ΤΜΗΜΑ ΤΗΛΕΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ
Τ.Ε.Ι. ΗΠΕΙΡΟΥ - ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ

Τίτλος Πτυχιακής:Διαδικασία Πιστοποίησης Χρηστών Μέσω PHP.

Υπεύθυνοι Πτυχιακής: **Τασούλης Αποστόλης.**
Χειλάκης Γιώργος.

Άρτα 4 Μαρτίου 2005

ΠΕΡΙΕΧΟΜΕΝΑ

Κεφάλαιο 3ο

Εισαγωγή

Με την ανάπτυξη και την εξάπλωση των WWW εφαρμογών σε περιβάλλοντα δικτύων ηλεκτρονικών υπολογιστών, έχει αρχίσει να γίνεται σήμερα επιτακτική η ανάγκη για ασφαλή διακίνηση πληροφοριών σε όλο και περισσότερο ευρύ φάσμα εφαρμογών και είδη χρηστών.

Σκοπός αυτής της πτυχιακής εργασίας είναι να καταγράψει την διαδικασία πιστοποίησης χρηστών μέσω της γλώσσας PHP. Θα αναφερθούμε σε θέματα ασφάλειας διαδικτύου όπως ποιοι οι κίνδυνοι στο διαδίκτυο,ποιοι οι λόγοι που οδηγούν στη ανάγκη πιστοποιήσης και όχι ανώνυμης πρόσβασης σε διάφορες διαδικτυακές υπηρεσίες.

Στη συνέχεια θα αναλύσουμε ποια είναι η γλώσσα PHP, αναφέροντας τα πλεονεκτήματα που μας προσφέρει σε σχέση με άλλες γλώσσες και κάνοντας μια αναφορά σχετικά με τον τρόπο με τον οποίο αυτή λειτουργεί. Τέλος θα γίνει μια αναφορά σε διάφορους τρόπους που μπορεί αυτή η γλώσσα να προσφέρει πιστοποίηση χρηστών παρουσιάζοντας μερικά παραδείγματα.

ν **ΚΕΦΑΛΑΙΟ 1°**

1. Χαρακτηριστικα Του Διαδικτύου

Στη καθημερινότητα του σύγχρονου ατόμου έχει προστεθεί τα τελευταία χρόνια η χρήση του Διαδικτύου (Internet), για επαγγελματική ή ιδιωτική χρήση. Συναντάται στην εργασία, την εκπαίδευση, τη ψυχαγωγία, την ενημέρωση ή στις συναλλαγές « σταδιακά δε σε όλο το φάσμα της ιδιωτικής και δημόσιας επικοινωνίας.

Το εύρος και η ταχύτητα διάδοσής του αποδίδονται κυρίως σε ένα χαρακτηριστικό του: **την απλότητά που φέρει**. Αυτή χαρακτηρίζει τη δομή του, τον μηχανισμό λειτουργίας του και τα τεχνικά μέσα υποστήριξής του: αρκεί κανείς να αναλογιστεί ότι το διαδίκτυο είναι απλά συνδεδεμένοι υπολογιστές, συνήθως μέσω υφιστάμενου τηλεφωνικού δικτύου, οι οποίοι ανταλλάσσουν μεταξύ τους δεδομένα απαντώντας ο ένας σε απλές ερωτήσεις του άλλου (π.χ. Ποιος είσαι; Τι δεδομένα θέλεις;).

Χάρη στην απλότητά με την οποία παρέχει υπηρεσίες ταχύτατης και φθηνής ψηφιακής επικοινωνίας, έχει επιτύχει να υποστηρίζει πολλαπλές εφαρμογές και να χρησιμοποιείται στη γραπτή, προφορική ή οπτική επικοινωνία για ποικίλες χρήσεις: από την ανταλλαγή ευχετήριων ηλεκτρονικών καρτών ως τη συμπλήρωση και αποστολή φορολογικής δήλωσης εισοδήματος στην εφορία.

Ωστόσο στο ίδιο αυτό χαρακτηριστικό της απλότητας του Διαδικτύου αποδίδεται η αδυναμία του να παρέχει υπηρεσίες με ενδογενή ασφάλεια: δηλαδή υπηρεσίες επικοινωνίας που παράγονται με τρόπο τέτοιο ώστε να εξασφαλίζουν στον χρήστη ικανοποιητικά επίπεδα ασφάλειας, χωρίς εκείνος να απαιτείται να παίρνει πρόσθετα μέσα προστασίας. Για παράδειγμα το διαδίκτυο δεν έχει σχεδιαστεί με τρόπο τέτοιο ώστε να εξασφαλίζεται η πηγή προέλευσης δεδομένων, με αποτέλεσμα μπορεί ο επιτιθέμενος σε ένα σύστημα να κρύψει τα ίχνη της επίθεσής του.

Παράλληλα, αναγνωρίζονται στον παγκόσμιο ιστό πρόσθετα χαρακτηριστικά που συμβάλλουν στον αυξημένο βαθμό επικινδυνότητας της χρήσης του:

- ***Νέες Τεχνολογίες:*** Η χρήση διαρκώς νέων τεχνολογιών συχνά συνοδεύεται από κενά σε θέματα ασφάλειας. Έως ότου φθάσουν σε τεχνολογική ωρίμανση, προσφέρουν πρόσφορο έδαφος για κακή χρήση ή εκμετάλλευση των αδυναμιών τους.

- ***Διεπαφή Πολλαπλών Συστημάτων:*** Στο διαδίκτυο συνυπάρχουν και επικοινωνούν πολλαπλές εφαρμογές και τεχνολογίες για τη παραγωγή διαδικτυακών υπηρεσιών. Αυτό υπό περιπτώσεις οδηγεί σε μείωση της αποδοτικότητας μηχανισμών εφαρμογής μέτρων ασφαλούς πλοήγησης, σε όλο το εύρος των χρησιμοποιούμενων τεχνολογιών και εφαρμογών.

- ***Ελλιπές Διεθνές Θεσμικό Πλαίσιο:*** Το θεσμικό πλαίσιο για την προστασία των χρηστών του διαδικτύου αποτελεί ένα ιδιόμορφο μωσαϊκό από διεθνικές πολιτικές που επιχειρούν να εξασφαλίσουν την προστασία των χρηστών σε ένα δίκτυο υπολογιστών όπου δεν υπάρχουν διακριτά σύνορα και εθνικά όρια. Οι υφιστάμενοι κανόνες μπορούν να χαρακτηριστούν από σχετική επάρκεια, ωστόσο δεν διαθέτουν ακόμα ευελιξία στο εύρος των θεμάτων που καλύπτουν και στην εφαρμογή τους, ώστε να διασφαλίζονται πλήρως οι χρήστες του διαδικτύου.

1.1 Κίνδυνοι Χρήσης Του Διαδικτύου.

Αντικείμενα Απειλών.

Οι κίνδυνοι στους οποίους εκτίθεται ο μέσος χρήστης του διαδικτύου έχουν ως άμεσο στόχο δυο κυρίως αντικείμενα: τα διακινούμενα δεδομένα και τη διαθεσιμότητα των διαδικτυακών υπηρεσιών.

Διακινούμενα Δεδομένα.

Η επικοινωνία μέσω διαδικτύου στηρίζεται στη ψηφιακή μεταφορά δεδομένων με τη χρήση πολλαπλών τεχνολογιών: ηλεκτρονικού ταχυδρομείου (: e-mails), ομάδες συζήτησης (: chat forums), ηλεκτρονικές φόρμες σε ιστοσελίδες ή άμεση φωνητική και οπτική επικοινωνία.

Τα δεδομένα εκείνα που ως χρήστες αποστέλλουμε στο διαδίκτυο, εκτίθενται σε κινδύνους που αφορούν, κατ' ελάχιστον, την εμπιστευτικότητα της επικοινωνίας και την ακεραιότητα των δεδομένων.

Πρόσθετα σε αυτά, διακινούνται συστημικά δεδομένα που παράγονται από τα συστήματα επικοινωνίας που χρησιμοποιεί ο χρήστης στο διαδίκτυο. Πρόκειται για δεδομένα που αφενός δεν ελέγχονται άμεσα από τον χρήστη, αφετέρου περιέχουν συχνά σημαντικές πληροφορίες που χρήζουν ασφάλειας.

Διαθεσιμότητα Διαδικτυακών Υπηρεσιών.

Στο διαδίκτυο διατίθεται μια πλειάδα ψηφιακών υπηρεσιών, όπως αυτές των μηχανών αναζήτησης (Search Engines), της εκτέλεσης on line εμπορικών ή χρηματοπιστωτικών συναλλαγών ή της παροχής ειδήσεων.

