ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

HYPERTEXT TRANSFER PROTOCOL

Ε. Χατζηευθυμιάδης & Δ. Μαρτάκος

Αθήνα 1996

HYPERTEXT TRANSFER PROTOCOL

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ
Πανεπιστημιόπολις, Κτήρια Πληροφορικής
157 84 Αθήνα
Ελλάδα

Ε. Χατζηευθυμιάδης
MSc στην Πληροφορική, Ερευνητής
Τμήμα Πληροφορικής
Εθνικό & Καποδιστριακό Πανεπιστήμιο Αθηνών
E_mail: shadj@di.uoa.gr

Δ. Μαρτάκος
Επίκουρος Καθηγητής
Τμήμα Πληροφορικής
Εθνικό & Καποδιστριακό Πανεπιστήμιο Αθηνών
E_mail: matrakos@di.uoa.gr

ΠΕΡΙΕΧΟΜΕΝΑ

1. Γενικά

Το HyperText Transfer Protocol (HTTP) [1, 2, 3] αποτελεί το βασικό πρωτόκολλο για την ανταλλαγή πληροφορίας στο πλαίσιο του WWW. Είναι ένα ιδιαίτερα ευέλικτο πρωτόκολλο επιπέδου εφαρμογής (application level) που καθορίζει απλές δοσοληψίες μεταξύ του WWW browser και ενός HTTP server. Βασικός στόχος του HTTP είναι η επίτευξη χαμηλών χρόνων απόκρισης (response times). Προς αυτή την κατεύθυνση το HTTP αναπτύχθηκε σαν πρωτόκολλο χωρίς μνήμη (stateless protocol) δηλ. δεν διατηρεί καμία πληροφορία για μία σύνδεση μετά από την διεκπεραίωση μίας σχετικής αίτησης. Η διατήρηση πληροφορίας κατάστασης μπορεί να επιτευχθεί εκτός από τον ίδιο τον HTTP server μέσω εξωτερικών προγραμμάτων που ακολουθούν το πρωτόκολλο CGI ή βάσεων δεδομένων. Τέλος το HTTP χαρακτηρίζεται αντικειμενοστρεφές (object oriented protocol). Μπορεί να εφαρμοστεί, με μικρές μετατροπές στις υποστηριζόμενες μεθόδους, σε name servers και κατανεμημένα συστήματα διαχείρισης αντικειμένων.

Το HTTP έχει υποστεί βελτιστοποίηση και ειδικό σχεδιασμό για κατανεμημένα, συλλογικά (collaborative) πληροφοριακά συστήματα υπερμέσων. Διαθέτει την απαιτούμενη ευελιξία για την υποστήριξη των hypertext jumps ενώ τα μεταδιδόμενα δεδομένα μπορεί να είναι απλό κείμενο, εικόνες, υπερκείμενο κα.

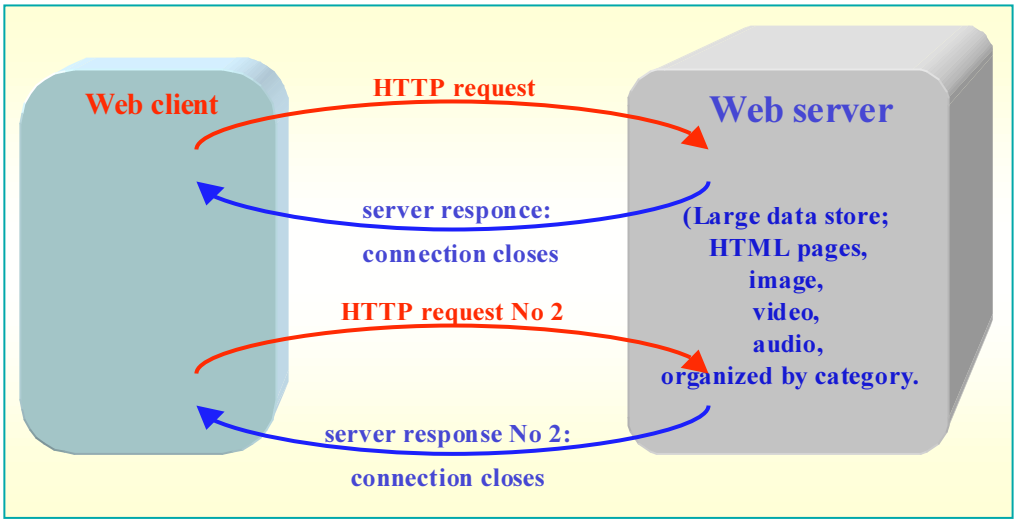
Τα μηνύματα του HTTP μοιάζουν σημαντικά με αυτά των πρωτοκόλλων FTP (File Transfer) και NNTP (Network News). Η βασική τους διαφορά είναι ο stateless χαρακτήρας του HTTP που δεν εντοπίζεται στα υπόλοιπα. Η απουσία μνήμης κρίνεται αποδοτική (efficient) για το πρωτόκολλο όταν ένας σύνδεσμος (link) από ένα αντικείμενο οδηγεί σε ένα αντικείμενο που βρίσκεται αποθηκευμένο σε άλλο server. Επίσης η ιδιότητα αυτή κρίνεται κατάλληλη εφόσον ο client επιστρέφει πληροφορία στον χρήστη με βάση URIs και όχι παλαιότερες ενέργειες του.

2. Λειτουργία

Το HTTP ακολουθεί το μοντέλο request/response. Ο client εγκαθιδρύει μία σύνδεση με τον server (κάνοντας χρήση του πρωτοκόλλου TCP) και αποστέλλει μία αίτηση προς αυτόν η οποία περιέχει:

- *Την μέθοδο που πρόκειται να εφαρμοστεί σαν αποτέλεσμα της αίτησης (request method). Η χρήση του όρου μέθοδος οφείλεται στον αντικειμενοστρεφή προσανατολισμό του πρωτοκόλλου.*

- *Ενα Universal Resource Identifier (URI). Ο πόρος στον οποίο πρόκειται να εφαρμοστεί η παραπάνω μέθοδος.*
- *Την έκδοση του χρησιμοποιούμενου πρωτοκόλλου.*
- *Ενα μήνυμα που ακολουθεί την μορφή MIME (Multipurpose Internet Mail Extensions) και περιέχει πληροφορία σχετικά με τον client, πιθανά το σώμα του μηνύματος κ.α.*



