αριθμός και ο τύπος των ορισμάτων διαφέρουν από ντιρεκτίβα σε ντιρεκτίβα· ορισμένες ντιρεκτίβες δεν έχουν ορίσματα.
- Μία ντιρεκτίβα καταλαμβάνει μία μεμονωμένη γραμμή στο αρχείο διαμόρφωσης, αλλά μπορούμε να την συνεχίσουμε σε επόμενη γραμμή τερματίζοντας την προηγούμενη με τον χαρακτήρα \ .
- Το σύμβολο # προηγείται τις ντιρεκτίβας και πρέπει να εμφανίζεται σε ξεχωριστή γραμμή.

Το σχήμα που ακολουθείται για την παρουσίαση των ντιρεκτίβων στην τεκμηρίωση του Apache είναι ίδιο για όλες τις ντιρεκτίβες:

- Syntax (σύνταξη) – Παρουσιάζει την σύνταξη της ντιρεκτίβας και όλες τις επιλογές της. Οι υποχρεωτικές παράμετροι αναγράφονται με πλάγια γραφή, ενώ οι προαιρετικές παράμετροι αναγράφονται με πλάγια γραφή και περικλείονται σε αγκύλες.
- Default (προεπιλεγμένη μνήμη) – Εάν μία ντιρεκτίβα έχει προεπιλεγμένη τιμή, αυτή θα αναφέρεται εδώ.
- Context (πλαίσιο) – Η καταχώρηση Αυτή υποδεικνύει εάν η ντιρεκτίβα είναι μία εγγενής ντιρεκτίβα του Apache ή ανήκει σε κάποιο από τα προγράμματα που το συνοδεύουν ή είναι μέρος ενός προγράμματος Multi-Processing Module (MPM) ή περιλαμβάνεται στο πακέτο του Apache αλλά δεν είναι έτοιμη για χρήση σε έναν server παραγωγής.
- Module (πρόγραμμα) – Η καταχώρηση αυτή υποδεικνύει το πρόγραμμα ή την ρουτίνα στην οποία ανήκει η ντιρεκτίβα.
- Compatibility (συμβατότητα) – Η καταχώρηση αυτή περιέχει πληροφορίες σχετικά με τις εκδόσεις του Apache που υποστηρίζουν την συγκεκριμένη ντιρεκτίβα.
- Override (προτεραιότητα) – Οι ντιρεκτίβες του Apache ταξινομούνται σε διαφορετικές κατηγορίες. Το πεδίο override χρησιμοποιείται για να καθορίσει ποιες κατηγορίες ντιρεκτίβων μπορούν να εμφανίζονται στα αρχεία διαμόρφωσης .htaccess ανά κατάλογο.

Μετά από τις παραπάνω καταχωρήσεις ακολουθεί μία συνοπτική επεξήγηση τις ντιρεκτίβας και μπορεί επίσης να περιλαμβάνεται μία παραπομπή προς σχετιζόμενες ντιρεκτίβες ή άλλες πληροφορίες.

3.5.2 Περιέκτες (Containers)

Οι περιέκτες ντιρεκτίβων (directive containers), οι οποίοι αποκαλούνται επίσης ενότητες (sections), περιορίζουν το πεδίο δράσης των ντιρεκτίβων. Εάν οι ντιρεκτίβες δεν βρίσκονται μέσα σε έναν περιέκτη, θεωρείται ότι ανήκουν στο προκαθορισμένο πεδίο δράσης του server (server config) και εφαρμόζονται στον server σαν σύνολο.

Οι προκαθορισμένοι περιέκτες του Apache είναι:

- <VirtualHost> - Καθορίζει έναν εικονικό server. Μέσω των εικονικών servers έχουμε την δυνατότητα να στεγάσουμε πολλαπλά Web Sites στην ίδια εγκατάσταση του Apache. Οι ντιρεκτίβες που περιλαμβάνει αυτός ο περιέκτης εφαρμόζονται σε ένα συγκεκριμένο Web Site. Μία τέτοια ντιρεκτίβα δέχεται σαν όρισμα ένα όνομα domain ή μία διεύθυνση IP και προαιρετικά έναν αριθμό θύρας. Στην συνέχεια θα πούμε περισσότερα για τους εικονικούς host.

- <Directory>, <DirectoryMatch> - Αυτοί οι περιέκτες επιτρέπουν στις ντιρεκτίβες να εφαρμόζονται σε έναν συγκεκριμένο κατάλογο ή σε μία ομάδα καταλόγων του συστήματος αρχείων. Οι περιέκτες Directory δέχονται σαν όρισμα ένα όνομα καταλόγου ή ένα μοτίβο επιλογής καταλόγων. Οι ντιρεκτίβες που περιλαμβάνουν εφαρμόζονται στους προσδιοριζόμενους καταλόγους και στους υποκαταλόγους τους. Ο περιέκτης DirectoryMatch δέχεται υποδειγματικές εκφράσεις σαν όρισμα. Η ακόλουθη ντιρεκτίβα για παράδειγμα εντοπίζει όλους τους υποκαταλόγους του καταλόγου www, των οποίων το όνομα αποτελείται από τέσσερις αριθμούς:
  <DirectoryMatch "^/www/.*/[0-9] {4} ">

- <Location>, <LocationMatch> - Αυτοί οι περιέκτες επιτρέπουν στις ντιρεκτίβες να εφαρμόζονται σε συγκεκριμένες διευθύνσεις URL ή ομάδες διευθύνσεων URL. Το σκεπτικό στο οποίο βασίζονται είναι παρόμοιο με αυτό στο οποίο βασίζονται οι αντίστοιχοι περιέκτες Directory.
  Ο περιέκτης LocationMatch δέχεται σαν όρισμα μία υποδειγματική έκφραση. Η ακόλουθη εντολή για παράδειγμα εντοπίζει τους καταλόγους που περιέχουν είτε το "/my/data", είτε το "/your/data":
  < LocationMatch "/ (my| your) / data ">

- <Files>, <FilesMatch> - Είναι παρόμοιοι με τους περιέκτες Directory και Location. Οι ενότητες Files επιτρέπουν στις ντιρεκτίβες να εφαρμόζονται σε συγκεκριμένα αρχεία ή ομάδες αρχείων.

3.6 Διαμόρφωση του Apache

Οι webserver γενικά είναι ευκολόχρηστα προγράμματα. Παρακάτω θα δούμε την διαμόρφωση του Apache Web server, η οποία δεν είναι τίποτα άλλο όπως αναφέραμε και πριν παρά ένα αρχείο κειμένου το οποίο ο Apache διαβάζει κατά την εκκίνησή του και διαμορφώνετε ανάλογα με τις ρυθμίσεις που έχουμε γράψει μέσα στο αρχείο. Αφού έχουμε κάνει την εγκατάσταση ο web server μας είναι εγκατεστημένος στον φάκελο C:\Program Files\Apache Group\Apache2 το httpd.conf (έτσι ονομάζετε το αρχείο διαμόρφωσης) που βρίσκετε στον υποφάκελο conf\ . Για να ενεργοποιηθεί κάποια αλλαγή που κάναμε πρέπει να επανεκκινήσουμε τον server από την εφαρμογή που είδαμε πιο πάνω στην γραμμή εργασιών. Ανοίγουμε λοιπόν το αρχείο με τον Text editor και ξεκινάμε.
Το αρχείο αυτό χωρίζεται σε 3 κομμάτια το α) Global Enviroment β) main server configuration γ) Virtual Hosts. Κάθε γραμμή του αρχείου που ξεκινάει με ένα # δεν διαβάζετε από τον server, θεωρείται ως σημείωση. Παρακάτω αναλύονται ένα
ένα τα τρία κομμάτια τις διαμόρφωσης.

3.6.1 Global Enviroment

Εδώ δηλώνουμε τις βασικές ρυθμίσεις του Server που αφορούν την λειτουργία της υπηρεσίας όπως την τοποθεσία που βρίσκονται τα αρχεία διαμόρφωσης του ή τον αριθμό τον συνδεδεμένων χρηστών που θα εξυπηρετεί.

Ας περάσουμε αναλυτικά στις επιμέρους ρυθμίσεις του «Global Enviroment». Το βασικότερο από όλα είναι ο προσδιορισμός της θέσης των αρχείων της εγκατάστασης δίνουμε λοιπόν την πρώτη μας γραμμή στο αρχείο. (εφόσον δεν έχουμε αλλάξει τον φάκελο εγκατάστασης ο φάκελος θα είναι ο "D:/Program Files/Apache Group/Apache2"). Ο Apache χρησιμοποιεί ένα αρχείο για να συντονίζει τις διαδικασίες που εκτελούνται στον υπολογιστή μας και έχει δημιουργήσει ο ίδιος.

Προεπιλεγμένα, δοκιμάζει να καταλάβει κάποιες θέσεις στην μνήμη για να αποθηκεύσει τα απαραίτητα δεδομένα. Αν αποτύχει δημιουργεί ένα αρχείο στον δίσκο μας. Με την παρακάτω γραμμή αναγκάζουμε τον Apache να χρησιμοποιεί συνέχεια αρχείο για την εγράφη των πληροφοριών. Η επιλογή αυτή χρειάζεται στην περίπτωση που θέλουμε να επέμβουμε σε αυτό το αρχείο με κάποια εφαρμογή. Προσοχή το αρχείο αυτό πρέπει να είναι πάντα στους τοπικούς δίσκους μας και όχι σε κάποιο φάκελο δικτύου.

#ScoreBoardFile logs/apache_runtime_statuser

Στο επόμενο βήμα ορίζουμε το αρχείο το οποίο θα αποθηκεύει το Id της διεργασίας του Server με άλλα λόγια μας δίνει τον χαρακτηριστικό αριθμό που παίρνει από το λειτουργικό σύστημα κάθε φορά που ξεκινάει. Με αυτό τον αριθμό μπορούμε να τερματίσουμε την λειτουργία της ή να πάρουμε πληροφορίες για αυτήν.