Η μη ορθή παροχή των υπηρεσιών αυτών καθώς επίσης η πλήρης αδυναμία διάθεσης τους, αποτελεί κίνδυνο που μειώνει σημαντικά τόσο τη χρησιμότητα του παγκόσμιου ιστού όσο και την αξιοπιστία του.

Αυτό γίνεται περισσότερο αντιληπτό τα τελευταία χρόνια που έχουν αυξηθεί οι επιθέσεις τύπου Denial Of Service, οι οποίες έχουν ως στόχο να πλήξουν έναν δικτυακό τόπο και να τον καταστήσουν ανενεργό.

Αντίστοιχη απειλή αποτελεί η εκμετάλλευση των διαθέσιμων υπηρεσιών για σκοπό διαφορετικό από αυτόν για τον οποίο έχουν σχεδιαστεί ή από χρήστες που δεν φέρουν εξουσιοδότηση χρήσης του. Για παράδειγμα, χρησιμοποιούνται οι δυνατότητες που φέρει ένα πληροφοριακό σύστημα, συνδεδεμένο στο διαδίκτυο, για να προκληθούν επιθέσεις σε άλλα συστήματα εμφανίζοντας αυτά (και τους χρήστες τους) ως πηγή προέλευσης των επιθέσεων.

1.2 Μορφές Και Πηγές Απειλών – Κινδύνων.

Ποιες είναι τελικά οι απειλές που καλείται να αντιμετωπίσει ένας χρήστης στο διαδίκτυο; Στις επόμενες παραγράφους παρατίθενται οι πλέον διαδεδομένες μορφές απειλών, εκ των οποίων ιδιαίτερη βαρύτητα θα πρέπει να δοθεί σε

εκείνες που απειλούν τα προσωπικά δεδομένα των χρηστών, όπως είναι τα Hoaxes και τα Εργαλεία Διαχείρισης Συστημάτων από Απόσταση.

<1.2.1 Hoaxes>

Με τον όρο hoaxes περιγράφονται μηνύματα που έχουν ως αποκλειστικό σκοπό τη διάδοσή τους, την οποία επιτυγχάνουν μέσω του αναληθούς περιεχομένου τους που εκφοβίζει ή παραπλανεί τον παραλήπτη. Ο αναγνώστης του μηνύματος, παρασύρεται από τις αναφερόμενες ψευδείς προειδοποιήσεις, προσφορές, εκκλήσεις βοήθειας ή λοιπές ενημερώσεις που φέρει το μήνυμα και το προωθεί σε λοιπούς χρήστες.

Στην κατηγορία των hoaxes ανήκουν και τα Chain Letters τα οποία υπόσχονται στον αναγνώστη καλή τύχη αν τα προωθήσει σε όσο το δυνατόν περισσότερους χρήστες.

Χαρακτηριστική περίπτωση ψευδούς προειδοποιητικού μηνύματος είναι το "H.I.V. Needle" που εμφανίστηκε πριν αρκετά χρόνια στο διαδίκτυο και προειδοποιούσε ότι για την ύπαρξη μολυσμένων με τον ιό του AIDS βελόνων σε δημόσιους χώρους (π.χ. σε καθίσματα του μετρό ή θεάτρων).

> Hey friends!
>
> Please read this, it was sent to me today.
>
> "What lies behind us and what lies ahead of us are tiny matters compared to what lives within us" This happened in Paris. A few weeks ago, in a movie theatre, person sat on something poking that was on one of the seats. When she got up to see what it was, she found a needle sticking out of the seat with a note attached saying: Youhave just been infected by HIV". The Disease Control Centre (in Paris) reports many similar events in many other cities recently.
>
> All tested needles ARE HIV Positive. The Centre also reports that needles have been found in the cash dispensers at public Banking Machines We ask everyone to use extreme caution when faced with this kind of situation. All public chairs/seats should be inspected with vigilance and caution before use. A careful visual inspection should be enough. In addition,they ask that each of you pass this message along to all members of your family and your friends of the potential danger.
> .....

(Περισσότερο ζημιογόνα για τον χρήστη ήταν η περίπτωση μηνύματος που προειδοποιούσε για την ύπαρξη ιού με την ονομασία «JDBGMGR.EXE» που «υπήρχε σχεδόν σε όλους τους Ηλεκτρονικούς Υπολογιστές και έπρεπε οπωσδήποτε να διαγραφεί». Το εν λόγω πρόγραμμα ωστόσο ήταν αξιόπιστο και κάθε άλλο παρά τυχαία ήταν η εμφάνισή του σε κάθε υπολογιστή.)

ΠΡΟΣΟΧΗ!!!
Κυκλοφορεί ένας ιός που μεταδίδεται με το Messenger ή με την
ατζέντα των διευθύνσεων. Μπορεί να είναι σιωπηλός για 14 μέρες
μέχρι να καταστρέψει το σύστημα. Για να τον εξουδετερώσετε :
1. Πηγαίνετε στο Start και στην εντολή Find
2. Στο "Files or Folders" γράψτε το όνομα jdbgmgr.exe
3. Κάντε την αναζήτηση στο C: (Συνήθως αποθηκεύεται στο C:\WINNT\System32)
4. Πατήστε το find
5. Αν εμφανιστεί ο ιός (το εικονίδιό του είναι ένα αρκουδάκι), θα έχει το όνομα jdbgmgr.exe ΜΗΝ ΤΟ ΑΝΟΙΞΕΤΕ!!!
6. Επελέξτε το με το δεξί κλικ και κάντε delete
7. Πηγαίνετε σον κάδο απορριμμάτων και πετάξτε το οριστικά.
ΑΝ ΒΡΕΙΤΕ ΤΟΝ ΙΟ ΣΤΟ ΣΥΣΤΗΜΑ ΤΟΥ ΥΠΟΛΟΓΙΣΤΗ ΣΑΣ ΣΤΕΙΛΤΕ ΑΥΤΟ ΤΟ
ΜΗΝΥΜΑ ΣΕ ΟΛΕΣ ΤΙΣ ΔΙΕΥΘΥΝΣΕΙΣ ΤΗΣ ΑΝΤΖΕΝΤΑΣ ΣΑΣ ΓΙΑΤΙ ΕΤΣΙ
ΜΕΤΑΔΙΔΕΤΑΙ.

Παρότι τα **Hoaxes** στη πλειονότητά τους είναι απλά ενοχλητικά μηνύματα που ο χρήστης μαθαίνει σταδιακά να αναγνωρίζει, επιδρούν στο διαδικτυακό περιβάλλον αρνητικά με τρεις κυρίως τρόπους:

- Οι χρήστες μαθαίνουν να αγνοούν προειδοποιητικά μηνύματα, συμπεριλαμβανομένων εκείνων που έχουν αληθές περιεχόμενο.

- Οδηγούν σε σπατάλη των διαδικτυακών πόρων, καθώς χρησιμοποιείται ο χρόνος των χρηστών και διαθεσιμότητα του δικτύου για την διακίνησή τους. Αυτό κύρια αφορά επαγγελματικούς πόρους, όπου είναι σημαντικό το κόστος στην παραγωγικότητα των εργαζομένων και στην αποδοτικότητα του δικτύου. Πρόσθετα υπάρχει, δυνητικά έστω, ο κίνδυνος υπερφόρτωσης των συστημάτων διαχείρισης ηλεκτρονικού ταχυδρομείου οπότε η υψηλή διακίνηση hoaxes μπορεί να τα καταστήσει ανενεργά.

- Μπορεί να οδηγήσει σε αρνητική διαφήμιση, με τη διάδοση δυσφημιστικού υλικού εις βάρος ενός προσώπου ή οργανισμού.

- Επιτρέπει σε Spammers, δηλαδή σε χρήστες που αποστέλλουν διαφημιστικό υλικό μαζικά –χωρίς την έγκριση των παραληπτών , να συλλέγουν έγκυρες διευθύνσεις χρηστών

<1.2.2 Spamming>

Παρατηρείται συχνά η εκμετάλλευση e-mail accounts για την αποστολή διαφημιστικών μηνυμάτων, χωρίς τη συναίνεση του χρήστη.

Η ορθή επιχειρηματική πρακτική (και υπό περιπτώσεις των υφιστάμενο θεσμικό πλαίσιο) ορίζει αφενός να μη χρησιμοποιείται λογαριασμός ηλεκτρονικού ταχυδρομείου χωρίς τη πρότερη άμεση ή έμμεση συγκατάβαση του δικαιούχου του, αφετέρου να παρέχεται στον χρήστη η δυνατότητα εύκολης εξαίρεσής του από λίστες αποστολής διαφημιστικών μηνυμάτων.