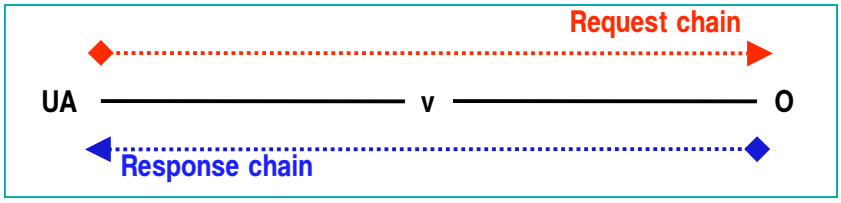
Source: HTML & CGI Unleashed, 1995

Σχήμα 1: Stateless HTTP

Ο server απαντάει με ένα μήνυμα που περιέχει:

- *Μία γραμμή κατάστασης (Status line) που περιέχει την έκδοση του πρωτοκόλλου και κωδικό επιτυχίας/αποτυχίας (success/error code).*
- *Ενα μήνυμα που ακολουθεί την μορφή MIME και περιέχει πληροφορία σχετικά με τον server, μεταπληροφορία σχετικά με το μεταφερόμενο αντικείμενο και πιθανά το σώμα του μηνύματος.*

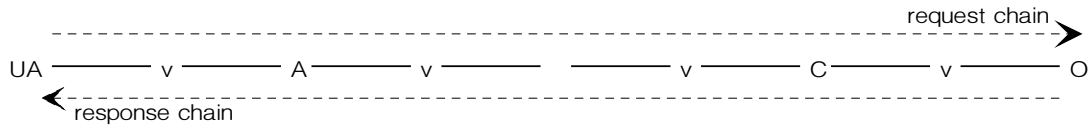
Η επικοινωνία HTTP συνήθως πραγματοποιείται μεταξύ ενός user agent (UA) και ενός origin server (O). Στην απλούστερη της μορφή αυτή η επικοινωνία πραγματοποιείται με μία μόνο σύνδεση (v) και έχει ως εξής:



Σχήμα 2: Απλή μορφή HTTP επικοινωνίας

Η επικοινωνία HTTP γίνεται περισσότερο σύνθετη όταν μεταξύ του UA και του O (request/response chain) παρεμβάλλονται ενδιάμεσοι (intermediaries). Αυτοί εμφανίζονται σε τρεις μορφές: proxy, gateway και tunnel.

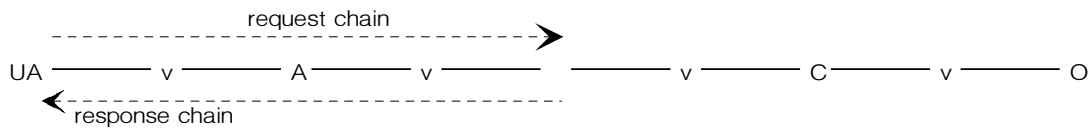
Ενας proxy ενδιάμεσος αποτελεί έναν πράκτορα προώθησης (forwarding agent) ο οποίος δέχεται αιτήσεις για κάποιο URI σε απόλυτη μορφή (absolute form), ανασκευάζει τα σχετικά μηνύματα μεταβάλλοντας όλα τα συστατικά τμήματα τους και τα προωθεί στον server ο οποίος προσδιορίζεται από το URI. Ενας gateway ενδιάμεσος αποτελεί ένα πράκτορα παραλαβής (receiving agent) ο οποίος τοποθετείται στο αμέσως υψηλότερο επίπεδο από ορισμένους servers και μεταφράζει τις αιτήσεις στο πρωτόκολλο που οι servers αυτοί αντιλαμβάνονται και μπορούν να ερμηνεύσουν. Ενας tunnel ενδιάμεσος λειτουργεί ως σημείο μεταγωγής (relay point) μεταξύ δύο συνδέσεων χωρίς να παρεμβαίνει στο περιεχόμενο των μηνυμάτων. Tunnels χρησιμοποιούνται όταν η επικοινωνία HTTP θα πρέπει να διέλθει από ενδιάμεσους όπως π.χ. firewalls.



Σχήμα 3: HTTP επικοινωνία με ενδιάμεσους

Στο σχήμα 3 παρουσιάζονται 3 ενδιάμεσοι (οι A, B και C) μεταξύ του UA και του O. Αυτοί μπορεί να εμπλέκονται σε περισσότερες από μία ταυτόχρονες HTTP επικοινωνίες δηλ. ο B μπορεί να δέχεται αιτήσεις και από άλλους clients πλην του A και να προωθεί αιτήσεις σε άλλους servers πλην του C ταυτόχρονα με την διεκπεραίωση μίας αίτησης του A.

Οποιοδήποτε μέρος (party) στην παραπάνω επικοινωνία που δεν δρα ως tunnel ενδιάμεσος μπορεί να αναπτύξει μία εσωτερική cache για την διεκπεραίωση των αιτήσεων που δέχεται. Το αποτέλεσμα εφαρμογής caching είναι ότι το μήκος της αλυσίδας request/response ελαττώνεται εάν ένας από τους συμμετέχοντες στην αλυσίδα διαθέτει μία cached απάντηση που μπορεί να χρησιμοποιηθεί για την μεταδιδόμενη αίτηση. Η περίπτωση αυτή παρουσιάζεται στο σχήμα 4 όπου ο B διαθέτει ένα αντίγραφο παλαιότερης απάντησης από τον O για μία συγκεκριμένη αίτηση ενώ το αντίγραφο αυτό δεν περιλαμβάνεται στις caches των A ή UA.



Σχήμα 4: Εφαρμογή caching στην HTTP επικοινωνία

Θα πρέπει να τονιστεί ότι υπάρχουν περιπτώσεις στις οποίες οι HTTP απαντήσεις δεν επιδέχονται caching.

Οπως επισημάνθηκε παραπάνω η HTTP επικοινωνία βασίζεται σε συνδέσεις του πρωτοκόλλου TCP/IP. Η εξ' ορισμού (default) TCP θύρα είναι η 80 αλλά δεν αποκλείεται η χρήση και άλλων. Επίσης δεν αποκλείεται η πραγματοποίηση της HTTP επικοινωνίας πάνω από άλλα πρωτόκολλα μεταφοράς στο Internet ή σε άλλα δίκτυα. Η μόνη προϋπόθεση που τίθεται από το HTTP για το πρωτόκολλο του δικτυακού υποστρώματος είναι η αξιόπιστη μεταφορά (reliable transport).