PidFile logs/httpd.pid

Ορισμός του TimeOut δηλαδή ο συνολικός χρόνος που θα περιμένει ο Server από την στιγμή που θα πραγματοποιηθεί μία σύνδεση μέχρι να λάβεί την εντολή GET.

Εξ ορισμού είναι 300 δευτερόλεπτα.

Timeout 300

Μια επιλογή η οποία επηρεάζει πάρα πολύ την ταχύτητα φόρτωσης των σελίδων (όταν αυτά έχουν πολλές εικόνες μπορεί να παρατηρηθεί αύξηση μέχρι 50% της ταχύτητας φόρτωσης) είναι η Keep alive με την ενεργοποίηση της δυνατότητας αυτής έχουμε την δυνατότητα από μία σύνδεση TCP που θα γίνει με τον server να εξυπηρετήσουμε πολλές αιτήσεις. π.χ αν θέλουμε να ανοίξουμε μία σελίδα που περιέχει ένα Ά κείμενο και αρκετές εικόνες θα χρειαστεί να δημιουργήσουμε μία σύνδεση για να πάρουμε το κείμενο και τόσες συνδέσεις όσες είναι οι εικόνες του κειμένου. Με το Keep Alive ενεργοποιημένο ο Server μπορεί να εξυπηρετήσει όλες τις αιτήσεις (για το κείμενο και την κάθε εικόνα) μέσα από μία σύνδεση με αποτέλεσμα να μειώνεται κατά πολύ ο χρόνος που απαιτείται για να φορτωθεί όλο το έγγραφο.

KeepAlive On

Επιπλέον ρυθμίσεις που αφορούν το keepalive είναι το MaxKeepAliveRequests με το οποίο ορίζουμε τον μέγιστο αριθμό τον αιτήσεων που θα εξυπηρετούνται από μία σύνδεση. Αν ορίσουμε 0 ο αριθμός τον αιτήσεων είναι απεριόριστος. Καλό θα ήταν όμως να έχουμε προσδιορίσει ένα νούμερο (αρκετά μεγάλο όπως αυτό του παραδείγματος) για να έχουμε καλές αποδόσεις.

MaxKeepAliveRequests 100

Εκτός από το πλίθος των αιτήσεων πρέπει να ορίσουμε και το χρονικό διάστημα που ο server θα κρατάει την σύνδεση ανοιχτή περιμένοντας μία άλλη αίτηση.

KeepAliveTimeout 15

Στο Global Enviroment ορίζουμε επίσης και το νούμερο των threads τα οποία ο apache αφήνει για την κάθε διεργασία που προέρχεται από αυτόν, όπως επίσης και το νούμερο των αιτήσεων που θα δέχεται ο server για κάθε διεργασία του.

<IfModule mpm_winnt.c>
ThreadsPerChild 250
MaxRequestsPerChild 0
</IfModule>

Αν ο υπολογιστής μας έχει πολλές IP και δεν θέλουμε να απαντάει σε όλες τις IP μας παρέχει την δυνατότητα να ορίσουμε σε ποιες IP και σε ποιες ports θέλουμε να παίζει. Αν θέλουμε να παίζει σε όλες τις IP μπορούμε να δηλώσουμε μόνο την Port

Listen 80

Αν πάλι θέλουμε να περιορίσουμε τις δυνατότητες του σε κάποιο συγκεκριμένο/α IP δεν έχουμε παρά να δηλώσουμε την IP του και την Port που θέλουμε να παίζει.

Listen 12.34.56.78:80

Τελειώνοντας με το Global Enviroment ακολουθεί μία λίστα με τα Modules(φόρμουλες) που είναι προεγκατεστημένα στον Apache καλό θα ήταν να μην πειράξουμε κάποια από αυτά γιατί μπορεί να επηρεάσουν την σωστή λειτουργία του Server αργότερα θα δούμε πώς μπορούμε να προσθέσουμε και αλλά modules.

LoadModule access_module modules/mod_access.so

LoadModule actions_module modules/mod_actions.so

LoadModule alias_module modules/mod_alias.so

LoadModule asis_module modules/mod_asis.so

LoadModule auth_module modules/mod_auth.so

#LoadModule auth_anon_module modules/mod_auth_anon.so

#LoadModule auth_dbm_module modules/mod_auth_dbm.so

#LoadModule auth_digest_module modules/mod_auth_digest.so

LoadModule autoindex_module modules/mod_autoindex.so

#LoadModule cern_meta_module modules/mod_cern_meta.so

LoadModule cgi_module modules/mod_cgi.so

#LoadModule dav_module modules/mod_dav.so

#LoadModule dav_fs_module modules/mod_dav_fs.so

LoadModule dir_module modules/mod_dir.so

LoadModule env_module modules/mod_env.so

#LoadModule expires_module modules/mod_expires.so

#LoadModule file_cache_module modules/mod_file_cache.so

#LoadModule headers_module modules/mod_headers.so

LoadModule imap_module modules/mod_imap.so

LoadModule include_module modules/mod_include.so

#LoadModule info_module modules/mod_info.so

LoadModule isapi_module modules/mod_isapi.so

LoadModule log_config_module modules/mod_log_config.so

LoadModule mime_module modules/mod_mime.so

#LoadModule mime_magic_module modules/mod_mime_magic.so

#LoadModule proxy_module modules/mod_proxy.so

#LoadModule proxy_connect_module modules/mod_proxy_connect.so

#LoadModule proxy_http_module modules/mod_proxy_http.so

#LoadModule proxy_ftp_module modules/mod_proxy_ftp.so

LoadModule negotiation_module modules/mod_negotiation.so

3.6.2 Διαμόρφωση του 'Main' server

Ο Apache έχει την δυνατότητα να δημιουργεί πολλούς εικονικούς Webservers (εξ ου και το όνομα του τρίτου μέρους της διαμόρφωσης 'Virtual Hosts') στο σημείο αυτό διαμορφώνουμε τον Κεντρικό Server. Η πρώτη ρύθμιση είναι το e-mail του administrator του server προφανώς εδώ βάζουμε το email μας. Ο κεντρικός server πρέπει να έχει ένα όνομα οπωσδήποτε αν επιθυμούμε να έχουμε και άλλους δευτερεύοντες server σε μία IP. Αν ο μόνος server που υπάρχει είναι ο κεντρικός δεν έχει μεγάλη σημασία αν έχουμε κάποιο όνομα, μπορούμε να ορίσουμε ότι θέλουμε. Για παράδειγμα στην περίπτωση που ο server έχει το όνομα www.something.com:80 το 80 είναι η port που παίζει. Προεπιλεγμένα όταν ανοίγουμε μία διεύθυνση με τον Browser αυτόματα ο Browser προσπαθεί να επικοινωνήσει με την port 80 αν το αλλάξουμε θα πρέπει να ορίζουμε και το Port στον Browser
(π.χ. http://www.something.com:99)

ServerName www.something.com:80

Ήρθε η στιγμή που πρέπει να δηλώσουμε τον φάκελο του οποίου τα αρχεία θα προβάλει ο Apache.

DocumentRoot "D:/Program Files/Apache Group/Apache2/htdocs"

Αφού ορίσαμε τον φάκελο που θα προβάλει ο Apache θα πρέπει να δηλώσουμε και το αρχείο το οποίο θα ανοίγει προεπιλεγμένα (εφόσον αυτό υπάρχει στον φάκελο) π.χ. όταν πηγαίνουμε σε μία διεύθυνση (www.something.com αντί να βλέπουμε τον φάκελο με όλα τα αρχεία βλέπουμε το www.something.com/index.html) τα αρχεία που συνηθίζονται να είναι τα «προεπιλεγμένα» είναι τα εξής index.html index.htm index.php default.htm

DirectoryIndex index.htm index.htm index.php default.htm

Κάθε αρχείο το οποίο έχουμε κάνει αίτηση για να πάρουμε από τον server αποστέλλετε στον Browser ανάλογα με το περιεχόμενο του έτσι ώστε να ανοίξει με την κατάλληλη εφαρμογή. Το αρχείο Mime.types περιέχει μία λίστα με όλα τα προγράμματα και την επέκταση που αντιστοιχεί στο κάθε ένα. Μπορούμε ανά πάσα στιγμή να αλλάξουμε την ιδιότητα της κάθε επέκτασης. Παρακάτω λοιπόν δηλώνουμε την ακριβή θέση του αρχείου.

TypesConfig conf/mime.types

Τα αρχεία με επεκτάσεις που δεν αναφέρονται στην λίστα θα πρέπει να οριστούν τι τύπου θα είναι. Η προεπιλεγμένη επιλογή είναι να φαίνονται σαν αρχεία κειμένου. Αν θέλουμε να την αλλάξουμε μπορούμε να δούμε τους τύπους τον δεδομένων που υπάρχουν στο αρχείο Mime.types

DefaultType text/plain

HostnameLookups: με την επιλογή αυτή όταν ένας χρήστης επισκεφθεί το site ο Webserver δεν κρατάει την IP του αλλά προσπαθεί να βρει το domain που αντιστοιχεί στην συγκεκριμένη IP αν το επιτύχει κρατάει το domain στα log files του. Προεπιλεγμένα η επιλογή αυτή είναι απενεργοποιημένη γιατί η διαδικασία εύρεσης του domain καταναλώνει το bandwidth της γραμμής μας.
HostnameLookups Off

Επόμενο βήμα είναι η διαμόρφωση τον Log files. Πρώτα από όλα θα ορίσουμε το error.log το αρχείο που θα αναφέρει όλα τα λάθη που έχουν εμφανιστεί στον Server κατά την λειτουργία του. Καλό θα ήταν να το συμβουλευόμαστε συχνά όταν κάτι δεν πάει καλά. Δηλώνουμε λοιπόν την θέση του.