Ωστόσο οι χρήστες συχνά εξαπατώνται παρέχοντας σε τρίτους την ηλεκτρονική τους διεύθυνση χωρίς να διασφαλίζουν την ορθή χρήση αυτής, είτε εξετάζοντας τη πολιτική προστασίας δεδομένων που ακολουθεί ο αντισυμβαλλόμενος είτε αξιοποιώντας τους όρους συνδιαλλαγής που σχετίζονται με τη χρήστη των δεδομένων τους.

Συνήθως η συγκέντρωση σχετικών στοιχείων πραγματοποιείται με τους ακόλουθους τρόπους:

- Με Ερωτηματολόγια on line ερευνών, όπου καλείται ο χρήστης να συμπληρώσει κατ' ελάχιστον δημογραφικά στοιχεία και στοιχεία επικοινωνίας.

- Μέσω ηλεκτρονικών καρτών εγγύησης, που συναντάται συνήθως στην εγκατάσταση προγραμμάτων μετά την οποία καλείται ο χρήστης να κάνει register την εγκατεστημένη εφαρμογή με αντάλλαγμα τη παροχή τακτικών ενημερώσεων, προνομιακών αναβαθμίσεων του προϊόντος και αποστολής διορθωτικών προγραμμάτων.

- Σε διαγωνισμούς (sweepstakes), για την συμμετοχή στους οποίους απαιτείται η αποστολή στοιχείων επικοινωνίας.

- Στη συμπλήρωση αιτήσεων μέλους σε δικτυακό κόμβο, όπου συνήθως για να αποκτήσει ο χρήστης πρόσβαση σε υπηρεσίες του κόμβου απαιτείται να συμπληρώσει αίτηση μέλους.

- Σε λοιπές δικτυακές υπηρεσίες όπου καλείται ο χρήστης να συμπληρώσει τη διεύθυνση του ηλεκτρονικού ταχυδρομείου του για να επωφεληθεί των διαθέσιμων υπηρεσιών. Χαρακτηριστικό παράδειγμα είναι η αποστολή ηλεκτρονικών καρτών μέσω δικτυακών τόπων.

<1.2.3 Δούρειοι Ίπποι (Trojan Horses)>

Οι επονομαζόμενοι Δούρειοι Ίπποι (ΔΙ), είναι προγράμματα που εξαπατούν τον χρήστη, ως προς το σκοπό και τη λειτουργία τους.

Οι πιο συνηθισμένες λειτουργίες των ΔΙ (Trojans) περιλαμβάνουν:

- Την πρόκληση ζημιών, όταν λειτουργούν ως ιοί ή (worms).
- Την ανάλυση του συστήματος του χρήστη, αναζητώντας ενδεχόμενες αδυναμίες του, όταν λειτουργούν ως scanning tools.
- Την απόκτηση του μεγαλύτερου δυνατού ελέγχου του συστήματος και την εκμετάλλευσή του όταν λειτουργούν ως back doors ή remote control tools.

Ο δεύτερος ισχυρότερος ιστορικά ιός (μετά τον Klez), ο CodeRed, αποτελεί παράδειγμα Δούρειοι Ίπποι, που μετά την εγκατάστασή του, επιχειρούσε να εντοπίσει λοιπά συνδεδεμένα συστήματα, τα οποία μόλυνε, ενώ παράλληλα εκτελούσε περιοδικά επιθέσεις τύπου Denial Of Service.

Δυο από τα ισχυρότερα παραδείγματα Δούρειων Ίππων που χρησιμοποιούνται ως εργαλεία remote control είναι το NetBus και το SubSeven. Και τα δυο μετά την client εγκατάστασή τους στο σύστημα ενός χρήστη επιτρέπουν, στον επιτιθέμενο να πραγματοποιεί εργασίες στο σύστημα του (άτυχου) χρήστη κατ' ελάχιστον για διασκέδαση ή ενόχληση.

Το SubSeven μάλιστα επιτρέπει στον επιτιθέμενο όχι μόνο να διασκεδάσει αλλά να εκτελέσει εργασίες όπως η αναζήτηση passwords.

<1.2.4 Back door and remote administration programs>

Πρόκειται για προγράμματα που επιτρέπουν τη διαχείριση του συστήματος ενός χρήστη από απόσταση εν αγνοία του πρώτου.

Με αυτό το τρόπο, καθώς ο υπολογιστής είναι συνδεδεμένος στο διαδίκτυο, επιτρέπει στον εισβολέα να ενεργοποιήσει τα εν λόγω προγράμματα

για να εκμεταλλευτεί το υπολογιστικό σύστημα του χρήστη π.χ. για να πραγματοποιήσει επίθεση σε άλλο συνδεδεμένο σύστημα ή για να εκτελέσει προγράμματα του συστήματος ή για να υποκλέψει δεδομένα.

Τα πιο γνωστά Back Door προγράμματα είναι τα BackOrifice, Netbus και SubSeven, εκ των οποίων τα δυο τελευταία λειτουργούν και ως Δούρειοι Ίπποι.

<1.2.5 Denial of service Attacks>

Οι επιθέσεις τύπου Denial of Service έχουν ως στόχο την διακοπή παροχής υπηρεσιών από έναν δικτυακό κόμβο ή πληροφοριακό σύστημα.

Τα προγράμματα που συνήθως χρησιμοποιούνται ακολουθούν τη τακτική μαζικής αποστολής μηνυμάτων – αιτημάτων στον στόχο ώστε να προκαλέσουν την αποτυχία ανταπόκρισής του και τη κατάρρευση του συστήματος.

Για την αποτελεσματικότητα των επιθέσεων ακολουθείται η τακτική χρήσης πολλών συστημάτων που πραγματοποιούν ταυτόχρονα επιθέσεις σε επιλεγμένους στόχους. Η επιλογή των συστημάτων αυτών συχνά αποτελούν συνδεδεμένους στο διαδίκτυο υπολογιστές χρηστών που αγνοούν ότι ο επιτιθέμενος εκμεταλλεύεται το σύστημά τους για τη διεξαγωγή επιθέσεων.

Ο τρόπος δε με τον οποίο ο επιτιθέμενος αποκτά τον έλεγχο δικτυωμένων συστημάτων απαιτεί την εγκατάσταση εφαρμογής Client που πραγματοποιεί επιθέσεις Distributed Denial Of Service (DDos) στο σύστημα- θύμα. Η εγκατάστασή της μπορεί να γίνει χρησιμοποιώντας δούρειο ίππος που θα ενεργοποιήσει απευθείας την εφαρμογή στο σύστημα.

<1.2.6 Cookies>

Τα cookies είναι συστημικά αρχεία που περιέχουν πληροφορίες σχετικές με τη πλοήγηση του χρήστη σε έναν δικτυακό τόπο.

Δημιουργούνται αυτόματα από το σύστημα και τα δεδομένα που φέρουν είναι στη πλειονότητά τους κωδικοποιημένα, ώστε να είναι αναγνώσιμα μόνο από τον δικτυακό κόμβο που τα δημιούργησε.

Τα δεδομένα που τηρούνται σε αυτά είναι συνήθως:

- ***Στοιχεία συμπεριφοράς του χρήστη:*** για παράδειγμα πότε επισκέφτηκε μια ιστοσελίδα, τι αναζήτηση έκανε σε αυτήν ή ποια διαφημιστικά banners επέλεξε. Δεδομένα συμπεριφοράς επιτρέπουν στον δικτυακό κόμβο να προσαρμόσει τις υπηρεσίες του, π.χ. την εμφάνιση της ιστοσελίδας ή τα προτεινόμενα προϊόντα, ώστε να είναι συμβατά με την πρότερη συμπεριφορά του χρήστη.
- ***Στοιχεία σύνδεσης του χρήστη*****:** όπως IP Address, λειτουργικό σύστημα, τύπο συστήματος πλοήγησης, service provider ή τον προηγούμενο δικτυακό κόμβο που είχε επισκεφτεί ο χρήστης. Τα δεδομένα που συλλέγονται αφορούν μόνο τον δεδομένο δικτυακό τόπου που διαχειρίζεται το cookie.

Τα cookies δημιουργούν προβληματισμούς σε θέματα ασφάλειας σε δυο επίπεδα: το πρώτο αφορά το είδος των δεδομένων προς αποθήκευση. Συνήθως ο χρήστης δεν είναι ενήμερος ως προς το είδος της πληροφορίας που εμπεριέχεται στα Cookies.