Γενικά, οι συνδέσεις εκκινούνται από τον client πριν από την αποστολή της αίτησης και τερματίζονται από τον server μετά την αποστολή της απάντησης (Σχήμα 1).

3. Μέθοδοι Αιτήσεων

Στην 1.0 έκδοση του HTTP υποστηρίζονται οι μέθοδοι GET, HEAD και PUT με τα εξής χαρακτηριστικά:

GET: Η μέθοδος GET αφορά στην ανάκτηση της οποιασδήποτε πληροφορίας (αντικειμένου) καθορίζεται από το URI της αίτησης (Request URI). Εάν το URI της αίτησης υποδεικνύει μία διαδικασία επεξεργασίας δεδομένων θα πρέπει να επιστραφούν, ως απάντηση, τα δεδομένα όπως αυτά προέκυψαν από την σχετική διαδικασία.

Μία αίτηση GET μπορεί να υποβληθεί υπό συγκεκριμένη συνθήκη (conditional GET). Στην περίπτωση αυτή, στην επικεφαλίδα της σχετικής αίτησης συμπεριλαμβάνεται το πεδίο If-modified-since. Το προσδιοριζόμενο αντικείμενο ανακτάται μόνο στην περίπτωση που η ημερομηνία της τελευταίας ενημέρωσης/ μεταβολής του είναι πιο πρόσφατη από την ημερομηνία που καθορίζεται από το πεδίο If-modified-since. Η δυνατότητα conditional GET στοχεύει στην ελαχιστοποίηση του δικτυακού φόρτου επιτρέποντας την χρήση των cached αντιγράφων στους clients. Με τον μηχανισμό αυτό αποφεύγεται η ανταλλαγή περιττών δεδομένων στις περιπτώσεις αντικειμένων που δεν διακρίνονται για τις συχνές μεταβολές τους.

HEAD: Η μέθοδος αυτή είναι τελείως ανάλογη με την GET. Χρησιμοποιείται για τον έλεγχο συνδέσμων υπερκειμένου (hypertext links) σχετικά με την δυνατότητα πρόσβασης τους, την εγκυρότητα καθώς και ενδεχόμενες πρόσφατες μεταβολές τους. Δεν προβλέπεται η δυνατότητα conditional HEAD.

Στην μέθοδο HEAD ο server δεν επιστρέφει, στην απάντηση του, το σώμα της προσδιοριζόμενης πληροφοριακής οντότητας (πεδίο Entity-body) παρά μόνο μεταπληροφορία για αυτήν. Η επιστρεφόμενη μεταπληροφορία είναι η ίδια με την

περίπτωση της μεθόδου GET. Οπως επισημάνθηκε παραπάνω χρησιμοποιείται κατά κύριο λόγο από τους browsers που εφαρμόζουν caching για την ανάκτηση αντικειμένων με βάση το πεδίο επικεφαλίδας Last-modified-since. Εάν η ημερομηνία αυτή είναι νεότερη από αυτή του αντικειμένου της cache ζητείται το περισσότερο πρόσφατο αντικείμενο. Οι μέθοδοι GET και HEAD έχει επικρατήσει να χρησιμοποιούνται μόνο για την ανάκτηση πληροφορίας (retrieval) και όχι για άλλες λειτουργίες.

POST: Η μέθοδος αυτή υποδεικνύει στον server να δεχτεί την οντότητα που μεταφέρεται στην αίτηση σαν ένα νέο στιγμιότυπο (εγγραφή, καταχώρηση) του πόρου που προσδιορίζεται από το URI. Η μέθοδος POST σχεδιάστηκε για την αντιμετώπιση αναγκών όπως:

- Υποβολή ενός μηνύματος σε μία bulletin board, newsgroup, mailing list ή παρόμοια συλλογή άρθρων.
- Πέρασμα παραμέτρων σε μία διαδικασία επεξεργασίας δεδομένων σαν αποτέλεσμα υποβολής φόρμας (form submission).
- Επέκταση μίας βάσης δεδομένων με την προσθήκη εγγραφών κα.

Η πραγματική λειτουργία η οποία εκτελείται σαν αποτέλεσμα της POST αίτησης προσδιορίζεται από τον server και συχνά εξαρτάται από το URI της αίτησης. Η οντότητα που περιέχεται στην αίτηση καθίσταται για τον πόρο που προσδιορίζεται στο URI ότι ένα αρχείο για τον υπερκείμενο κατάλογο που το περιλαμβάνει ή μία εγγραφή για την βάση δεδομένων στην οποία ανήκει. Οι απαντήσεις σε POST αιτήσεις δεν επιδέχονται caching.

Οι αιτήσεις POST μπορεί να μην έχουν σαν αποτέλεσμα πόρους που είναι προσπελάσιμοι σε μελλοντική αναφορά (αντιπροσωπεύονται από κάποιο URI). Το αποτέλεσμα των POST αιτήσεων (αναφορικά με τους πόρους που ενδεχομένως διαμορφώθηκαν) περιγράφεται στα success/error codes τα οποία επιστρέφονται από τον server.

4. Access Authentication

Το πρωτόκολλο HTTP παρέχει ένα απλό μηχανισμό access authentication ο οποίος βασίζεται στην ανταλλαγή προκλήσεων - απαντήσεων (challenge/responses) μεταξύ του client και του server. Το βασικό σχήμα authentication προβλέπει ότι ο πράκτορας χρήστη (UA) θα πρέπει να παρέχει ένα user-ID και ένα password για κάθε περιοχή (realm). Σε ένα server μπορούν να καθοριστούν πολλαπλές περιοχές η πρόσβαση στις οποίες θα πρέπει να ελέγχεται. Για κάθε προστατευόμενη περιοχή μπορεί να οριστεί

ένα συγκεκριμένο σχήμα authentication καθώς και η σχετική βάση δεδομένων (authentication database).

Οταν ο server λάβει μία αίτηση για πόρο μέσα στον χώρο προστασίας (protection space) απαντάει με μία πρόκληση. Στο τελευταίο αυτό μήνυμα προσδιορίζεται η ανάγκη authentication του χρήστη για την συγκεκριμένη περιοχή καθώς και το όνομα της περιοχής αυτής (realm name). Ο client απαντάει με το user-ID και το password σε κρυπτογραφημένο μήνυμα (base64 encoded).