ErrorLog logs/error.log

Το error log έχει 8 επίπεδα ο πίνακας παρακάτω δείχνει αναλυτικά το κάθε επίπεδο και τα λάθη που περιγράφει το κάθε ένα.

Επίπεδο	**Περιγραφή**	**Παράδειγμα**
Emerg	Πολύ επείγοντα προβλήματα που καθιστούν αδύνατη την λειτουργία του web server	*"Child cannot open lock file. Exiting"*
Alert	Προβλήματα που χρειάζονται άμεση λύση	*"getpwuid: couldn't determine user name from uid"*
Crit	Σοβαρά σφάλματα κατά την λειτουργία του web server	*"socket: Failed to get a socket, exiting child"*
Error	Λάθη κατά την λειτουργία	*"Premature end of script headers"*
Warn	Λάθη που δεν χρειάζονται μεγάλη προσοχή	*"child process 1234 did not exit, sending another SIGHUP"*
Notice	Κανονική λειτουργία με ορολογία	*"httpd: caught SIGBUS, attempting to dump core in"*
Info	Παρέχει πληροφορίες	*"Server seems busy, (you may need to increase StartServers, or Min/MaxSpareServers)..."*
debug	Σε περίπτωση γενικού σφάλματος πλήρη ενημέρωση των Log files με όλες τις κινήσεις του server	*"Opening config file ..."*

Όπως θα παρατηρήσατε όσο κατεβαίνουμε στα επίπεδα τόσο πιο αναλυτικές περιγραφές γίνονται . Όταν έχουμε κάποιο πρόβλημα στην λειτουργία του server δεν έχουμε παρά να κατέβουμε σε ένα επίπεδο που μπορούμε να εντοπίσουμε το πρόβλημα μας.

LogLevel warn

(στο warn βάζουμε ένα από τα επίπεδα μας καλό θα είναι να μην είναι από τα πιο αναλυτικά γιατί μετά τα log files μεγαλώνουν πολύ)

Το επόμενο Log που θα εξετάσουμε είναι το Access_log το οποίο καταχωρεί όλες τις προσπελάσεις των αρχείων που δίνει ο web server στους χρήστες. Η πρώτη γραμμή ορίζει ποιες πληροφορίες θα φαίνονται στο Log και η δεύτερη την θέση που θα αποθηκεύεται το αρχείο

LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined CustomLog logs/access.log common

Ενεργοποιώντας την αυτόματη αναγνώριση γλώσσας του χρήστη

Το πρώτο πράγμα που πρέπει να δηλώσουμε είναι ποια θα είναι η προεπιλεγμένη γλώσσα.

DefaultLanguage en

Ακολουθεί μία λίστα με όλες τις γλώσσες που θα υποστηρίζει ο server μας

AddLanguage fr .fr

AddLanguage de .de
AddLanguage el .el
AddLanguage en .en

Κάθε φορά που κάποιος Browser επισκέπτεται τον server μας ανάλογα με την γλώσσα του ο server επιλέγει ανάλογα, αν είναι ελληνική το αρχείο ονομάζετε index.htm.el αν είναι γερμανική το index.htm.de αν είναι από κάποια χώρα που δεν υποστηρίζετε η γλώσσα του τότε αυτόματα επιλέγετε η αγγλική σελίδα. Τελειώνοντας με τις ρυθμίσεις της κωδικοσελίδας και το δεύτερο μέρος της διαμόρφωσης πρέπει να ορίσουμε σαν Default την ελληνική κωδικοσελίδα αν φυσικά γράφουμε ελληνικά.

AddDefaultCharset ISO-8859-7

3.6.3 Virtual Hosts

Οι εικονικοί server όπως αναφέραμε και στην αρχή μπορούν να λειτουργούν με δυο διαφορετικούς τρόπους. **IP-based Virtual Hosts** που σε αυτή την περίπτωση πρέπει να έχουμε στην διάθεση μας πολλά IP address ένα για τον κάθε host έτσι ο apache μπορεί να καταλάβει πια σελίδα θα προβάλει συνήθως, επειδή όμως αυτό δεν είναι εφικτό χρησιμοποιούμε τον δεύτερο τρόπο που είναι ο **Name-based Virtual Hosts.**

Το πρώτο πράγμα λοιπόν που πρέπει να δηλώσουμε είναι η IP του κάθε Vhost

NameVirtualHost 195.242.133.107

Μετά από την δήλωση του vhost πρέπει να δούμε τις παραμέτρους του. Δηλώνοντας ξανά σε ποία IP αναφερόμαστε δίνουμε τις παραμέτρους που είναι οι ίδιες με αυτές που έχουμε στη διαμόρφωση του main server.

VirtualHost 111.22.33.44 - δήλωση του vhost που αναφερόμαστε (Αρχή)

ServerName www.domain.gr - το domain του vhost

DocumentRoot "D:/wed/domain " - ο φάκελος του vhost

</VirtualHost> - Τέλος παραμετροποίησης του Vhost

Name-based Virtual Hosts

Όπως φαίνεται και στο όνομα ο κάθε Virtual host έχει διαφορετικό όνομα (domain) έτσι, όταν φτάνει μία αίτηση ο apache κοιτάζει την διεύθυνση που έχει πληκτρολογήσει αν αφορά κάποιον από τους Virtual hosts τότε εμφανίζει την σελίδα αυτού που τον αφορά.

Αλλιώς αν δεν βρεθεί πουθενά κάποια αντίστοιχη διεύθυνση παρουσιάζει την σελίδα του “main server “.

Στην περίπτωση αυτή η IP είναι πάντα ίδια για αυτό δηλώνουμε στο πρώτο πεδίο που στην προηγούμενη περίπτωση βάλαμε την IP του Vhost ένα «αστεράκι * » για να δηλώσουμε ότι δεν υπάρχει διάκριση στις IP.

NameVirtualHost *

Ακολουθεί ένα παράδειγμα με 2Vhost Που έχουν διαφορετικά domain το www.c-lab.gr και το www.pcstuff.gr

<VirtualHost *> Αρχή του πρώτου Vhost

ServerName www.c-lab.gr - το domain του vhost με βάση το οποίο θα γίνετε ο διαχωρισμός

DocumentRoot "D:/wed/c-lab" - ο φάκελος του vhost

</VirtualHost> - Τέλος πρώτου Vhost

<VirtualHost *> - Αρχή του δεύτερου Vhost

ServerName www.pcstaff.gr - το domain του vhost με βάση το οποίο θα γίνετε ο διαχωρισμός

DocumentRoot "D:/wed/pc " - ο φάκελος του vhost

</VirtualHost> - Τέλος δεύτερου Vhost

Να μην ξεχνάμε ότι και στις δυο περιπτώσεις (ip based , name based) έχουμε την δυνατότητα (μέσα στον κάθε Vhost) να βάλουμε οποιαδήποτε από τις ρυθμίσεις του main server.

3.7 Δυνατότητες του Apache Server

Ο Apache Server είναι ένα λογισμικό ανοικτού κώδικα (open course) όπως αναφέραμε και στην αρχή με αποτέλεσμα την ελεύθερη διάδοση του και την δυνατότητα νόμιμης αντιγραφής του. Έχει την δυνατότητα να δέχεται τα λεγόμενα Modules τα οποία είναι εφαρμογές που προστίθενται στον web server και μας προσφέρουν επιπλέον δυνατότητες. Όταν κατεβάζουμε τον Apache κάποια από τα modules είναι ήδη εγκατεστημένα. Πιο κάτω βλέπουμε τα πιο βασικά modules του Apache τα οποία τον χαρακτηρίζουν:

DBM databases for authentication (mod_auth_dbm)

Παρέχει στους χρήστες σελίδες προστατευμένες με κωδικό, με ευκολία χρήσης και υποστήριξη πολλών χρηστών χωρίς να επιβαρύνουν την λειτουργία του Server

Παραμετροποιημένες απαντήσεις σε προβλήματα που παρουσιάζονται (core)

Παρέχει την δυνατότητα δημιουργίας αρχείων τα οποία εμφανίζονται σε περίπτωση προβλήματος και παρουσιάζονται στον χρήστη έτσι ώστε να μπορέσει να εντοπίσει το λάθος.

Αναγνώριση πολλαπλών Directory Index (mod_dir)

Δυνατότητα ορισμού του αρχείου που θα προβάλλετε όταν ο χρήστης ζητάει ένα Url το οποίο παραπέμπει σε φάκελο π.χ. το www.pcstuff.gr/test/ να ανοίγει το αρχείο που έχουμε ορίσει (index.html ή οποιοδήποτε άλλο όνομα)

Απεριόριστες και ευέλικτος χειρισμός τον διευθύνσεων (url) (mod_rewrite)

Ο Apache δεν έχει όριο στις αναδρομολογίσεις των διευθύνσεων. Παρέχει μία ευέλικτη μηχανή η οποία επιτρέπει στον χρήστη να αναδρομολογίση την διεύθυνση οποιασδήποτε σελίδας σε κάποια άλλη.

Αναγνώριση περιεχομένου (mod_negotiation)

Ο Apache μπορεί αυτόματα να επιλέγει την γλώσσα ή τον τύπο των δεδομένων που μπορεί να δεχθεί ο πελάτης και ανάλογα να στέλνει τα δεδομένα στην κατάλληλη γλώσσα ή αντίστοιχα τον κατάλληλο τύπο δεδομένων

Εικονικοί Server (Virtual Hosts) (core)

Ένα από τα χαρακτηριστικά του Apache το οποίο χρησιμοποιείται πάρα πολύ. Ο server μας με το συγκεκριμένο χαρακτηριστικό μπορεί να ξεχωρίζει τις αιτήσεις που αφορούν διαφορετικό IP ή από διαφορετικό Domain δημιουργώντας τους έναν εικονικό server που τους εξυπηρετεί. Έτσι, από ένα Server μπορούμε να εξυπηρετούμε πολλά websites .