Το δεύτερο επίπεδο αφορά την προσπέλαση των διαθέσιμων cookies από κάθε δικτυακό κόμβο, ο οποίος παρότι δεν μπορεί να αξιοποιήσει το περιεχόμενο ενός cookie αν δεν έχει παραχθεί από αυτόν, μπορεί κατ' ελάχιστον να αξιοποιήσει τη πληροφορία ύπαρξής του από συγκεκριμένους δικτυακούς κόμβους. Για παράδειγμα οι Barnes&Noble μπορεί να δει ότι στο σύστημά μου έχω cookie της Amazon.Πιο αναλυτικά στα cookies θα αναφερθούμε σε παρακάτω κεφάλαια.

<1.2.7 Web Bugs>

Τα Web Bugs είναι γραφικά που ενσωματώνονται συνήθως σε μηνύματα ηλεκτρονικού ταχυδρομείου ή σε ιστοσελίδες και καταγράφουν στοιχεία που σχετίζονται με τον χρήστη.

Ενδεικτικά, ενσωματωμένο σε e-mail:

I. καταγράφει την IP διεύθυνση του χρήστη που διαβάζει το μήνυμα,
II. αν ο χρήστης διάβασε το μήνυμα και αν ναι πόσες φορές,
III. πόσες φορές το μήνυμα προωθήθηκε και διαβάστηκε

Αντίστοιχα, ενσωματωμένο σε ιστοσελίδα:

I. καταγράφει τον χρήστη (μέσω του IP Address),

II. τις ιστοσελίδες που επισκέπτεται και ευρύτερα τη συμπεριφορά του στο διαδίκτυο π.χ. ποια links επιλέγει,
III. τον χρόνο και τη διάρκεια της επίσκεψης ενός χρήστη σε μια ιστοσελίδα.

Μάλιστα web bug μπορεί να ενσωματωθεί και σε Word αρχεία, με ανάλογη χρήση.

Η συνδυασμένη χρήση Web Bugs και Cookies επιτρέπει σε δικτυακούς τόπους να προσδιορίζουν την ταυτότητα του χρήστη –μέσω της διεύθυνσής του- και τα ενδιαφέροντα ή τις συνήθειές του.

Παράδειγμα Web Bug

Παράδειγμα Web Bug βρίσκεται στην ιστοσελίδα www.investorplace.com.
Το γραφικό έχει τόσο μικρό μέγεθος που εντοπίζεται όχι οπτικά αλλά διαβάζοντας τον κώδικα της ιστοσελίδας.

```
...
<td>
        <!-- START MORE WAYS COLUMN 1 INCLUDE -->
        <A
HREF="http://ad.doubleclick.net/jump/investorplace.com/;sz=127x155;tile=3;ord=1052337845"
>
        <IMG
SRC="http://ad.doubleclick.net/ad/investorplace.com/;sz=127x155;tile=3;ord=105
2337845" border=0 height="155" width="127"></A>
        <!-- END ORE WAYS COLUMN 1 INCLUDE -->
</td> ...
```

Χαρακτηριστικό των Web Bugs είναι το ότι φέρουν Border=0.

<1.2.8 Ιοί (viruses) – worms (σκουλήκια)>

Οι ιοί είναι προγράμματα ή εντολές που προσαρτώνται σε προγράμματα ή δεδομένα και εκτελούνται παράλληλα με αυτά. Μπορούν να προκαλέσουν την αλλοίωση ή καταστροφή δεδομένων.

Τα Worms αντίστοιχα, είναι προγράμματα που μετά τη διείσδυσή τους σε ένα σύστημα, αντιγράφονται σε συνδεμένα συστήματα, επιχειρώντας τη διαρκή εξάπλωσή τους.

Και οι δυο κατηγορίες προγραμμάτων έχουν ως στόχο να πλήξουν το σύστημα στο οποίο εκτελούνται, προκαλώντας ζημιές όπως η διαγραφή δεδομένων, η αλλαγή της παραμετροποίησης του συστήματος, η μείωση της απόδοσής του και δη της δικτυακής επικοινωνίας (π.χ. μείωση της ταχύτητας μετάδοσης δεδομένων μεταξύ υπολογιστών).

Worm Ganga:
Εμφανίστηκε στις 17/3/2003 σε μηνύματα ηλεκτρονικού ταχυδρομείου με attachments που έχουν σύντομα ονόματα π.χ. OC.SCR. Όταν εκτελείται εγκαθίσταται στο σύστημα με το όνομα "Scandisk.exe" ή "Qakuesia.exe" και τροποποιεί όλα τα exe αρχεία του συστήματος. Η επίδρασή του περιορίζεται στη διακοπή λειτουργιών προϊόντων anti-virus που είναι ενεργοποιημένα στο σύστημα. Επίσης χρησιμοποιεί τις διευθύνσεις ηλεκτρονικού ταχυδρομείου που εντοπίζει στο Microsoft Outlook Address Book ή σε αποθηκευμένες ιστοσελίδες.

Ο ισχυρότερος ιός που ιστορικά έχει καταγραφεί είναι ο Klez ο οποίος παραμένει ως και τον Απρίλιο του 2003 πρώτος σε βαθμό εμφάνισης στο διαδικτύου σε σχέση με άλλους ιούς, όπως φαίνεται στον ακόλουθο πίνακα:

Κατάταξη	*Ιός*	*Ποσοστό*
1.	Worm/Klez.E (& G)	18.7%
2.	W32/Yaha.E	8.9%
3.	Worm/Yaha.M	7.5%
4.	Worm/Sobig.A	6.1%
5.	Worm/Lovegate.F	5.4%
6.	Worm/Sircam	5.0%
7.	W32/Funlove	3.4%
8.	W32/Nimda	2.8%
9.	W32/Elkern	2.7%
10.	Worm/BugBear	2.5%
11.	Worm/Ganda	1.8%
12.	Worm/Yaha.L	1.6%
	Others	33.6%

Πηγή: Απρίλιος, 30-04-2003, Central Command

<1.2.9 Καταχρηστικοί Όροι Ηλεκτρονικών Συναλλαγών>

Παράλληλα με το υφιστάμενο θεσμικό πλαίσιο, στην διαδικτυακή αγορά έχουν αναπτυχθεί βέλτιστες εμπορικές πρακτικές που έχουν ως στόχο να εξασφαλίσουν λειτουργικότητα και πίστη στις συναλλαγές μέσω διαδικτύου.

Η απουσία των πρακτικών αυτών είναι συχνά αρκετή για να χαρακτηρισθούν ως καταχρηστικοί οι όροι συναλλαγών. Για παράδειγμα η διαδικτυακή επιχειρηματική ηθική ορίζει ότι πρέπει να δηλώνεται ρητά στον χρήστη ο τρόπος και η έκταση χρήσης των προσωπικών του δεδομένων. Η απουσία αυτής της πρόβλεψης δεν εξασφαλίζει την ιδιωτικότητα των δεδομένων του πελάτη και μπορεί να οδηγήσει σε καταχρηστική χρήση αυτών.

Παράλληλα, παρατηρείται η ύπαρξη όρων που ενδέχεται να λειτουργήσουν εις βάρος των χρηστών και τίθενται ως υποχρεωτικοί από τους αντισυμβαλλόμενους – συχνά χωρίς την έγκαιρη ενημέρωση των χρηστών. Χαρακτηριστικός είναι ο όρος υποχρεωτικού registration (υποχρεωτική εγγραφή) του προϊόντος, που συχνά ανακαλύπτει ο χρήστης στη διαδικασία εγκατάστασης του λογισμικού, μετά την απόκτησή του. Παρότι ο όρος είναι συμβατός με τις συνθήκες εξουσιοδοτημένης χρήσης που επιχειρεί να εξασφαλίσει ο προμηθευτής λογισμικού, υποχρεώνει παράλληλα τον χρήστη στην παράδοση στοιχείων του ίδιου ή του συστήματός του π.χ. το e-mail του ή το είδος του λειτουργικού συστήματος που χρησιμοποιεί.

<1.2.10 Pop-Up Windows>

Η πλοήγηση στο διαδίκτυο συχνά οδηγεί στην εμφάνιση ιστοσελίδων που δεν έχουν επιλεγεί από τον χρήστη αλλά «προσφέρονται» αυτόματα από τους επισκεπτόμενους δικτυακούς τόπους.

Η πρακτική χρήσης Pop Up Παραθύρων πρωτίστως παραβιάζει την ελευθερία του χρήστη να επιλέγει εκείνος τον τρόπο πλοήγησής του στον παγκόσμιο ιστό και τα δεδομένα που τον ενδιαφέρουν.

Πρόσθετα μειώνει τη ποιότητα της σύνδεσης στο διαδίκτυο καθώς εκμεταλλεύεται–μικρή έστω- χωρητικότητα του δικτύου: αυτό γίνεται περισσότερο εμφανές στις περιπτώσεις εκείνες όπου, αφενός η σύνδεση είναι απλή, αφετέρου με το άνοιγμα μιας ιστοσελίδας ανοίγουν δεκάδες άλλα παράθυρα.