Τα στοιχεία ενός χρήστη δεν απαιτείται να παρέχονται σε κάθε αίτηση που αφορά στην ίδια προστατευμένη περιοχή όταν έχει προηγουμένως επιτραπεί η πρόσβαση (authorised request). Με βάση το σχήμα authentication και διάφορες άλλες παραμέτρους λειτουργίας του HTTP server μπορεί να οριστεί ένα χρονικό διάστημα στο οποίο τα διαπιστευτήρια (credentials) του χρήστη για τον συγκεκριμένο χώρο μπορούν να παραμείνουν εν ισχύ.

Το βασικό σχήμα authentication δεν αποτελεί μία ασφαλή μέθοδο για τον έλεγχο πρόσβασης στους πόρους ενός HTTP server. Βασίζεται στην υπόθεση ότι μεταξύ του client και του server παρεμβάλλεται μια έμπιστη υπηρεσία μεταφοράς (trusted carrier). Αυτή η υπόθεση, όμως, δεν ανταποκρίνεται στην πραγματικότητα σε ανοικτά δίκτυα ευρείας περιοχής όπως το Internet.

Το HTTP δεν περιορίζει τις εφαρμογές WWW στον απλό μηχανισμό challenge/response που περιγράφηκε παραπάνω για το θέμα access authentication. Αλλοι περισσότερο εξελιγμένοι μηχανισμοί μπορούν να χρησιμοποιηθούν όπως η κρυπτογράφηση στο επίπεδο μεταφοράς, η ενθυλάκωση (encapsulation) μηνυμάτων κ.α. Η κρυπτογράφηση σε χαμηλότερο επίπεδο υιοθετείται από τον Commerce Server της Netscape Communications Co. μέσω του Secure Sockets Layer (SSL). Τα χαρακτηριστικά και οι δυνατότητες του SSL θα αποτελέσουν αντικείμενο επόμενης παραγράφου.

5. Πεδία επικεφαλίδας

Στην παράγραφο αυτή παρουσιάζονται τα σημαντικότερα από τα πεδία που συγκροτούν την επικεφαλίδα των HTTP μηνυμάτων.

Allow: Προσδιορίζει το σύνολο των μεθόδων που είναι εφαρμόσιμες στον πόρο που υποδεικνύεται από το URI της αίτησης. Δεν έχει δεσμευτικό χαρακτήρα για τον client.

Authorization: Επιστρέφει στον server πληροφορία authentication ύστερα από μήνυμα challenge. Ο μηχανισμός authentication περιγράφηκε στην προηγούμενη παράγραφο.

Expires: Το πεδίο αυτό προσδιορίζει την ημερομηνία/ώρα ύστερα από την οποία η υποδεικνυόμενη οντότητα θα πρέπει να θεωρηθεί παρωχημένη (stale) και μη έγκυρη. Οι εφαρμογές δεν θα πρέπει να εφαρμόζουν caching σε παρόμοια στοιχεία μετά την ημερομηνία αυτή. Η παρουσία του πεδίου δεν σημαίνει κατά ανάγκη ότι η αρχική οντότητα θα πάψει να υπάρχει ή θα αλλάξει μετά την προσδιοριζόμενη ημερομηνία. Η κύρια λειτουργικότητα του πεδίου αφορά στον μηχανισμό caching. Επίσης επιτρέπει στον παροχέα της πληροφορίας να προσδιορίσει το ευμετάβλητο (volatility) των πόρων.

From: Το πεδίο αυτό περιέχει την διεύθυνση ηλεκτρονικού ταχυδρομείου (e-mail address) του χρήστη που ελέγχει τον αιτούμενο UA. Από την πλευρά του server χρησιμοποιείται για την καταγραφή κίνησης χρηστών (user logging).

If-modified-since: Η χρήση αυτού του πεδίου επικεφαλίδας περιγράφηκε παραπάνω, σε σχέση με την μέθοδο conditional GET.

Last modified: Το πεδίο αυτό υποδεικνύει την ημέρα και ώρα στην οποία ο server πιστεύει ότι υπήρξε μεταβολή του περιεχομένου του πόρου. Ο client αποδέκτης της πληροφορίας την συγκρίνει με το αντίγραφο που ενδεχομένως διατηρεί στην cache του και ενεργεί κατάλληλα.

Referer: Επιτρέπει σε ένα client να προσδιορίσει το URI του πόρου από τον οποίο προήλθε το αιτούμενο URI (Request-URI). Επιτρέπει στο server να διαμορφώσει καταλόγους τέτοιων URIs για την βελτιστοποίηση του μηχανισμού caching καθώς επίσης και για την συντήρηση του υπάρχοντος web.

Pragma: Μέσω του πεδίου αυτού μεταδίδονται οδηγίες (directives) προς τα μέλη της αλυσίδας request/response για την διεκπεραίωση συνδέσεων. Οταν μεταδίδεται η οδηγία "no-cache", σε μία αίτηση, η εφαρμογή θα πρέπει να την προωθήσει προς τον server (Ο) ακόμα και αν διατηρεί ένα αντίγραφο του ζητούμενου URI. Οι οδηγίες θα πρέπει να περάσουν από ένα proxy ή gateway άσχετα από την σημασία που έχουν για τις εφαρμογές αυτές και να φτάσουν σε όλα τα μέλη της request/ response αλυσίδας. Δεν είναι δυνατή η αποστολή οδηγιών προς ένα μόνο μέλος της αλυσίδας.