Πλήρως Παραμετροποιήσιμα Log files (mod_log_config) (mod_setenvif)

Τα αρχεία αναφοράς του apache παραμετροποιούνται και δημιουργούνται με την μορφή που θέλουμε εμείς. Μπορούμε να έχουμε ξεχωριστό Log για τον κάθε Vhost, δυνατότητα φιλτραρίσματος των δεδομένων που θα εμφανιστούν στο αρχείο, αναγνώρισης του hostname του κάθε επισκέπτη όπως και αυτόματη αλλαγή (rotate) των logs όταν περάσει κάποιο χρονικό όριο.

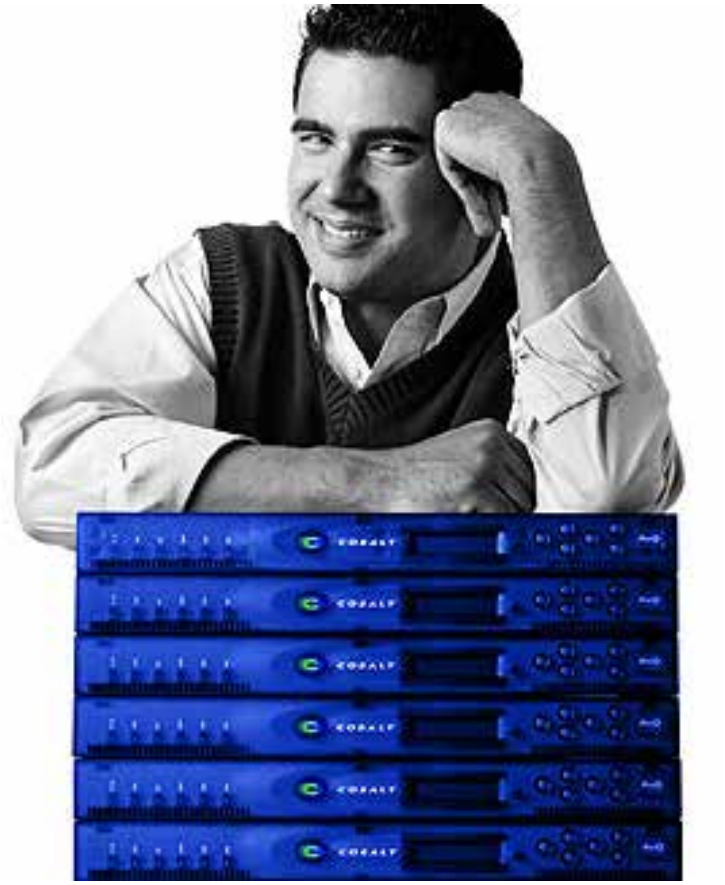
3.7.1 Εγκατάσταση επιμέρους προγραμμάτων (Modules)

Στο πρώτο μέρος της διαμόρφωσης (Global Enviroment) είδαμε μία λίστα με τα Modules που υποστηρίζει ο apache από την εγκατάσταση του . Επίσης παρατηρήσαμε ότι κάποια είχαν ένα # μπροστά που σημαίνει ότι δεν ήταν ενεργοποιημένα. Για να τα ενεργοποιήσουμε δεν έχει παρά μόνο να βγάλουμε το # που υπάρχει μπροστά από κάθε module και αυτόματα θα μπορούμε να το χρησιμοποιήσουμε. Αν θέλουμε κάτι παραπάνω από τα modules που υπάρχουν από την εγκατάσταση μπορούμε να επισκεφθούμε την διεύθυνση http://modules.apache.org να κατεβάσουμε το Module που μας ταιριάζει και ακολουθώντας τις οδηγίες εγκατάστασης του κατασκευαστή να το εγκαταστήσουμε. Από την μεριά του Apache δεν έχουμε παρά να προσθέσουμε μία γραμμή στο global enviroment που θα αρχίζει με την φράση Loadmodule θα αναφέρει το όνομα του module και την θέση που βρίσκετε το αρχείο

π.χ. **LoadModule asis_module modules/mod_asis.so**

ΚΕΦΑΛΑΙΟ 4

ΔΙΑΧΕΙΡΙΣΗ ΧΡΗΣΤΩΝ



4. Πιστοποίηση του client συστήματος

Η πιστοποίηση των χρηστών χρησιμοποιείται για σκοπούς παρακολούθησης ή ελέγχου πρόσβασης. Η προδιαγραφή του πρωτοκόλλου HTTP παρέχει δύο μηχανισμούς πιστοποίησης: βασικό(basic) και σύνοψης (digest). Και στις δύο περιπτώσεις, η διαδικασία που ακολουθείται είναι η εξής:

1. Ένα client σύστημα προσπαθεί να προσπελάσει το προστατευμένο περιεχόμενο στον web server.
2. Το Apache ελέγχει εάν το client σύστημα παρέχει όνομα χρήστη και κωδικό πρόσβασης. Εάν όχι, το Apache επιστρέφει τον κωδικό κατάστασης 401 του http, ο οποίος υποδεικνύει ότι απαιτείται πιστοποίηση του χρήστη.
3. Το client σύστημα διαβάζει την απάντηση και ζητά από τον χρήστη να εισάγει όνομα χρήστη και κωδικό πρόσβασης (συνήθως εμφανίζοντας ένα παράθυρο στην οθόνη).
4. Το client σύστημα προσπαθεί ξανά να προσπελάσει την ιστοσελίδα, στέλνοντας αυτή την φορά της πληροφορίες ονόματος χρήστη και κωδικού πρόσβασης σαν μέρος της αίτησης HTTP. Το client σύστημα απομνημονεύει το όνομα του χρήστη και τον κωδικό πρόσβασης και μεταδίδει τα στοιχεία αυτά στις επόμενες αιτήσεις που θα γίνουν προς το ίδιο site, έτσι ώστε ο χρήστης να μην είναι υποχρεωμένος να τα πληκτρολογεί ξανά και ξανά σε κάθε αίτηση που στέλνει.
5. Το Apache ελέγχει την εγκυρότητα των διαπιστευτηρίων του χρήστη και παραχωρεί πρόσβαση ή αρνείται την πρόσβαση ανάλογα με την ταυτότητα του χρήστη και άλλους κανόνες ελέγχου πρόσβασης.

Στο βασικό σχήμα πιστοποίησης το όνομα του χρήστη και ο κωδικός πρόσβασης του μεταδίδονται σε μορφή απλού κειμένου, σαν μέρος της αίτησης HTTP. Η προσέγγιση αυτή θα μπορούσε να είναι επικίνδυνη από την άποψη της ασφάλειας, επειδή ένας εισβολέας θα μπορούσε πολύ εύκολα να υποκλέψει την επικοινωνία που διαμείβεται μεταξύ του server και της εφαρμογής browser, να μάθει το όνομα χρήστη και τον κωδικό πρόσβασης και να τα χρησιμοποιήσει για οποιονδήποτε κακό σκοπό.

Το δεύτερο σχήμα πιστοποίησης παρέχει αυξημένη ασφάλεια επειδή μεταδίδει μία σύνοψη αντί για τον κωδικό πρόσβασης σε μορφή απλού κειμένου. Η σύνοψη βασίζεται σε έναν συνδυασμό αρκετών παραμέτρων, συμπεριλαμβανομένου του ονόματος χρήστη, του κωδικού πρόσβασης και της μεθόδου της αίτησης. Εξετάζοντας την σύνοψη ο server μπορεί να ελέγξει ότι το client σύστημα γνωρίζει τον κωδικό πρόσβασης, ακόμη κι αν ο κωδικός πρόσβασης δεν μεταδίδεται μέσω του δικτύου.

4.1 Μέθοδοι διαχείρισης χρηστών

Όταν το module πιστοποίησης λαμβάνει το όνομα χρήστη και των κωδικό πρόσβασης από το client σύστημα, πρέπει να επαληθεύσει ότι τα στοιχεία αυτά είναι έγκυρα ελέγχοντας τα έναντι μίας υπάρχουσας βάσης δεδομένων χρηστών. Τα ονόματα των χρηστών και οι κωδικοί πρόσβασης μπορούν να αποθηκεύονται σε διάφορες μορφές βάσεων δεδομένων. Το Apache υποστηρίζει μηχανισμούς πιστοποίησης βασιζόμενους σε **αρχεία** και **βάσεις δεδομένων**.

4.2 Οι λειτουργίες πιστοποίησης του Apache

Το Apache παρέχει την βασική υποδομή για την πιστοποίηση της ταυτότητας των χρηστών και τον έλεγχο πρόσβασης. Τα modules πιστοποίησης επιτρέπουν την επικύρωση των κωδικών πρόσβασης ελέγχοντας τους έναντι ενός αρχείου (ή βάσης δεδομένων) κωδικών πρόσβασης.

Το Apache διαθέτει τρεις ντιρεκτίβες οι οποίες σχετίζονται με την πιστοποίηση και μπορούν να χρησιμοποιούνται με οποιαδήποτε από τα modules πιστοποίησης: AuthName, AuthType και Require.

Η ντιρεκτίβα AuthName δέχεται σαν όρισμα ένα αλφαριθμητικό, το οποίο αντιπροσωπεύει το όνομα ενός τομέα πιστοποίησης(authentication realm). Υπ'αυτή την έννοια, τομέας(realm) θεωρείται μία λογική περιοχή του web server για την οποία ζητάμε τον κωδικό πρόσβασης.

Η ντιρεκτίβα AuthType καθορίζει τον τύπο πιστοποίησης της εφαρμογής browser: basic (βασικός) ή digest (βασιζόμενος σε αλγόριθμο σύναψης).