Παράλληλα, η χρησιμότητα των pop up παραθύρων είναι αμφισβητήσιμη: συνήθως φέρουν διαφημιστικά ή ενημερωτικά δεδομένα, με περιεχόμενο

ανεξάρτητο του επισκεπτόμενου δικτυακού τόπου από τον οποίο πραγματοποιήθηκε η ενεργοποίησή

Είναι δε ενοχλητικό κατά τη προσπέλαση μιας ιστοσελίδας να εμφανίζεται ένα pop – up παράθυρο που διακόπτει τον χρήστη από το διάβασμα των δεδομένων της ιστοσελίδας ή από τη συμπλήρωση φόρμας.

Ας σημειωθεί ότι η ίδια τεχνολογία των Pop Up παραθύρων δεν είναι απορριπτέα αλλά ο σκοπός και ο τρόπος χρήσης της. Η λειτουργική χρήση των pop ups είναι εμφανής σε δικτυακούς κόμβους όπου χρησιμοποιείται ως βοηθητικό εργαλείο που ενεργοποιείται με σκοπό αντίστοιχο της συμπεριφοράς του χρήστη.

Ενδεικτικά, επισκεπτόμενος πλήρες –όχι συνοπτικό - άρθρο της Le Monde, το οποίο υποδεικνύει το έντονο ενδιαφέρον του χρήστη στην ύλη της εφημερίδας, εμφανίζεται παράθυρο με προσφορά συνδρομής στην εφημερίδα που είναι συμβατό με τη διαδικτυακή συμπεριφορά του χρήστη και επομένως αποδεκτό.

Αντίστοιχα, κατά την έξοδο του χρήστη από την ιστοσελίδα της LeMonde ενεργοποιείται ευχαριστήριο pop up παράθυρο. Αντίστοιχα από στην έξοδο του χρήστη από την Allmedia (www.allmedia.com) ενεργοποιείται διαφημιστικό pop-up window και όχι κατά τη διάρκεια της πλοήγησής του στο Site.

<1.2.11 Scanning>

Η σύνδεση ενός συστήματος στο διαδίκτυο αφήνει συχνά περιθώρια σε άλλους χρήστες να το αναλύσουν με στόχο να εντοπίσουν τυχόν αδυναμίες τους, βάσει των οποίων μπορούν να αποκτήσουν αυξημένη πρόσβαση ή\ και έλεγχο σε αυτό.

Τα διαθέσιμα εργαλεία Scanning συνήθως αναζητούν τα ports του συστήματος που είναι διαθέσιμα, από τα οποία μπορούν να εξάγουν σημαντική πληροφορία, για παράδειγμα ποιο λειτουργικό χρησιμοποιεί και σε τι βαθμό είναι ενημερωμένο με τα τελευταία διαθέσιμα patches. Ενδεικτικά συστήματα που έχουν ανοικτά προς το διαδίκτυο το port 445 χρησιμοποιούν Windows 2000 ενώ όσα έχουν το port 139 πιθανότατα φέρουν Windows 9x ή NT.

Ένα ισχυρό εργαλείο Scanning είναι το Nessus, το οποίο όχι μόνο εντοπίζει τα ανοικτά ports αλλά και τα αδύναμα στοιχεία του συστήματος , όπως αν ένα πρόγραμμα χρειάζεται αναβάθμιση σε ασφαλέστερη έκδοση από αυτή στην οποία βρίσκεται.

<1.2.12 E-mail spoofing>

Το e-mail spoofing αποτελεί πρακτική παραποίησης ή απόκρυψης της πραγματικής πηγής από την οποία προήλθε το μήνυμα ηλεκτρονικού ταχυδρομείου. Χρησιμοποιείται συνήθως για να παραπλανήσει τον χρήστη ώστε να συλλεγούν από αυτόν χρήσιμα δεδομένα.

Ενδεικτικά αποστέλλονται μηνύματα με υποτιθέμενο αποστολέα τον διαχειριστή ενός συστήματος, ζητώντας από τον χρήστη να επιβεβαιώσει το Password που χρησιμοποιεί αλλιώς θα γίνει απενεργοποίηση του λογαριασμού του.

Ας σημειωθεί ότι η αποστολή Passwords μέσω ηλεκτρονικού ταχυδρομείου δεν αποτελεί ενδεδειγμένη διαδικτυακή πρακτική. Χρησιμοποιείται μόνο στις περιπτώσεις αποστολής προσωρινών passwords από τους δικτυακούς κόμβους στους νέους χρήστες που επιθυμούν να ενεργοποιήσουν τον λογαριασμό τους, και έχει ως στόχο να επιβεβαιώσει τα στοιχεία επικοινωνίας του νέου μέλους.

<1.2.13 Packet sniffing>

Ο όρος packet sniffing περιγράφει τεχνικές ελέγχου της κίνησης δεδομένων μεταξύ συστημάτων ενός δικτύου.

Χρησιμοποιούνται σε εφαρμογές που έχουν ως στόχο την ανάλυση της κινητικότητας ενός δικτύου (Intrusion Detection Systems-Συστήματα Ανίχνευσης Εισβολών) ή σε εφαρμογές υποκλοπής διακινούμενων δεδομένων.

Η χρήση σχετικών προγραμμάτων από μη εξουσιοδοτημένους χρήστες ενός δικτύου παραβιάζει την ιδιωτικότητα της επικοινωνίας μεταξύ χρηστών, ως προς την ίδια την ύπαρξη επικοινωνίας μεταξύ χρηστών και ως προς το περιεχόμενο αυτής - στο βαθμό που τα διακινούμενα δεδομένα δεν είναι κωδικοποιημένα (κρυπτογραφημένα).

<1.2.14 Cross-site scripting>

Η πλοήγηση στο διαδίκτυο βασίζεται στην αποστολή ερωτήσεων από τον χρήστη στον διαδικτυακό κόμβο, στην λήψη απαντήσεων από αυτόν και αντίστροφα.

Ενδέχεται οι λαμβανόμενες απαντήσεις να μη φέρουν μόνο τα δεδομένα που ζητήθηκαν αλλά εκτελέσιμο κώδικα που επηρεάζει τον χρήστη ή το σύστημά του εν αγνοία του.

Για παράδειγμα ενώ ο χρήστης επιλέγοντας μια διεύθυνση ζητά πληροφορίες για ένα προϊόν, επιστρέφονται πληροφορίες για το προϊόν αυτό οπτικά διαθέσιμες στον χρήστη –κείμενο, φωτογραφίες κ.α.-, καθώς επίσης άμεσα εκτελέσιμος κώδικας (script) για τον οποίο ο χρήστης δεν είναι γνώστης ούτε της ύπαρξης του ούτε της λειτουργίας του.

Αντίστοιχα τα Scripts χρησιμοποιούνται για να αλλοιώσουν τη δομή μια φόρμας σε ιστοσελίδα ώστε, επί παραδείγματι, να εισάγει ο χρήστης στοιχεία σε λάθος αναφερόμενα κελιά. Αλλοιώσεις πραγματοποιούνται και σε ηλεκτρονικές διευθύνσεις που είναι ενσωματωμένες στις ιστοσελίδες, ώστε αν επιλεγούν επιστρέφεται στον χρήστη άλλη σελίδα από την αναφερόμενη.

Παράλληλα μπορούν να χρησιμοποιηθούν για να μεταβάλλουν τη συμπεριφορά μιας ιστοσελίδας π.χ. ενώ ο χρήστης δίνει λέξεις κλειδιά σε μηχανή αναζήτησης τα επιστρεφόμενα αποτελέσματα φιλτράρονται με πρόσθετες λέξεις κλειδιά που ορίζονται στο Script.

Η πλέον διαδεδομένη χρήση των Scripts είναι σε Chat Clients και Πίνακες Ανακοινώσεων (:Bulletin Boards), όπου η καταχώρηση μηνύματος μπορεί να συνοδευτεί από επικίνδυνο κώδικα. Όταν ο χρήστης της συνομιλίας λαμβάνει από τον δικτυακό διακομιστή το μήνυμα, ταυτόχρονα δέχεται και τον κώδικα ο οποίος εκτελείται αυτόματα αν δεν έχουν ληφθεί μέτρα προστασίας από τον χρήστη.

1.3 Λόγοι Που Οδηγούν Στην Ανάγκη Πιστοποίησης Και Όχι Ανώνυμης Πρόσβασης Στις Διαδικτυακές Υπηρεσίες.