6. Secure Sockets Layer (SSL)

Το πρωτόκολλο SSL στοχεύει στην διασφάλιση της μυστικότητας (privacy) και της αξιοπιστίας (reliability) στην επικοινωνία μεταξύ δύο εφαρμογών. Αναπτύχθηκε από την Netscape Communications, υποβλήθηκε στο W3C ως πρόταση της εταιρίας για την υιοθέτηση του ως προτύπου, και είναι διαθέσιμο σε μορφή Internet Draft [4]. Σχεδιάστηκε για την υποστήριξη πρωτοκόλλων επιπέδου εφαρμογής όπως τα HTTP, NNTP, FTP και Telnet. Αποτελείται από δύο επίπεδα (layers). Στο κατώτερο επίπεδο

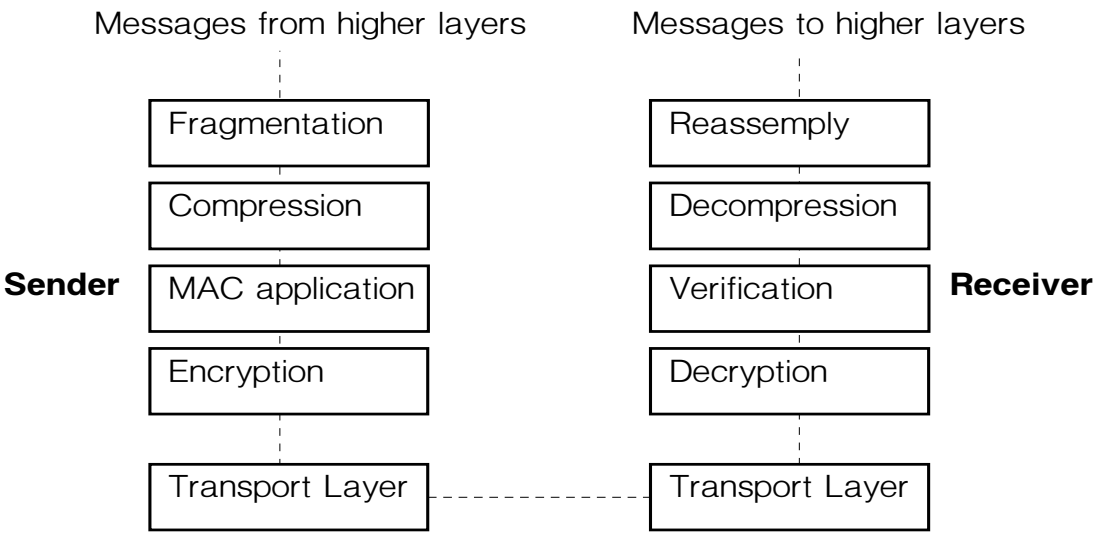
του SSL τοποθετείται το SSL Record Protocol. Το SSL Record Protocol προϋποθέτει για την λειτουργία του ένα αξιόπιστο πρωτόκολλο μεταφοράς όπως το TCP και χρησιμοποιείται για την ενθυλάκωση (encapsulation) πρωτοκόλλων υψηλότερου επιπέδου. Ενα από αυτά τα επίπεδα είναι το SSL Handshake Protocol. Το τελευταίο πρωτόκολλο επιτρέπει στον server και στον client να πιστοποιήσουν (authenticate) ο ένας τον άλλο (κάνοντας χρήση digital signature και certificate) και να διαπραγματευτούν αλγόριθμο και κλειδιά κρυπτογράφησης. Η διαπραγμάτευση αυτή γίνεται πριν την ανταλλαγή δεδομένων μεταξύ πρωτοκόλλων επιπέδου εφαρμογής. Το βασικό πλεονέκτημα του SSL είναι η ανεξαρτησία του από τα πρωτόκολλα αυτά.

Το πρωτόκολλο SSL παρέχει ασφάλεια σύνδεσης (connection security) η οποία έχει τρεις βασικές ιδιότητες:

- *Η σύνδεση είναι ιδιωτική. Κρυπτογράφηση χρησιμοποιείται μετά από την αρχική χειραψία (handshake) για τον καθορισμό ενός μυστικού κλειδιού. Για την κρυπτογράφηση των δεδομένων χρησιμοποιούνται συμμετρικοί αλγόριθμοι όπως DES, RC4 κλπ.*
- *Η σύνδεση μπορεί να πιστοποιηθεί χρησιμοποιώντας μη-συμμετρικό αλγόριθμο κρυπτογράφησης (ή δημόσιου κλειδιού) όπως RSA, DSS κλπ..*
- *Η σύνδεση είναι αξιόπιστη Στην μεταφορά των μηνυμάτων περιλαμβάνεται έλεγχος εγκυρότητας (integrity check) με την χρήση αλγορίθμου MAC (Message Authenticity Check) βάσει κλειδιών (keyed MAC).*

Μία SSL σύνοδος διαθέτει μνήμη (stateful session). Ο συντονισμός των καταστάσεων του client και του server αποτελεί ευθύνη του SSL Handshake Protocol. Μία σύνοδος μπορεί να περιέχει πολλαπλές ασφαλείς συνδέσεις.

Το επίπεδο SSL Record δέχεται δεδομένα (μηνύματα) από τα ανώτερα στρώματα σε μη-κενά blocks που δεν έχουν κάποιο συγκεκριμένο μήκος. Τα blocks αυτά κερματίζονται (fragmentation) από το επίπεδο σε εγγραφές (SSLPlaintext records) μέγιστου μήκους 214 bytes. Πολλαπλά μηνύματα ανωτέρων πρωτοκόλλων μπορούν να συμπεριληφθούν σε μία τέτοια εγγραφή. Οι εγγραφές συμπιέζονται (προαιρετικά) κάνοντας χρήση του αλγορίθμου που έχει οριστεί στην τρέχουσα κατάσταση της συνόδου. Με την εφαρμογή της συμπίεσης οι SSLPlaintext εγγραφές μετασχηματίζονται σε SSLCompressed δομές. Η συμπίεση είναι lossless. Ο γενικός μηχανισμός λειτουργίας του SSL παρουσιάζεται στο Σχήμα 5 που ακολουθεί:



Σχήμα 5: Ανταλλαγή μηνυμάτων μέσω SSL

7. Secure HyperText Transfer Protocol (S-HTTP)

Το πρωτόκολλο S-HTTP [5] αποτελεί μία επέκταση του πρωτοκόλλου HTTP το οποίο παρουσιάστηκε στις παραγράφους που προηγήθηκαν. Παρέχει υπηρεσίες ασφαλείας που μπορούν να εφαρμοστούν μεμονωμένα με στόχο την εμπιστευτικότητα των δοσοληψιών (transaction confidentiality) καθώς και την αυθεντικότητα/ακεραιότητα (authenticity/integrity) της πηγής της μεταδιδόμενης πληροφορίας. Το πρωτόκολλο επιτρέπει την διαπραγμάτευση (σε επίπεδο δοσοληψίας) μεταξύ των επικοινωνούντων μερών των αλγορίθμων κρυπτογράφησης, διαχείρισης κλειδιών καθώς και άλλων παραμέτρων ασφαλείας. Το S-HTTP είναι συμβατό με το HTTP.