Η ντιρεκτίβα Require μας επιτρέπει να καθορίσουμε μία λίστα των χρηστών ή των ομάδων στις οποίες θα επιτρέπεται η πρόσβαση. Η σύνταξη της είναι Require <ένα ή περισσότερα ονόματα χρήστη> ή Require group <ένα ή περισσότερα ονόματα ομάδων>.

Εάν θέλουμε να παραχωρούμε πρόσβαση σε οποιονδήποτε παρέχει ένα έγκυρο όνομα χρήστη και κωδικό πρόσβασης, μπορούμε να το κάνουμε ως εξής:

Require valid-user

Με τις παραπάνω ντιρεκτίβες μπορούμε να ελέγχουμε ποιος θα έχει πρόσβαση σε συγκεκριμένους εικονικούς host, καταλόγους, αρχεία κ.λ.π. Αν και η πιστοποίηση και η παροχή πρόσβασης είναι δύο διαφορετικές έννοιες, σε πρακτικό επίπεδο συνδέονται στενά στο Apache. Η πρόσβαση παραχωρείται με βάση την ταυτότητα του χρήστη ή με βάση τις ομάδες στις οποίες είναι μέλος.

Τα εργαλεία πιστοποίησης που περιλαμβάνει το Apache παρέχουν τις ακόλουθες δυνατότητες:

- Αποθήκευση στοιχείων ταυτότητας- Οι πληροφορίες για τα ονόματα χρηστών και τις ομάδες αποθηκεύονται σε αρχεία κειμένου ή βάσης δεδομένων
- Διαχείριση χρηστών – Εργαλεία για την δημιουργία και διαχείριση χρηστών και ομάδων στο αρχείο που αποθηκεύει αυτές τις πληροφορίες
- Αυθεντικότητα πληροφοριών – Καθορίζετε εάν τα αποτελέσματα της ρουτίνας πιστοποίησης είναι αυθεντικά και πρέπει να τηρηθούν απαρέγκλιτα

4.3 Πιστοποίηση με βάση τις πληροφορίες που περιέχονται σε ένα αρχείο

Το module mod_auth του Apache παρέχει μία απλή μορφή πιστοποίησης μέσω αρχείων κειμένου τα οποία περιέχουν το ονόματα χρήστη και τους κωδικούς πρόσβασης.

4.3.1 Αποθήκευση στοιχείων ταυτότητας

Στο σημείο αυτό θα πρέπει να καθορίσουμε το αρχείο στο οποίο είναι αποθηκευμένα τα ονόματα των χρηστών, οι κωδικοί πρόσβασης και προαιρετικά το αρχείο που περιέχει την λίστα ομάδων.

Το αρχείο που περιέχει τα στοιχεία για τους χρήστες περιλαμβάνει τα ονόματα των χρηστών και τους κωδικούς πρόσβασης σε κρυπτογραφημένη μορφή. Εδώ χρησιμοποιείται ο αλγόριθμος MD5 και οι καταχωρήσεις στο αρχείο αυτό δείχνουν ως εξής:

admin : $aprl$Ug2….$hYTomGHIYFBLnh5hJ6YU/

Το αρχείο ομάδων περιέχει μία λίστα με όλες τις ομάδες. Κάθε καταχώρηση ομάδας περιλαμβάνει τα ονόματα των χρηστών που είναι μέλη αυτής της ομάδας, διαχωρισμένα με κενά διαστήματα μεταξύ τους ως εξής:

Web: admin john Peter

Οι ντιρεκτίβες AuthUserFile και AuthGroupFile δέχονται σαν όρισμα μία διαδρομή καταλόγων η οποία δείχνει στην θέση των αρχείων με τις πληροφορίες για τους χρήστες και τις ομάδες. Το αρχείο για τις ομάδες δεν είναι υποχρεωτικό.

4.4 Διαχείριση χρηστών σε αρχείο κειμένου

Το Apache περιλαμβάνει το βοήθημα htpasswd.exe. Το βοήθημα αυτό είναι ειδικά σχεδιασμένο ώστε να διευκολύνει την διαχείριση του αρχείου με τους κωδικούς πρόσβασης των χρηστών. Οι κωδικοί πρόσβασης κρυπτογραφούνται και η διαδικασία είναι διαφανής τόσο για τον χρήστη όσο και για τον επόπτη.

Όταν προσθέτουμε έναν χρήστη την πρώτη φορά θα πρέπει να εισάγουμε την εξής εντολή:

#> htpasswd –c file userid

όπου file είναι το αρχείο που περιέχει την λίστα με τα ονόματα των χρηστών και τους κωδικούς πρόσβασης, και userid είναι το όνομα του χρήστη που θέλουμε να προσθέσουμε. Στην συνέχει αφού μας ζητηθεί ένας κωδικός πρόσβασης το αρχείο θα δημιουργηθεί. Για παράδειγμα η ακόλουθη εντολή

htpasswd - c “C: \Program Files\Apache Group\Apache2\conf\htusers” admin

Θα δημιουργήσει το αρχείο κωδικών πρόσβασης htusers και θα προσθέσει στον χρήστη αυτό admin.

Η επιλογή **– c** λέει στο htpasswd ότι πρέπει να δημιουργήσει το αρχείο. Όταν θέλουμε να προσθέσουμε χρήστες σε ένα υφιστάμενο αρχείο κωδικών πρόσβασης δεν πρέπει να χρησιμοποιήσουμε το – c, αν γίνει αυτό το υπάρχον αρχείο θα διαγραφεί και θα αντικατασταθεί από ένα καινούργιο.

Το αρχείο κωδικών πρόσβασης πρέπει να αποθηκεύετε έξω από τον αρχικό κατάλογο εγγράφων έτσι ώστε μια εφαρμογή Web Browser να μην μπορεί να το προσπελάσει. Σε περίπτωση που δεν το κάνουμε αυτό θα μπορούσε ο οποιοσδήποτε να μεταφέρει στο σύστημα του το αρχείο αυτό και να αποκτήσει την λίστα με τα ονόματα και τους κωδικούς πρόσβασης των χρηστών.

Η ντιρεκτίβα Authoritative

Η ντιρεκτίβα Authoritative δέχεται σαν όρισμα μία από τις τιμές on ή off. Η τιμή αυτής της ντιρεκτίβας είναι πάντα on που σημαίνει ότι τα αποτελέσματα της ρουτίνας πιστοποίησης πρέπει να τηρηθούν επακριβώς. Δηλαδή, αν ο χρήστης δεν βρεθεί ή δεν ικανοποιεί κανέναν από τους κανόνες δεν του παραχωρείται πρόσβαση.

4.5 Χρήση μιας Βάσης Δεδομένων για τον Έλεγχο Πρόσβασης

Η αποθήκευση των ονομάτων των χρηστών και των κωδικών πρόσβασης σε αρχεία απλού κειμένου είναι βολική αλλά δεν επιτρέπει την επέκταση της εφαρμογής μας σε ευρύτερη κλίμακα. Το Apache είναι υποχρεωμένο να ανοίγει και να διαβάζει σειριακά τα αρχεία, όταν ψάχνει για έναν συγκεκριμένο χρήστη. Εάν οι χρήστες είναι πολλοί αυτή η διαδικασία είναι πολύ χρονοβόρα. Το module mod_auth_dbm μας επιτρέπει να αντικαταστήσουμε τα αρχεία απλού κειμένου με βάσεις δεδομένων οι οποίες μπορούν να χειρίζονται πολύ μεγαλύτερο αριθμό χρηστών χωρίς την μείωση της απόδοσης. Το module mod_auth_dbm περιλαμβάνεται στο Apache αλλά δεν είναι ενεργοποιημένο εξαρχής.

4.5.1 Αποθήκευση στοιχείων ταυτότητας

Το module mod_auth_dbm παρέχει δυο ντιρεκτίβες, τις AuthDBMUserFile και AuthDBMGroupFile, οι οποίες δείχνουν τα αρχεία βάσεων δεδομένων που περιέχουν τις πληροφορίες για τους χρήστες και τις ομάδες αντίστοιχα. Αντίθετα με τα αρχεία απλού κειμένου και οι δύο αυτές ντιρεκτίβες μπορούν να δείχνουν στο ίδιο αρχείο πληροφορίες τόσο για τους χρήστες όσο και για τις ομάδες.

4.5.2 Διαχείριση Χρηστών σε βάση δεδομένων

Το Apache διαθέτει ένα script γραμμένο σε Perl με όνομα dbmmanage.pl, το οποίο μας επιτρέπει να δημιουργήσουμε και να διαχειριστούμε χρήστες και ομάδες σε ένα αρχείο βάσης δεδομένων. Για να μπορούμε να το κάνουμε αυτό θα πρέπει να εγκαταστήσουμε το επιπλέον πακέτο MD5 για κρυπτογράφηση των κωδικών πρόσβασης. Εάν χρησιμοποιούμε την ActiveState Perl, εκκινούμε τον διαχειριστή πακέτων (package manager) της Perl και εισάγουμε την εντολή :

install Crypt-PasswdMD5

Για να προσθέσουμε έναν χρήστη σε μία βάση δεδομένων εισάγουμε την εντολή:

Perl ./dbmmanage .pl dbfile adduser ταυτότητα_χρήστη

Ακολούθως θα μας ζητηθεί ο κωδικός πρόσβασης και ο χρήστης θα προστεθεί στο υπάρχον αρχείο βάσης δεδομένων ή θα δημιουργηθεί ένα νέο αρχείο βάσης δεδομένων αν δεν υπάρχει ήδη.

Όταν προσθέτουμε ένα χρήστη μπορούμε αν θέλουμε να καθορίσουμε τις ομάδες στις οποίες θα είναι μέλος σαν ορίσματα τα οποία διαχωρίζονται μεταξύ τους με κόμματα.

Η εντολή:

#> dbmmanage /usr/local/apache2/conf/dbmusers adduser john teacher, director

προσθέτει τον χρήστη john στην βάση δεδομένων /usr/local/apache2/conf/dbmusers και ταυτόχρονα τον ορίζει σαν μέλος στις ομάδες teacher και director.