Το Web είναι ένα ανώνυμο μέσο. Οι διάφοροι διακομιστές μπορούν να μάθουν πολλά πράγματα για τους υπολογιστές που συνδέονται στο Net.Όπως ξέρουμε κάθε υπολογιστής έχει μια μοναδική IP διεύθυνση όπου μπορούμε από αυτή να δούμε διάφορα πράγματα για την ταυτότητα του χρήστη που είναι συνδεδεμένος (π.χ. την γεωγραφική του θέση).

Το Web είναι ένα σχετικά ανώνυμο μέσο, αλλά συνήθως είναι χρήσιμο για να ξέρετε ττοιος επισκέπτεται την τοποθεσία σας. Ευτυχώς για την μυστικότητα των επισκεπτών, μπορείτε να ανακαλύψετε πολύ λίγα για αυτούς χωρίς την βοήθεια τους.

Με λίγη δουλειά, οι διακομιστές μπορούν να μάθουν πολλά για τους υπολογιστές και τα δίκτυα που συνδέονται μαζί τους. 'Ενας Web browser συνήθως προσδιορίζει τον εαυτό του λέγοντας στον διακομιστή ποιο browser, έκδοση browser και λειτουργικό σύστημα έχετε. Μπορείτε να προσδιορίσετε ποια ανάλυση και βάθος χρωμάτων έχουν οι οθόνες των επισκεπτών σας και πόσο μεγάλα είναι τα παράθυρα των Web browser.

Κάθε υπολογιστής συνδεδεμένος στο Internet έχει μια μοναδική IP διεύθυνση. Αυτό την IP διεύθυνση ενός επισκέπτη μπορείτε να συμπεράνετε κάποια πράγμα. Μπορείτε να ανακαλύψετε ποιος κατέχει ένα IP και μερικές φορές, μπορείτε να μαντέψετε τη γεωγραφική θέση του επισκέπτη. Μερικές διευθύνσεις είναι πιο χρήσιμες από άλλες. Γενικά, τα άτομα με σταθερές Internet συνδέσεις, έχουν μια μόνιμη διεύθυνση. Οι πελάτες που καλούν μια εταιρεία παροχής υπηρεσιών Internet, συνήθως έχουν προσωρινές IP διευθύνσεις. Την επόμενη φορά που θα δείτε αυτή τη διεύθυνση, μπορεί να χρησιμοποιείται από διαφορετικό υπολογιστή και την επόμενη φορά που θα δείτε τον επισκέπτη, θα χρησιμοποιεί μάλλον μια διαφορετική διεύθυνση.

Ευτυχώς για τους Web χρήστες, καμία από τις πληροφορίες που δίνουν οι browser τους δεν τους προσδιορίζει. Αν θέλετε να μάθετε το όνομα ενός χρήστη ή άλλες πληροφορίες, θα πρέπει να τους ρωτήσετε.

Πολλές Web τοποθεσίες παρέχουν ελκυστικούς τρόπους να κάνουν τους χρήστες να παρέχουν τις πληροφορίες τους. Η εφημερίδα New York Times (http: *//mm.*nytimes.com), δίνει τα περιεχόμενα της δωρεάν, αλλά μόνο σε άτομα που θέλουν να παρέχουν πληροφορίες, όπως το όνομα, το φύλο και το συνολικό εισόδημα τους. Η τοποθεσία συζητήσεων και νέων Slashdot (http://www.slashdot.org), επιτρέπει σε εγγεγραμμένους χρήστες να συμμετέχουν σε συζητήσεις με ψευδώνυμα και να προσαρμόζουν το περιβάλλον που χρησιμοποιούν. Οι περισσότερες τοποθεσίες ηλεκτρονικού εμπορίου καταγράφουν τις πληροφορίες των πελατών τους όταν κάνουν την πρώτη τους

Αφού ζητήσετε και λάβετε πληροφορίες από τον επισκέπτη σας, χρειάζεστε έναν τρόπο να συσχετίσετε τις πληροφορίες του μαζί του, την επόμενη φορά που θα επισκεφθεί. Αν θέλετε να κάνετε την υπόθεση ότι μόνο ένα άτομο επισκέπτεται την τοποθεσία σας, από ένα συγκεκριμένο λογαριασμό ενός συγκεκριμένου υπολογιστή και ότι κάθε επισκέπτης χρησιμοποιεί μόνο ένα υπολογιστή, θα μπορούσατε να αποθηκεύσετε ένα cookie στον υπολογιστή του χρήστη για να προσδιορίζετε τον χρήστη. Αυτό φυσικά δεν ισχύει για όλους τους χρήστες - συνήθως, πολλά άτομα μοιράζονται τον ίδιο υπολογιστή και πολλά άτομα χρησιμοποιούν περισσότερους από έναν υπολογιστές. Τουλάχιστον κάποιες φορές, θα χρειαστεί να ρωτήσετε τον επισκέπτη ποιος είναι. Εκτός του ότι πρέπει να ρωτήσετε ποιος είναι, θα πρέπει επίσης να ζητήσετε από τον χρήστη να σας δώσει κάποια απόδειξη ότι είναι αυτός που λέει ότι είναι.

Αν ζητήσετε από ένα χρήστη να αποδείξει την ταυτότητα του, αυτό ονομάζεται **έλεγχος ταυτότητας.** Η συνηθισμένη μέθοδος ελέγχου ταυτότητας που χρησιμοποιείται σε Web τοποθεσίες, είναι να ζητάτε από τους επισκέπτες να εισάγουν ένα μοναδικό όνομα σύνδεσης και ένα κωδικό πρόσβασης. Ο έλεγχος ταυτότητας συνήθως χρησιμοποιείται για να επιτρέπει ή να μην επιτρέπει πρόσβαση σε συγκεκριμένες σελίδες ή πόρους, αλλά μπορεί να είναι προαιρετικός ή να χρησιμοποιείται για άλλους σκοπούς, όπως για λόγους προσαρμογής.

1.3.1 Ασφαλή Πρόσβαση Σε Διαδικτυακές Υπηρεσίες.

Ηλεκτρονικό Εμπόριο και Ασφάλεια.

Ο ρόλος της ασφάλειας στο Ηλεκτρονικό Εμπόριο είναι ζωτικής σημασίας. Πολλοί μπορούν να ενδιαφέρονται για διάφορες πληροφορίες. Οι απλοί χρήστες έχουν περιορισμένο χρόνο για να μάθουν και να δουλέψουν σε θέματα ασφάλειας στα διάφορα συστήματα. ανεξαρτήτου το πόσο ο κάθε χρήστης αξιολογεί το βαθμό που είναι σημαντικά τα δεδομένα που είναι αποθηκευμένα στο σκληρό τους δίσκο πρέπει όλοι οι χρήστες να παίρνουν διάφορα μέτρα ασφάλειας.

Μια επιχείρηση ηλεκτρονικού Εμπορίου για παράδειγμα αντιμετωπίζει προέρχεται από τους κακόβουλους χρήστες, εισβολείς (cracker).

Μερικά από τα μέτρα που μπορεί να εφαρμοστούν όπως χρησιμοποίηση του Ελέγχου Ταυτότητας και η Κρυπτογράφηση θα αναλυθούν παρακάτω.

Ασφαλές Ηλεκτρονικό Ταχυδρομείο.

Ο χρήστης ηλεκτρονικού ταχυδρομείου που έχει αποκτήσει προσωπικό ψηφιακό πιστοποιητικό από μια Αρχή Πιστοποίησης έχει τη δυνατότητα να ανταλλάσσει κρυπτογραφημένα μηνύματα, διαφυλάσσοντας έτσι την ασφάλεια των μηνυμάτων του και το απαραβίαστο της προσωπικής του ηλεκτρονικής αλληλογραφίας.

Ο χρήστης κρυπτογραφεί το μήνυμα του με το δημόσιο κλειδί του παραλήπτη και το υπογράφει με την ψηφιακή του υπογραφή. Έτσι, μόνο ο παραλήπτης μπορεί να αποκρυπτογραφήσει το μήνυμα, με το ιδιωτικό του κλειδί, και να διαβάσει το περιεχόμενο του μηνύματος. Ακόμη, ο παραλήπτης είναι σίγουρος ότι ο αποστολέας είναι αυτός που δηλώνει ότι απέστειλε το μήνυμα, βασιζόμενος στην ψηφιακή υπογραφή που φέρει το μήνυμα, καθώς επίσης και ότι το περιεχόμενο του μηνύματος δεν έχει αλλοιωθεί.

Πρόσβαση Σε Ασφαλείς Δικτυακούς Τόπους.