Στους client/servers που υποστηρίζουν το πρωτόκολλο S-HTTP μπορούν να ενσωματωθούν πολλαπλά πρότυπα αναφορικά με την μορφή των κρυπτογραφημένων μηνυμάτων. Ιδιαίτερα δημοφιλή μεταξύ αυτών είναι τα PKCS-7 και PEM. Το πρωτόκολλο δεν απαιτεί public key certificate στην πλευρά του client. Ετσι, ορισμένες προσωπικές δοσοληψίες μπορούν να πραγματοποιηθούν χωρίς να απαιτείται από τους μεμονωμένους χρήστες να έχουν εξασφαλίσει κάποιο δημόσιο κλειδί. Οι ασφαλείς δοσοληψίες πραγματοποιούνται από άκρο σε άκρο (end-to-end) σε αντίθεση με το HTTP. Οι μηχανισμοί authorisation του HTTP απαιτούσαν από τον client να επιχειρήσει την προσπάθεια πρόσβασης η οποία ενδεχομένως να απορρίπτονταν πριν να ενεργοποιηθεί κάποιος μηχανισμός διασφάλισης της επικοινωνίας.

Στο S-HTTP ο client με τον server μπορούν να διαπραγματευτούν τις παραμέτρους της ασφαλούς επικοινωνίας. Ενδεικτικά αναφέρονται οι παρακάτω περιοχές διαπραγμάτευσης:

Τρόπος εκτέλεσης της δοσοληψίας

- *Θα πρέπει οι αιτήσεις να φέρουν υπογραφή?*
- *Θα πρέπει οι αιτήσεις να είναι κρυπτογραφημένες?*
- *Χρειάζονται συνδυασμός των δύο παραπάνω τακτικών?*

Αλγόριθμοι κρυπτογράφησης

- *Θα χρησιμοποιηθεί ο RSA ή ο DSA αλγόριθμος για την μεταφορά υπογραφών?*
- *Θα χρησιμοποιηθεί ο DES ή ο RC2 για την κρυπτογράφηση?*

Επιλογή certificate

- *Χρησιμοποίηση του Mastercard certificate?*

Τα μηνύματα που ανταλλάσσονται μέσω του πρωτοκόλλου μπορούν να εξασφαλιστούν με: υπογραφή (signature), κρυπτογράφηση (encryption) και πιστοποίηση (authentication). Επίσης είναι δυνατός ο οποιοσδήποτε συνδυασμός των τριών.

8. HTTP Deamons

Οι WWW servers φέρουν εξειδικευμένο λογισμικό για την παραλαβή και διεκπεραίωση των αιτήσεων που υποβάλλονται μέσω του πρωτοκόλλου HTTP. Αυτό το λογισμικό καλείται HTTPD (HTTP Deamon). Διακρίνονται δύο κατηγορίες HTTPDs: αυτοί που είναι διαθέσιμοι στο δίκτυο Internet (συνήθως μαζί με τον πηγαίο κωδικά τους) καθώς και τα προϊόντα που είναι εμπορικά διαθέσιμα. Βασικοί αντιπρόσωποι στην πρώτη κατηγορία είναι ο CERN και ο NCSA HTTPD. Για την δεύτερη κατηγορία αξίζει να αναφερθούν ο Commerce Server της Netscape Communications Co. και ο Website της εταιρίας OReilly. Στις παραγράφους που ακολουθούν πρόκειται να παρουσιαστούν, εν συντομία, οι παραπάνω εφαρμογές.

Ο CERN server αποτελεί τον πρώτο HTTP server που κατασκευάστηκε ποτέ. Αναπτύχθηκε από την ομάδα των επιστημόνων που συνέλαβαν και υλοποίησαν την έννοια του WWW. Διαθέτει πολλαπλές δυνατότητες με πλέον ενδιαφέρουσα την λειτουργία του ως proxy server. Η έννοια του proxy ενδιάμεσου αναλύθηκε στην παράγραφο 2. Επίσης ο CERN server μπορεί να πραγματοποιήσει caching στα διακινούμενα αντικείμενα. Ο CERN server είναι διαθέσιμος τόσο σε μορφή πηγαίου κώδικα όσο και σε μορφή precompiled binary για διάφορα συστήματα UNIX. Αποτέλεσε την βάση για την κατασκευή πολλών άλλων HTTP deamons.

Ο NCSA HTTPD είναι ο πλέον δημοφιλής public domain server στις ΗΠΑ. Προέρχεται από το ερευνητικό κέντρο στο οποίο αναπτύχθηκε ο Mosaic browser με τις νεότερες εκδόσεις του οποίου μπορεί να εγκαθιδρύσει ασφαλείς συνδέσεις (κάνοντας χρήση

των προγραμμάτων ασφαλείας PEM και PGP). Η τρέχουσα έκδοση 1.5 του server ενσωματώνει δυναμική διαχείριση διεργασιών και δυνατότητα user authentication σε διάφορα επίπεδα (file, directory, κα.). Ο server είναι διαθέσιμος public domain σε μορφή πηγαίου κώδικα αλλά και precompiled binaries. Οι επιδόσεις του κρίνονται άριστες και συγκρίσιμες με εμπορικά εργαλεία όπως είναι οι servers της Netscape Communications. Η έκδοση 1.3 του NCSA HTTPD είχε μεταφερθεί και στην πλατφόρμα MS-Windows 3.1 υποστηρίζοντας (με ορισμένες διαφοροποιήσεις) CGI scripts που είχαν γραφτεί για UNIX περιβάλλον. Το λογισμικό αυτό μπορούσε να υποστηρίξει 8 ταυτόχρονους χρήστες.