Αν θέλουμε να διαγράψουμε τον χρήστη john μπορούμε να εισάγουμε την ακόλουθη εντολή:

#> dbmmanage dbfile delete john

ΚΕΦΑΛΑΙΟ 5

ΑΣΦΑΛΕΙΑ

5. Ασφάλεια στους Web server

Καθώς η χρήση του Διαδικτύου διαδίδεται με διαρκώς γρηγορότερο ρυθμό και ο αριθμός των εταιρειών, των ατόμων και των κυβερνητικών υπηρεσιών που το χρησιμοποιούν αυξάνεται συνεχώς, το ίδιο ισχύει για το πλήθος και τα είδη των συναλλαγών που χρειάζονται προστασία. Ορισμένα καίρια παραδείγματα είναι οι τραπεζικές συναλλαγές, το ηλεκτρονικό εμπόριο και η διακίνηση εμπιστευτικών πληροφοριών, όπως π.χ ιατρικά στοιχεία και εταιρικά έγραφα. Οι προϋποθέσεις που πρέπει να καλύπτονται για την διεξαγωγή επικοινωνιών με ασφάλεια μέσω του διαδικτύου είναι τρεις : εμπιστευτικότητα, ακεραιότητα και πιστοποίηση.

Εμπιστευτικότητα

Η εμπιστευτικότητα είναι η προφανέστερη προϋπόθεση για την ασφαλή επικοινωνία. Εάν διακινείτε ή προσπελάζετε «ευαίσθητες» πληροφορίες, όπως αριθμούς πιστωτικών καρτών ή το προσωπικό ιατρικό ιστορικό σας, σίγουρα δεν θα θέλατε τα στοιχεία αυτά να πέσουν στα χέρια οποιουδήποτε αναρμόδιου ατόμου.

Ακεραιότητα

Η πληροφορία που περιέχεται στα μηνύματα που διακινούνται πρέπει να προστατεύεται από οποιονδήποτε εξωτερικό κίνδυνο.

Πιστοποίηση

Ανάμεσα στις δύο πλευρές που επικοινωνούν είτε αυτά είναι άτομα είτε οργανισμοί θα πρέπει να υπάρχει εμπιστοσύνη μεταξύ τους. Για να γίνει αυτό πρέπει να υπάρχει κάποιος τρόπος να πιστοποιείτε η ταυτότητα αυτών που επικοινωνούν.

Η επιστήμη της κρυπτογράφησης μελετά τους αλγόριθμους και τις μεθόδους που χρησιμοποιούνται για την ασφαλή διακίνηση μηνυμάτων, έτσι ώστε να επιτυγχάνονται οι στόχοι της εμπιστευτικότητας, της ακεραιότητας και της

πιστοποίησης της ταυτότητας των συμβαλλόμενων μερών σε οποιαδήποτε μορφή επικοινωνίας. Η κρυπτοανάλυση είναι η επιστήμη που ασχολείται με το «σπάσιμο» των συστημάτων κρυπτογράφησης.

5.1 Το Πρωτόκολλο SSL

Το SSL και TLS είναι ακρωνύμια των όρων Secure Sockets Layer και Transport Layer Security, αντίστοιχα. Πρόκειται για μία οικογένεια πρωτοκόλλων τα οποία σχεδιάστηκαν αρχικά για να παρέχουν ασφάλεια στις συναλλαγές που διεξάγονται μέσω του πρωτοκόλλου HTTP, αλλά μπορούν επίσης να χρησιμοποιηθούν με μία ποικιλία άλλων πρωτοκόλλων του διαδικτύου. Όταν το πρωτόκολλο HTTP τρέχει πάνω από το SSL, αναφέρεται σαν "ασφαλές HTTP".

Το πρωτόκολλο TLS είναι ένα πρότυπο που αναπτύχθηκε με στόχο να προτυποποίηση το SSL σαν ένα πρωτόκολλο για το Διαδίκτυο. Αποτελεί απλά μία τροποποίηση της $3^{ης}$ έκδοσης του SSL που έγινε το 1995, με έναν μικρό αριθμό επιπλέον λειτουργιών και δυνατοτήτων. Το ακρωνύμιο TLS είναι το αποτέλεσμα της διαφωνίας ανάμεσα σε δύο εταιρείες που η κάθε μία πρότεινε το δικό της όνομα. Λόγο του ότι το όνομα αυτό δεν έχει καθιερωθεί, όπως και οι περισσότεροι άνθρωποι, θα αναφερόμαστε συλλογικά και στα δύο πρωτόκολλα με τον όρο SSL.

Πιο κάτω θα δούμε πως χειρίζεται το SSL τα θέματα της εμπιστευτικότητας, της ακεραιότητας των δεδομένων και της πιστοποίησης στα οποία αναφερθήκαμε πριν.

5.2 Η ανάγκη για Εμπιστευτικότητα

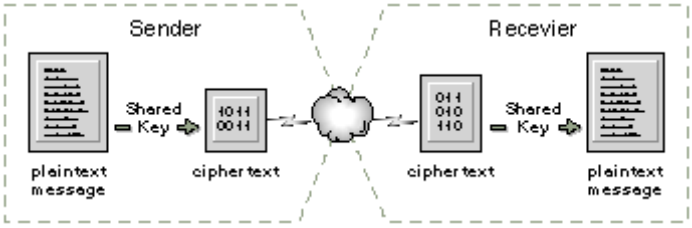
Για να προστατεύσει τα δεδομένα από απόπειρες υποκλοπής το πρωτόκολλο SSL τα κρυπτογραφεί. Η κρυπτογράφηση είναι η διαδικασία μετατροπής ενός μηνύματος (απλού κειμένου) σε μία νέα, κρυπτογραφημένη μορφή η οποία αποκαλείται κρυπτογράφημα. Ενώ το απλό κείμενο είναι αναγνώσιμο από οποιονδήποτε, το κρυπτογράφημα είναι απόλυτα ακατανόητο για οποιονδήποτε το υποκλέπτει. Η αποκρυπτογράφηση είναι η αντίστροφη διαδικασία η οποία μετασχηματίζει το κρυπτογράφημα στο αρχικό απλό κείμενο του μηνύματος.

Συνήθως οι διαδικασίες κρυπτογράφησης και αποκρυπτογράφησης εμπλέκουν και απαιτούν ένα επιπλέον στοιχείο πληροφορίας: ένα κλειδί. Εάν ο αποστολέας και ο παραλήπτης έχουν το ίδιο κλειδί, η διαδικασία αναφέρεται σαν συμμετρική κρυπτογράφηση. Εάν ο αποστολέας και ο παραλήπτης του μηνύματος έχουν διαφορετικά κλειδιά αλλά με συμπληρωματικό ρόλο, η διαδικασία αποκαλείται ασύμμετρη κρυπτογράφηση η κρυπτογράφηση δημοσίου κλειδιού.

5.2.1 Συμμετρική κρυπτογράφηση

Όταν χρησιμοποιείται το ίδιο κλειδί τόσο για την κρυπτογράφηση, όσο και για την αποκρυπτογράφηση του μηνύματος, η διαδικασία αναφέρεται συνολικά σαν συμμετρική κρυπτογράφηση. Οι DES, Triple-Des, RC4 και RC2 είναι αλγόριθμοι οι οποίοι χρησιμοποιούνται για συμμετρική κρυπτογράφηση. Πολλοί από αυτούς τους αλγόριθμους μπορούν να δέχονται κλειδιά διαφορετικού μεγέθους, το οποίο μετριέται σε bits. Γενικά, με δεδομένο έναν αλγόριθμο, όσο μεγαλύτερος είναι ο αριθμός των bits του κλειδιού, τόσο πιο ασφαλής θεωρείται ο αλγόριθμος και τόσο αργότερα λειτουργεί, λόγω των περισσοτέρων υπολογισμών που απαιτούνται για την εκτέλεση του.

Η συμμετρική κρυπτογράφηση είναι σχετικά γρήγορη σε σύγκριση με την κρυπτογράφηση δημοσίου κλειδιού, την οποία θα δούμε πιο κάτω. Η συμμετρική κρυπτογράφηση ωστόσο έχει δύο βασικές αδυναμίες. Η πρώτη είναι ότι το κλειδί πρέπει να αλλάζει περιοδικά για να αποφεύγεται οποιαδήποτε πιθανότητα αποκάλυψης του, η οποία θα έδινε πρόσβαση σε ένα κακόβουλο άτομο σε μεγάλες ποσότητες υλικού κρυπτογραφημένου με το ίδιο κλειδί. Το δεύτερο σχετίζεται με την διανομή του κλειδιού: Πώς θα φτάσει το κλειδί με ασφάλεια στα δύο επικοινωνούντα μέρη; Αυτός ήταν ένας από τους πρώτους περιοριστικούς παράγοντες και πριν από την ανακάλυψη της κρυπτογράφησης δημοσίου κλειδιού η λύση γι'αυτό το πρόβλημα ήταν να ταξιδεύουν υπεύθυνα άτομα σε όλο τον κόσμο σε τακτά χρονικά διαστήματα κουβαλώντας μαζί τους βαλίτσες γεμάτες με κλειδιά κρυπτογράφησης.