Η αποδοχή της Αρχής Πιστοποίησης συνεπάγεται την προσθήκη ψηφιακών πιστοποιητικών στον πλοηγητή (browser) του χρήστη του Διαδικτύου. Με βάση τα ιδιαίτερα χαρακτηριστικά του πιστοποιητικού αυτού, ο χρήστης έχει τη δυνατότητα να επισκεφτεί ασφαλείς δικτυακούς τόπους και να προσπελάσει δεδομένα, χωρίς αυτά να είναι δημοσιευμένα σε κοινή θέα.

Για παράδειγμα, ασφαλείς δικτυακοί τόποι είναι οι ιστοσελίδες http://mail.auth.gr και http://accounts.auth.gr για την διαχείριση του ηλεκτρονικού ταχυδρομείου και των λογαριασμών αντίστοιχα. Τα στοιχεία που υποβάλλει ο χρήστης και τα δεδομένα που βλέπει στους παραπάνω δικτυακούς τόπους δεν είναι διαθέσιμα σε κοινή θέα.

Προστασία Ευαίσθητων Δεδομένων Σε Γραμματείες Τμημάτων Και Διοικητικούς Φορείς.

Οι γραμματείες των τμημάτων ενός Ακαδημαϊκού Ιδρύματος καθώς επίσης και οι διοικητικές υπηρεσίες έχουν στη διάθεσή τους ιδιαίτερα ευαίσθητα δεδομένα που πρέπει να προστατευτούν.

Η βαθμολογία φοιτητών, τα οικονομικά στοιχεία των εργαζομένων, τα διοικητικά έγγραφα, οι πρυτανικές αποφάσεις, είναι μερικά σημαντικά δεδομένα που δεν πρέπει να είναι κοινώς προσπελάσιμα, παρά μόνο από εξουσιοδοτημένα μέλη και επίσης πρέπει να προστατεύονται από παραβιάσεις και αλλοιώσεις.

Η πιστοποίηση της ταυτότητας των χρηστών και η προστασία τέτοιου είδους δεδομένων μπορεί να επιτευχθεί με την Υποδομή Δημοσίου Κλειδιού. Με

τα ψηφιακά πιστοποιητικά για τους χρήστες επιβεβαιώνεται η ταυτότητά τους και με τους μηχανισμούς κρυπτογράφησης βεβαιώνεται η ασφάλεια των δεδομένων.

Προστασία Ερευνητικών Δεδομένων.

Η προστασία ερευνητικών αποτελεσμάτων και μελετών είναι ιδιαίτερα σημαντική σε ένα ακαδημαϊκό ίδρυμα. Τα ευαίσθητα ερευνητικά δεδομένα που αποθηκεύονται σε εξυπηρετητές πρέπει να προστατεύονται από μη εξουσιοδοτημένη πρόσβαση. Επίσης, η δικτυακή μεταφορά τους σε εξουσιοδοτημένα μέλη της ακαδημαϊκής κοινότητας πρέπει να είναι ασφαλείς.

Η Υποδομή Δημοσίου Κλειδιού παρέχει μηχανισμούς ασφαλείας για αποθήκευση και μεταφορά ερευνητικών δεδομένων. Τα ερευνητικά δεδομένα κρυπτογραφούνται, έτσι ώστε μόνο εξουσιοδοτημένα μέλη να έχουν τη δυνατότητα να τα αποκρυπτογραφήσουν και να τα αποκτήσουν.

Πρόσβαση Σε Ηλεκτρονικές Βιβλιοθήκες.

Η πρόσβαση σε ηλεκτρονικές βιβλιοθήκες είναι ένα αναγκαίο εργαλείο για την ακαδημαϊκή έρευνα και μελέτη.

Στην πλειοψηφία, οι ηλεκτρονικές βιβλιοθήκες παρέχουν τη δυνατότητα σύνδεσης χρηστών που έχουν διεύθυνση δικτύου (IP) με συγκεκριμένη μορφή (π.χ. οι χρήστες του Αριστοτελείου Πανεπιστημίου Θεσσαλονίκης μπορούν να προσπελάσουν τα ψηφιακά δεδομένα της βιβλιοθήκης του Α.Π.Θ μόνο αν έχουν διεύθυνση δικτύου της μορφής 155.207.x.y). Η λύση αυτή όχι μόνο δεν είναι ασφαλής, αλλά παρεμποδίζει και το έργο των ακαδημαϊκών μελών όταν αυτοί βρίσκονται εκτός του Ακαδημαϊκού Ιδρύματος ή συνδέονται μέσω κάποιου παροχέα δικτυακών υπηρεσιών (Internet Provider), οπότε και αποκτούν διεύθυνση δικτύου διαφορετικής μορφής.

Τα προβλήματα αυτά μπορούν να επιλυθούν με ένα πιο ευέλικτο σχήμα ταυτοποίησης των εξουσιοδοτημένων χρηστών. Η Υποδομή Δημοσίου Κλειδιού παρέχει ψηφιακά πιστοποιητικά για κάθε χρήστη, έτσι ώστε να επιβεβαιώνεται η ταυτότητά του και να έχει τη δυνατότητα πρόσβασης σε ηλεκτρονικές βιβλιοθήκες μόνο με βάση την ακαδημαϊκή του ιδιότητα.

Ασφαλή Και Αποτελεσματική Πλοήγηση Στο Διαδίκτυο.

Μέσα από την πιστοποίηση και όχι από την ανώνυμη πρόσβαση στο Internet ο χρήστης όπως και διαχειριστής ενός συστήματος μπορεί να έχει διάφορες επιπλέον πληροφορίες που στη συνέχεια να τον βοηθήσουν να λύσει διάφορα προβλήματα που πιθανόν να του παρουσιαστούν. Έτσι για παράδειγμα ένας διαχειριστής μπορεί να έχει την δυνατότητα παρακολούθησης της συμπεριφοράς ενός χρήστη μέσα στο Net.

Πλέγμα Δεδομένων (Data GRID).

Το Πλέγμα Δεδομένων είναι μια σχετικά νέα έννοια στην νέα ψηφιακή κοινωνία και αποδεικνύεται μια πολύ ουσιώδης δομή για τα Ακαδημαϊκά Ιδρύματα. Η δικτυακή αυτή δομή επιτρέπει σε ερευνητές, εργαστήρια και πανεπιστήμια από όλο τον κόσμο να συνενώνουν τις δυνάμεις τους για να έχουν μια δυναμική συνεργασία σε διάφορες ερευνητικές περιοχές.

Βασιζόμενοι σε μια κατανεμημένη δομή που περιλαμβάνει ηλεκτρονικές βιβλιοθήκες, δικτυακούς πόρους, χώρους αποθήκευσης ψηφιακών δεδομένων, υπολογιστικά συστήματα μεγάλης ισχύος ανά τον κόσμο, τα ακαδημαϊκά μέλη έχουν το δικαίωμα να χρησιμοποιήσουν τα μέσα αυτά, ανεξάρτητα από την φυσική τους τοποθεσία, με στόχο την έρευνα.

Για παράδειγμα χιλιάδες αστρονόμοι που ανήκουν σε διάφορα ακαδημαϊκά εργαστήρια του κόσμου και εστιάζουν σε μια ερευνητική περιοχή μπορούν να δημιουργήσουν ένα Πλέγμα Δεδομένων και να διαμοιράζονται όλα τα φυσικά μέσα που χρειάζονται για την έρευνα τους, ανεξάρτητα από την χωροταξική τους θέση.

Η πρόσβαση σε ερευνητικά δεδομένα, σε αποτελέσματα μελετών, σε δικτυακούς πόρους, σε χώρους αποθήκευσης δεδομένων και γενικότερα σε μέσα που χρησιμοποιούνται για έρευνα πρέπει να περιορίζεται μόνο σε εξουσιοδοτημένα μέλη της ακαδημαϊκής κοινότητας. Αυτό επιτυγχάνεται με την Υποδομή Δημοσίου Κλειδιού και με την αντιστοίχηση ψηφιακών πιστοποιητικών σε κάθε χρήστη, ώστε να επιβεβαιώνεται η ταυτότητάς τους.

Δημιουργία Ερευνητικών Ιστοσελίδων Με Δημόσια Και Ιδιωτικά Τμήματα.

Πολλά ερευνητικά προγράμματα που εκπονούνται στα πλαίσια ακαδημαϊκών προγραμμάτων έχουν οργανωμένες ιστοσελίδες, όπου και δημοσιεύονται διάφορα στοιχεία και αποτελέσματα για το ερευνητικό έργο που επιτελείται.