Ο Commerce Server της Netscape Communications Co. καλύπτει ανάγκες ασφαλούς ηλεκτρονικού εμπορίου και επικοινωνιών αφού ενσωματώνει το πρωτόκολλο SSL (ονομάζεται και Secure server). Πέρα από τα θέματα ασφαλείας, το προϊόν είναι το ίδιο ακριβώς με το Communication Server της ίδιας εταιρίας. Είναι διαθέσιμος για διάφορες hardware πλατφόρμες και λειτουργικά συστήματα. Ενδεικτικά αναφέρονται τα SUN Sparc Solaris, IBM AIX, HP-UX, DEC OSF και MS-Windows NT (Intel, Alpha). Το προϊόν είναι ιδιαίτερα εύκολο στην εγκατάσταση του η οποία γίνεται μέσα από τον Netscape Navigator WWW browser σε γραφικό περιβάλλον. Επίσης, μέσα από το ίδιο πρόγραμμα γίνεται η όλη διαχείριση του server (κάνοντας χρήση του NSAPI το οποίο πρόκειται να παρουσιαστεί σε επόμενο κεφάλαιο). Ουσιαστικά, μετά την διαδικασία εγκατάστασης υπάρχουν δύο HTTP servers: πέραν του κανονικού server ο οποίος δέχεται και διεκπεραιώνει τις συνήθεις HTTP αιτήσεις υπάρχει και ο administration server. Ο administration server, που τίθεται διαρκώς σε λειτουργία, δέχεται αιτήσεις σε συγκεκριμένη TCP θύρα η οποία βέβαια διαφέρει από αυτή που χρησιμοποιεί ο κανονικός server. Μέσω του administration server είναι δυνατή η ενεργοποίηση και η παύση του κανονικού server καθώς και η ρύθμιση των παραμέτρων λειτουργίας του (server configuration). Η εισαγωγή του administration server επιτρέπει την πλήρη διαχείριση του όλου συστήματος από έναν απλό WWW browser εξ αποστάσεως (μέσω LAN/WAN).

Ο Commerce Server υποστηρίζει δυναμική διαχείριση διεργασιών για την αποδοτικότερη εξυπηρέτηση σε περιόδους φόρτου. Οταν εκκινηθεί ο Netscape server στην μνήμη του συστήματος μπορούν να αναπτυχθούν περισσότερες από μία σχετικές διεργασίες (HTTP deamons) που είναι ακριβή αντίγραφα μίας βασικής. Οταν το σύστημα δεχτεί ταυτόχρονες αιτήσεις αυτές διεκπεραιώνονται από διαφορετικές διεργασίες. Αν οι ταυτόχρονες αιτήσεις υπερβούν τον αρχικό αριθμό των διεργασιών, ο διαχειριστής διεργασιών μπορεί να πυροδοτήσει την γέννηση νέων αντιγράφων (τα οποία παραμένουν ενεργά μετά την χρήση τους μέχρι την παύση του server). Η ίδια δυνατότητα διατίθεται και στον NCSA HTTPD που παρουσιάστηκε παραπάνω. Ο server υποστηρίζει τα πρωτόκολλα SSL, HTTP, CGI ενώ επίσης ενσωματώνει και το

Netscape Server Application Programming Interface (NSAPI) για την επέκταση ή μεταβολή της βασικής λειτουργικότητας του (authentication, logging, gateways κλπ.). Πέρα από την ασφάλεια του SSL ο Commerce Server μπορεί να πραγματοποιήσει access authentication σύμφωνα με την περιγραφή της παραγράφου 4.

Ο Website της εταιρίας OReilly αποτελεί ένα HTTPD εύκολο στην χρήση και στην εγκατάσταση του. Το προϊόν είναι διαθέσιμο για την πλατφόρμα Windows NT αλλά μπορεί να εκτελεστεί και σε περιβάλλον Windows 95. Συνοδεύεται από εξωτερικό (σε αντίθεση με τον Netscape Server) λογισμικό που καλείται Server Admin. Το λογισμικό αυτό επιτρέπει τον καθορισμό όλων των παραμέτρων λειτουργίας του server (access control, logging, directory mapping κα.). Αλλο λογισμικό που συνοδεύει τον Website είναι το WebView. Το WebView είναι ένα φιλικό προς το χρήστη εργαλείο που χρησιμοποιείται για την ανοικοδόμηση και την συντήρηση του τοπικού ιστού (local web). Διαχειρίζεται τα τοπικά αρχεία HTML καθώς και τους συνδέσμους υπερκειμένου που ενσωματώνονται σε αυτά.

Ενα πολύ σημαντικό χαρακτηριστικό του server της OReilly είναι η αυτόματη δεικτοδότηση (indexing) των περιεχομένων του ιστού. Προς τον σκοπό αυτό το προϊόν ενσωματώνει 2 υποστηρικτικά εργαλεία (utilities): τα WebIndex και WebFind. Τα δύο προγράμματα συνεργάζονται για την παροχή της δυνατότητας full-text search στους χρήστες του server. Μέσω του WebIndex είναι δυνατή η οριοθέτηση του τμήματος του ιστού στο οποίο οι χρήστες μπορούν να εκτελέσουν αναζητήσεις (searchable web). Το WebFind σε αντίθεση με το WebIndex αποτελεί μία εφαρμογή CGI. Το WebFind παρουσιάζει στους χρήστες προς συμπλήρωση μία φόρμα αναζήτησης. Στην συνέχεια το πρόγραμμα εκτελεί την προδιαγεγραμμένη αναζήτηση στον τοπικό ιστό.

Ο Website υποστηρίζει τους συνήθεις μηχανισμούς user authentication και access control. Επίσης υποστηρίζει το πρωτόκολλο CGI καθώς και 2 διαφοροποιήσεις αυτού. Ο Website υποστηρίζει το Windows CGI (μέσω του οποίου μπορούν να αναπτυχθούν εφαρμογές σε Visual Basic, Borland Delphi, Visual C/C++ κλπ.), το POSIX CGI (για εφαρμογές που έχουν γραφτεί για συστήματα UNIX) και τέλος το τυποποιημένο CGI (για Perl scripts, Korn shell scripts και console εφαρμογές).

Ο πίνακας που ακολουθεί προέρχεται από το [6] και συνοψίζει τα βασικά χαρακτηριστικά ορισμένων από τους HTTP servers.