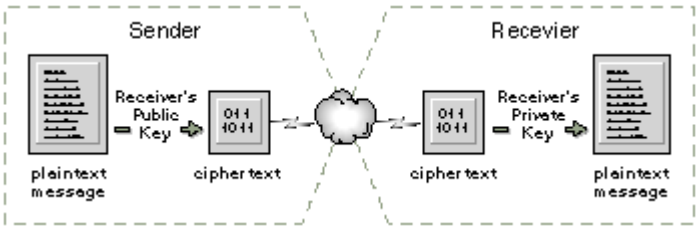


5.2.2 Κρυπτογράφηση δημοσίου κλειδιού

Η κρυπτογράφηση δημοσίου κλειδιού ακολουθεί διαφορετική προσέγγιση. Αντί τα δύο συμβαλλόμενα μέρη να μοιράζονται το ίδιο κλειδί, υπάρχει ένα ζεύγος κλειδιών: ένα δημόσιο (public key) και ένα ιδιωτικό κλειδί (private key). Το δημόσιο κλειδί είναι ελεύθερα διαθέσιμο, ενώ το ιδιωτικό κλειδί είναι μυστικό, γνωστό μόνο στον κάτοχο του. Αυτά τα δύο κλειδιά είναι συμπληρωματικά, ένα μήνυμα κρυπτογραφημένο με ένα από αυτά τα κλειδιά μπορεί να αποκρυπτογραφηθεί μόνο με την χρήση του άλλου κλειδιού. Αν κάποιος θέλει να στείλει σε εμάς ένα ασφαλές μήνυμα, μπορεί να το κρυπτογραφήσει χρησιμοποιώντας το δημόσιο κλειδί μας για να είναι βέβαιος ότι μόνο ο κάτοχος του ιδιωτικού κλειδιού-δηλαδή εμείς- θα μπορεί να το αποκρυπτογραφήσει. Το μήνυμα δεν μπορεί να αποκρυπτογραφηθεί ακόμη

και αν κάποιος εισβολέας αποκτήσει πρόσβαση στο δημόσιο κλειδί. Η κρυπτογράφηση δημοσίου κλειδιού μπορεί να χρησιμοποιηθεί επίσης για την κάλυψη των άλλων δύο προϋποθέσεων, της ακεραιότητας του μηνύματος και της πιστοποίησης της ταυτότητας του αποστολέα. Ο δημοφιλέστερος αλγόριθμος κρυπτογράφησης δημοσίου κλειδιού είναι ο RSA.

Το πρωτόκολλο SSL χρησιμοποιεί κρυπτογράφηση δημοσίου κλειδιού σε μία αρχική φάση «χαιρετισμού» για την ασφαλή διακίνηση συμμετρικών κλειδιών, τα οποία μπορούν κατόπιν να χρησιμοποιηθούν για την κρυπτογράφηση της επικοινωνίας.



5.3 Η Ανάγκη για ακεραιότητα

Η εκτέλεση ενός ειδικού υπολογισμού με τα περιεχόμενα του μηνύματος και η αποθήκευση του αποτελέσματος μέσα στο ίδιο το μήνυμα είναι ένας τρόπος με τον οποίο μπορεί να διατηρηθεί η ακεραιότητα των δεδομένων. Όταν το μήνυμα φτάσει στον προορισμό του ο παραλήπτης μπορεί να εκτελέσει τον ίδιο υπολογισμό και να συγκρίνει τα δύο αποτελέσματα. Εάν τα περιεχόμενα του μηνύματος έχουν αλλοιωθεί κατά την πορεία προς τον προορισμό, τα αποτελέσματα των δύο υπολογισμών θα είναι διαφορετικά.

Για την εκτέλεση αυτής της διαδικασίας υπάρχουν ειδικοί αλγόριθμοι οι οποίοι δημιουργούν συνόψεις μηνυμάτων (message digest). Μία σύνοψη μηνύματος είναι μία μέθοδος για την δημιουργία μιας σταθερού μεγέθους αναπαράστασης ενός τυχαίου μηνύματος το οποίο προσδιορίζει με μονοσήμαντο τρόπο το αρχικό μήνυμα. Μπορούμε να την θεωρήσουμε σαν το δακτυλικό αποτύπωμα του αρχικού μηνύματος. Ένας καλός αλγόριθμος σύνοψης μηνυμάτων πρέπει να είναι «μη-αναστρέψιμος» και μονοσήμαντος, σε πρακτικό τουλάχιστο επίπεδο. Το «μη-αναστρέψιμος» σημαίνει ότι κανείς

δεν θα μπορεί να ανακτήσει το αρχικό μήνυμα από την σύνοψη και το μονοσήμαντος σημαίνει ότι δεν μπορούν να υπάρχουν δύο διαφορετικά μηνύματα με την ίδια σύνοψη. Οι MD5 και SHA είναι δύο γνωστοί αλγόριθμοι σύνοψης μηνυμάτων. Ωστόσο, από μόνοι τους αυτοί οι αλγόριθμοι δεν εγγυώνται την ακεραιότητα του αρχικού μηνύματος, επειδή ένας εισβολέας θα μπορούσε να αλλάξει τόσο το κείμενο του αρχικού μηνύματος, όσο κι την παραγόμενη σύνοψη.

Οι κωδικοί πιστοποίησης μηνυμάτων (message authentication codes ή MAC) είναι παρόμοιοι με τις συνόψεις μηνυμάτων , αλλά ενσωματώνουν ένα κοινόχρηστο μυστικό κλειδί στην διαδικασία. Το αποτέλεσμα του αλγορίθμου εξαρτάται τόσο από το μήνυμα, όσο και από το κλειδί. Επειδή ένας εισβολέας δεν έχει πρόσβαση στο κλειδί, δεν μπορεί να τροποποιήσει και το μήνυμα και την σύνοψη. Ένας γνωστός αλγόριθμος που χρησιμοποιεί κωδικούς πιστοποίησης μηνυμάτων είναι ο HMAC.

Το πρωτόκολλο SSL χρησιμοποιεί κωδικούς MAC για την αποτροπή επιθέσεων αναμετάδοσης και για να διασφαλίζει την ακεραιότητα των μεταδιδόμενων πληροφοριών.

5.4 Η ανάγκη για Πιστοποίηση

Το SSL χρησιμοποιεί πιστοποιητικά (certificates) για να πιστοποιήσει την ταυτότητα των δύο συμβαλλόμενων μερών σε μία επικοινωνία. Η κρυπτογράφηση δημοσίου κλειδιού μπορεί να χρησιμοποιηθεί για την ψηφιακή υπογραφή μηνυμάτων. Στην πραγματικότητα, κρυπτογραφώντας απλώς ένα μήνυμα με το μυστικό μας κλειδί ο παραλήπτης του μπορεί να είναι σίγουρος ότι το μήνυμα αυτό προήλθε από εμάς. Άλλοι αλγόριθμοι ψηφιακών υπογραφών υπολογίζουν πρώτα μία σύνοψη του μηνύματος και κατόπιν υπογράφουν την σύνοψη.

Μπορούμε να είμαστε σίγουροι ότι το άτομο που δημιούργησε το ζεύγος δημοσίου /ιδιωτικού κλειδιού είναι αυτό που έστειλε το μήνυμα. Αλλά πως

μπορούμε να συσχετίσουμε το κλειδί με ένα άτομο ή με έναν οργανισμό τον οποίο θα μπορούμε να εμπιστευτούμε στον πραγματικό κόσμο; Εάν δεν υπάρχει κάποιος μηχανισμός πιστοποίησης, ένας εισβολέας θα μπορούσε να προσποιηθεί ότι είναι εμείς και να διανέμει ένα διαφορετικό δημόσιο κλειδί, ισχυριζόμενος ότι αυτό είναι το έγκυρο. Η εμπιστοσύνη μπορεί να επιτευχθεί με τη χρήση ψηφιακών πιστοποιητικών. Τα ψηφιακά πιστοποιητικά είναι ηλεκτρονικά έγγραφα τα οποία περιέχουν ένα δημόσιο κλειδί και πληροφορίες για τον κάτοχό του (όνομα, διεύθυνση κ.τ.λ). Για να είναι χρήσιμο το πιστοποιητικό, πρέπει να είναι υπογεγραμμένο από έναν αξιόπιστο και ανεξάρτητο φορέα (λέγονται συνήθως φορείς έκδοσης πιστοποιητικών, Certification Authority, CA) ο οποίος πιστοποιεί ότι οι πληροφορίες του ψηφιακού πιστοποιητικού είναι σωστές. Ορισμένοι από αυτούς τους φορείς είναι εμπορικές οντότητες και παρέχουν υπηρεσίες έκδοσης πιστοποιητικών σε εταιρείες οι οποίες δραστηριοποιούνται επαγγελματικά μέσω του διαδικτύου.

Ο φορέας CA εγγυάται ότι οι πληροφορίες που περιέχονται στο πιστοποιητικό είναι σωστές και ότι αυτό ανήκει πράγματι στο άτομο ή στον οργανισμό που φέρεται ως κάτοχος του. Τα πιστοποιητικά έχουν συγκεκριμένη περίοδο εγκυρότητας και μπορούν να λήξουν ή να ανακληθούν. Επίσης, ένα πιστοποιητικό μπορεί να είναι «αλυσιδωτό», έτσι ώστε η διαδικασία πιστοποίησης να περνά από περισσότερα του ενός στάδια. Για παράδειγμα, μία αξιόπιστη οντότητα μπορεί να παρέχει πιστοποιητικά σε εταιρείες, οι οποίες με την σειρά τους μπορούν να φροντίζουν για την πιστοποίηση της ταυτότητας των υπαλλήλων τους.

Για να είναι αποτελεσματική και αξιόπιστη η όλη διαδικασία, ο φορέας έκδοσης πιστοποιητικών πρέπει να έχει στα χέρια του τις κατάλληλες αποδείξεις για την ταυτότητα ενός ιδιώτη ή ενός οργανισμού, πριν εκδώσει ένα πιστοποιητικό γι'αυτόν. Εξ ορισμού, οι εφαρμογές browser περιλαμβάνουν ένα σύνολο πιστοποιητικών για τους μεγάλους και αξιόπιστους φορείς έκδοσης πιστοποιητικών που υπάρχουν σήμερα.