	CERN	NCSA	Plexus	WN	GN	Netsite	Webserver
Protocols							
HTTP/0.9	yes	yes	yes	yes	yes	yes	yes
HTTP/1.0	yes	yes	yes	yes	yes	yes	yes
Gopher	no	no	no	no	yes	no	no
Communications							
Service multiple connections	yes	yes	yes	yes	yes	yes	yes
Limit no of connections	no	no	yes	no	no	yes	yes

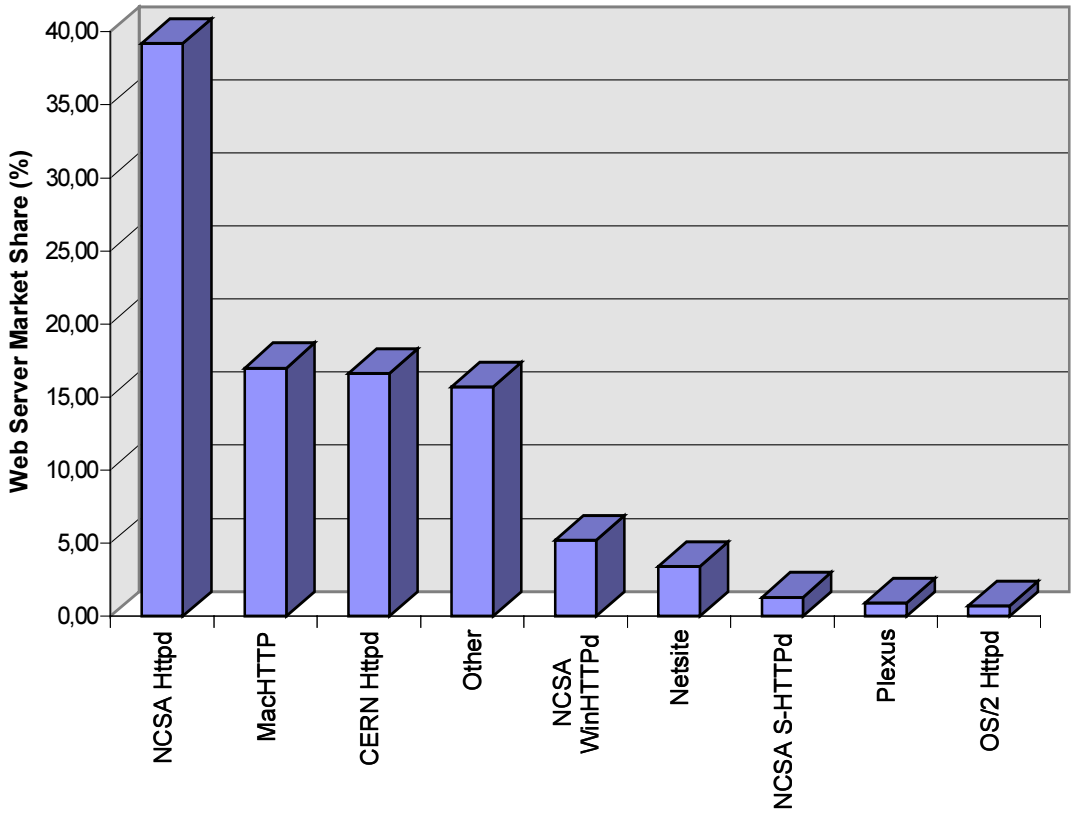
	CERN	NCSA	Plexus	WN	GN	Netsite	Webserver
Can service multiple ports	no	no	yes	no	yes	no	no
Mutiple homed hosts	no	no	yes	no	no	no	no
Performance	good	excellent	good	good	good	excellent	good
Access control							
Access control by IP	yes	yes	yes	yes	yes	yes	yes
Access control by password	yes	yes	limited	no	no	yes	yes
Authorised user lists	yes	yes	no	no	no	yes	yes
Authorised group lists	yes	yes	no	no	no	yes	yes
Per directory authorisation	yes	yes	yes	yes	yes	yes	yes
Per file authorisation	yes	no	yes	yes	yes	no	yes
Reliable user authentication	yes	yes	limited	no	no	yes	yes
Proxy support							
Proxy support	yes	no	no	no	no	no	no
Caching	yes	no	no	no	no	no	no
Scripts							
Executable scripts	yes	yes	yes	yes	yes	yes	yes
Executable scripts anywhere	no	yes	no	yes	yes	yes	yes
Server side includes	no	yes	yes	yes	no	yes	limited
Clickable images	yes	yes	limited	yes	no	yes	yes
Fill-out forms	yes	yes	limited	no	no	yes	yes

Στον πίνακα που ακολουθεί εκθέτονται τα βασικότερα χαρακτηριστικά των WWW servers που είναι διαθέσιμοι για το περιβάλλον Windows NT.

Platform/ Features	Website O'Reilly	Naviserver Navisoft	Netscape Communications	Purveyor Process	WebServer Quarterdeck
Windows 95	Yes	No	No	Yes	16-bit only
Windows NT	Yes	Yes	Yes	Yes	NT3.5 only
Full CGI with VB toolkit	Yes	No	No	No	No
Tree display, link verify	Yes	No	No	No	No
Remote administration	Yes	Yes	Yes	NT only	No
Application	Both	Service only	Service only	Service only	Service only
Multi-homing	Yes	Yes	No	NT only	No
Image map maker	Yes	No	No	No	No
Integrated keyword indexing	Yes	Yes	No	No	No
Integrated search engine	Yes	Yes	No	No	No
Wizards	Yes	No	No	No	No

Source: http://clubweb.ora.com/Compare1.htm

Στο σχήμα 6 παρουσιάζονται τα μερίδια αγοράς των διαφόρων WWW servers [7].



Source: Internet WWW Site Survey

Σχήμα 6: Μερίδια αγοράς των WWW servers

Βιβλιογραφία

[1] Berners-Lee T., Fielding R.T. and Frystyk Nielsen H.: *Hypertext Transfer Protocol - HTTP/1.0*, HTTP Working Group, Internet Draft (October 1995).

[2] Berners-Lee T., Cailliau R., Luotonen A., Frystyk Nielsen H. and Secret A.: The World -Wide Web, Communications of the ACM, Vol.37 No.8. (August 1994).

[3] December John and Ginsburg M.: *HTML & CGI Unleashed*, Sams.net Publishing (1995).

[4] Freier A., Kocher P. and Karlton P.: Secure Sockets Layer Version 3.0, Netscape Communications Co. (December 1995).

[5] Rescorla E. and Schiffman A.: *The Secure HyperText Transfer Protocol*, Web Transaction Security Working Group, Internet Draft (July 1995).

[6] Stein Lincoln: *How to Set Up and Maintain a World Wide Web Site*, Addison-Wesley (1995).

[7] Χάρισμας Δ.: Επιχειρηματικοί Internet Servers, NET LETTER, Ετος 2, Τεύχος 2, (Φεβρ.1996)