5.5 Το Πρωτόκολλο SSL και τα πιστοποιητικά

Το βασικό πρότυπο που ορίζει τις προδιαγραφές για τα πιστοποιητικά ονομάζεται X.509 και έχει προσαρμοστεί για χρήση στο διαδίκτυο. Ένα πιστοποιητικό συμβατό με το πρότυπο X.509 περιέχει τις ακόλουθες πληροφορίες.

- Issuer (εκδότης) – Το όνομα αυτού που υπογράφει το πιστοποιητικό
- Subject (υποκείμενο) – Το άτομο στο οποίο ανήκει το κλειδί που πιστοποιείται
- Subject public key – Το δημόσιο κλειδί αυτού του ατόμου
- Πληροφορίες ελέγχου – Δεδομένα όπως οι ημερομηνίες έναρξης / λήξης της εγκυρότητας του πιστοποιητικού
- Υπογραφή – Η υπογραφή που καλύπτει τα παραπάνω δεδομένα

Μπορούμε να εξετάσουμε ένα πραγματικό πιστοποιητικό συνδεόμενοι σε έναν ασφαλές server με μία εφαρμογή browser. Εάν η σύνδεση ολοκληρωθεί επιτυχώς, θα εμφανιστεί ένα μικρό εικονίδιο λουκέτου ή κάποιο αντίστοιχο οπτικό εικονίδιο στην γραμμή καταστάσεων της εφαρμογής browser. Εάν χρησιμοποιούμε τον Internet Explorer μπορούμε να κάνουμε κλικ στο εικονίδιο κλειδωμένου λουκέτου για να ανοίξουμε μία σελίδα η οποία περιέχει πληροφορίες για την σύνδεση SSL και το πιστοποιητικό του απομακρυσμένου server. Μπορούμε επίσης να προσπελάσουμε τις ίδιες πληροφορίες επιλέγοντας Properties > Certificates από το μενού File.

5.6 Σύνοψη του πρωτοκόλλου SSL

Αφού είδαμε ότι το SSL επιτυγχάνει τη εμπιστευτικότητα μέσω της κρυπτογράφησης, την ακεραιότητα μέσω των κωδικών πιστοποίησης μηνυμάτων και την πιστοποίηση της ταυτότητας των επικοινωνούντων μερών μέσω των πιστοποιητικών και ψηφιακών υπογραφών, ας δούμε τώρα την διαδικασία για την υλοποίηση μιας σύνδεσης μέσω του SSL.

5.7 Η διαδικασία για την υλοποίηση μιας σύνδεσης μέσω του SSL είναι:

- Ø Ο χρήστης χρησιμοποιεί μια εφαρμογή browser για να συνδεθεί στον απομακρυσμένο Apache server.
- Ø Ξεκινάει η φάση του χαιρετισμού (handshake).Η εφαρμογή browser και ο server ανταλλάσσουν κλειδιά και πληροφορίες πιστοποιητικών.
- Ø Η εφαρμογή browser ελέγχει την εγκυρότητα του πιστοποιητικού του server. Επίσης ελέγχει ότι το πιστοποιητικό δεν έχει λήξει, ότι έχει εκδοθεί από έναν αξιόπιστο φορέα CA, κ.ο.κ.
- Ø Προαιρετικά, ο server μπορεί επίσης να απαιτήσει από το client σύστημα να του παρουσιάσει ένα έγκυρο πιστοποιητικό.
- Ø Ο server και το client σύστημα χρησιμοποιούν ο ένας το δημόσιο κλειδί του άλλου για να συμφωνήσουν με ασφάλεια σε ένα συμμετρικό κλειδί.
- Ø Η φάση του χαιρετισμού ολοκληρώνεται και η μετάδοση συνεχίζεται χρησιμοποιώντας συμμετρική κρυπτογράφηση

5.8 Εγκατάσταση του SSL στο Apache

Η υποστήριξη για το SSL παρέχεται από το εργαλείο mod_ssl, το οποίο περιλαμβάνεται στο Apache, αλλά δεν είναι ενεργοποιημένο εξ ορισμού. Με την σειρά του, το mod_ssl απαιτεί την βιβλιοθήκη OpenSSL (μια υλοποίηση των πρωτοκόλλων SSL/TLS) σε μορφή ανοικτού κώδικα μαζί με άλλους αλγόριθμους κρυπτογράφησης.

Επειδή οι απαιτούμενες βιβλιοθήκες OpenSSL περιλαμβάνονται στην ρουτίνα εγκατάστασης του Apache 2.0 για windows, δεν απαιτείται να μεταφερθούν ή να εγκατασταθούν οποιαδήποτε άλλα αρχεία. Το πρόγραμμα openssl.exe περιλαμβάνεται στον κατάλογο bin/ της διανομής του Apache. Είναι ένα βοήθημα για την παραγωγή πιστοποιητικών, κλειδιών, αιτήσεων υπογραφής πιστοποιητικών, κ.ο.κ.

ΚΕΦΑΛΑΙΟ 6

ΣΥΓΚΡΙΣΗ

APACHE 1.3

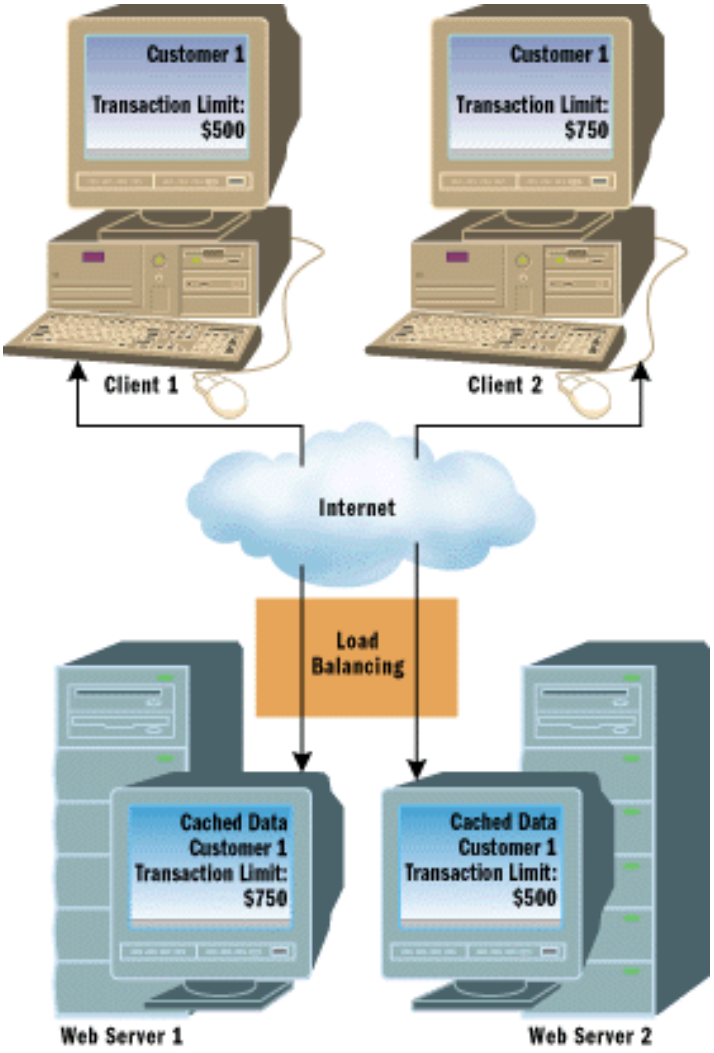
& APACHE 2.0

ΜΕΛΛΟΝΤΙΚΑ

ΣΧΕΔΙΑ

6. Σύγκριση Apache 1.3 και Apache 2.0

Το Apache 2.0 είναι ένας webserver γενικής χρήσης, με σκοπό να παρέχει μια ισορροπία ευελιξίας, φορητότητας και απόδοσης. Παρόλα αυτά δεν έχει ως σκοπό να θέσει τα αρχεία σε συγκριτική μέτρηση επιδόσεων, το Apache 2.0 είναι ικανό για υψηλή απόδοση σε πολλές πραγματικές καταστάσεις.
Έναντι στο Apache 1.3, η έκδοση 2.0 περιέχει πολλές πρόσθετες βελτιστοποιήσεις για να αυξήσει το ρυθμό απόδοσης και την εξελιξιμότητα.
Οι περισσότερες από αυτές τις βελτιώσεις επιτρέπονται εξ ορισμού. Εντούτοις υπάρχουν οι επιλογές διαμόρφωσης σύνταξης-χρόνου (compile-time) και χρόνου εκτέλεσης (run-time) που μπορούν να έχουν σημαντικές επιπτώσεις στην απόδοση.



6.1 Μελλοντικά σχέδια για τον apache server

Τα μελλοντικά σχέδια για τον Apache Server είναι τα εξής:

- Να συνεχίσει να είναι “open source” (ανοικτού κώδικα) HTTP server χωρίς να υπάρχει κάποια χρέωση για να μπορείς να τον χρησιμοποιήσεις.
- Να συμβαδίζει με την εξέλιξη και την πρόοδο του πρωτοκόλλου HTTP και του ιστού (Web) γενικά
- Να συλλέξει από τους χρήστες του τις προτάσεις και εισηγήσεις για διορθώσεις και βελτιώσεις
- Να ανταποκρίνεται στις ανάγκες των προμηθευτών μεγάλου όγκου απαιτήσεων καθώς επίσης και των περιστασιακών χρηστών.

ΒΙΒΛΙΟΓΡΑΦΙΑ

ΙΣΤΟΣΕΛΙΔΕΣ

- http://httpd.apache.org/docs/
- http://gnuwin.epfl.ch/apps/Apache/en/index.html
- http://www.keepmedia.com/pubs/PCMagazine/2002/01/15/431150/?extID=10047&data=apache_server
- http://www.logidac.com/apache/apxs/

ΒΙΒΛΙΑ

- Μάθετε PHP, MySQL και Apache σε 24 ώρες
- Apache Server