Στα ερευνητικά αυτά έργα είναι πιθανό να συμμετέχουν επιστημονικοί συνεργάτες από άλλα ακαδημαϊκά ιδρύματα και να κρίνεται αναγκαία η απομακρυσμένη προσπέλαση συγκεκριμένων συνεργατών στα ερευνητικά δεδομένα. Έτσι δημιουργείται η ανάγκη να υπάρχουν ιστοσελίδες που να παρέχουν πληροφορίες και να παρουσιάζουν το ερευνητικό έργο σε κάθε ενδιαφερόμενο, αλλά παράλληλα να υπάρχει η δυνατότητα απομακρυσμένης πρόσβασης από συγκεκριμένα ακαδημαϊκά μέλη σε δεδομένα της έρευνας που δεν είναι προς κοινή δημοσίευση.

Η διάκριση των εξουσιοδοτημένων ακαδημαϊκών μελών που μπορούν να έχουν πρόσβαση σε όλα τα ερευνητικά δεδομένα και στους υπόλοιπους ενδιαφερόμενους που έχουν περιορισμένη πρόσβαση, μπορεί να υλοποιηθεί με βάση την Υποδομή Δημοσίου Κλειδιού και την χρήση πιστοποιητικών. Ανάλογα με τα χαρακτηριστικά του πιστοποιητικού του χρήστη θα επιτρέπεται η αντίστοιχη προσπέλαση στην ερευνητική ιστοσελίδα.

Υποβολή Ψηφιακά Υπογεγραμμένων Εργασιών.

Σε μερικά μαθήματα δίνεται η δυνατότητα υλοποίησης ή παράδοσης εργασιών μέσα από το περιβάλλον μιας ιστοσελίδας.

Η Υποδομή Δημοσίου Κλειδιού παρέχει έναν ασφαλή τρόπο να καθοριστεί ο αποστολέας της εργασίας, ότι η εργασία δεν έχει αλλοιωθεί και έχει υποβληθεί στο χρονικό διάστημα της ανάθεσης, όπως αυτό έχει αρχικά οριστεί (χρονοσφράγιση-timestamp).

Υπογεγραμμένο Λογισμικό.

Η Υποδομή Δημοσίου Κλειδιού παρέχει ψηφιακά πιστοποιητικά σε χρήστες για να υπογράψουν το λογισμικό που αναπτύσσουν.

Οι ψηφιακές υπογραφές που συνοδεύουν το λογισμικό είναι τέτοιες ώστε οι αποδέκτες του λογισμικού να γνωρίζουν ποιος ανέπτυξε το λογισμικό καθώς επίσης και να είναι βέβαιοι ότι μπορούν να χρησιμοποιήσουν άμεσα το λογισμικό χωρίς να παρουσιαστούν προβλήματα ασφαλείας (εγκατάσταση ηλεκτρονικών ιών).

1.4 Πιστοποίηση (Authentication)

Πιστοποίηση είναι η επιβεβαίωση της ταυτότητας ενός ατόμου ή η επιβεβαίωση της πηγής αποστολής των πληροφοριών. Δηλαδή, το άτομο που επιθυμεί να επιβεβαιώσει την ταυτότητά ενός άλλου άτομου ή κάποιου εξυπηρετητή με το οποίο επικοινωνεί, βασίζεται στην πιστοποίηση.

Η πιστοποίηση μπορεί να υλοποιηθεί με τις παρακάτω μεθόδους:

1. **Κωδικούς Πρόσβασης π.χ. το PIN μιας τραπεζικής κάρτας ή το μυστικό κωδικό ενός λογαριασμού (password).**
2. **Κάτι που έχουμε στην ιδιοκτησία μας, π.χ. το κλειδί μιας πόρτας ή μια τραπεζική κάρτα.**
3. **Βιομετρικές μετρήσεις, π.χ. δακτυλικά αποτυπώματα, φωνή κτλ**
4. **Μέσω της τοποθεσίας**
5. **Μέσω της Κρυπτογράφησης**
6. **Ψηφιακές Υπογραφές**
7. **Ψηφιακά Πιστοποιητικά**

1.5 Αρχές Ελέγχου Ταυτότητας

Ο **έλεγχος ταυτότητας** προσπαθεί να αποδείξει ότι κάποιος είναι πραγματικά αυτός που λέει ότι είναι. Υπάρχουν πολλοί πιθανοί τρόποι να παρέχετε

έλεγχο ταυτότητας, αλλά με πολλά μέτρα ασφάλειας, οι πιο ασφαλείς μέθοδοι είναι αυτές που δημιουργούν τα περισσότερα προβλήματα.

Οι τεχνικές πιστοποίησης περιλαμβάνουν κωδικούς πρόσβασης, ψηφιακές υπογραφές, βιομετρικές μετρήσεις, όπως δακτυλικά αποτυπώματα και μέτρα που περιλαμβάνουν υλικό, όπως έξυπνες κάρτες. Μόνο δύο από αυτές είναι σε κοινή χρήση στο Web: οι κωδικοί πρόσβασης και οι ψηφιακές υπογραφές.

Οι βιομετρικές μετρήσεις και οι περισσότερες λύσεις υλικού περιλαμβάνουν ειδικές συσκευές εισόδου και θα πρέπει να περιορίζονται σε πιστοποιημένους χρήστες συγκεκριμένων υπολογιστών. Αυτό μπορεί να είναι αποδεκτό ή ακόμα και επιθυμητό για πρόσβαση στο εσωτερικό σύστημα μιας εταιρείας, αλλά απομακρύνει πολλά από τα πλεονεκτήματα ενός συστήματος διαθέσιμου μέσω του Web.

Οι κωδικοί πρόσβασης είναι απλοί να τους χειριστείτε, απλοί στην χρήση και δεν απαιτούν ειδικές συσκευές εισόδου. Παρέχουν κάποιο επίπεδο πιστοποίησης, αλλά μπορεί να μην είναι κατάλληλοι από μόνοι τους για συστήματα υψηλής ασφάλειας.

Ένας κωδικός πρόσβασης είναι μια απλή ιδέα. Εσείς και το σύστημα ξέρετε τον κωδικό πρόσβασης σας. Αν ένας επισκέπτης υποστηρίξει ότι είναι "εσείς" και ξέρει τον κωδικό πρόσβασης σας, το σύστημα έχει λόγο να πιστεύει ότι αυτός είναι "εσείς". Εφόσον κανείς άλλος δεν ξέρει ή δεν μπορεί να μαντέψει τον κωδικό πρόσβασης, αυτό είναι ασφαλές. Οι κωδικοί πρόσβασης, από μόνοι τους, έχουν διάφορες πιθανές αδυναμίες και δεν παρέχουν δυνατό έλεγχο ταυτότητας.

Πολλοί κωδικοί πρόσβασης μαντεύονται εύκολα. Εάν άφηναν τους χρήστες να επιλέξουν τους δικούς τους κωδικούς πρόσβασης, περίπου 50% θα επέλεγαν κωδικό πρόσβασης που μαντεύετε εύκολα. Οι συνηθισμένα κωδικοί πρόσβασης που ανήκουν σε αυτήν την περιγραφή, περιλαμβάνουν λέξεις από λεξικό ή το όνομα του χρήστη. Εις βάρος της χρησιμότητας, μπορείτε να εξαναγκάσετε τους χρήστες να συμπεριλαμβάνουν αριθμούς ή στίξη στους κωδικούς πρόσβασης, αλλά αυτό θα δυσκολέψει κάποιους χρήστες να θυμούνται τους κωδικούς τους. Μπορεί να βοηθήσει αν εκπαιδεύσετε τους χρήστες να επιλέγουν καλύτερους κωδικούς, αλλά ακόμα και όταν εκπαιδεύονται, περίπου το 25% των χρηστών θα εξακολουθήσει να επιλέγει ένα εύκολο κωδικό πρόσβασης. Θα πρέπει να εφαρμόσετε πολιτικές κωδικών πρόσβασης που θα εμποδίσουν τους χρήστες να επιλέγουν εύκολους συνδυασμούς, ελέγχοντας τους νέους κωδικούς πρόσβασης ως προς ένα λεξικό ή απαιτώντας κάποιους αριθμούς ή σύμβολα στίξης ή μια μίξη πεζών και κεφαλαίων. Ένα κίνδυνο που έχουν οι αυστηροί κανόνες, είναι ότι πολλοί έγκυροι χρήστες θα τους ξεχάσουν.

Οι κωδικοί πρόσβασης που δεν είναι εύκολο να τους θυμηθούμε, αυξάνουν την πιθανότητα ότι οι χρήστες θα κάνουν κάτι ανασφαλές, όπως να γράψουν τον κωδικό πρόσβασης σε μια σημείωση που θα κολλήσουν πάνω στην οθόνη τους.

Οι χρήστες πρέπει να εκπαιδευτούν να μην γράφουν τους κωδικούς πρόσβασης ή να μην κάνουν άλλα ανόητα πράγματα, όπως να δίνουν τους κωδικούς πρόσβασης μέσω τηλεφώνου.

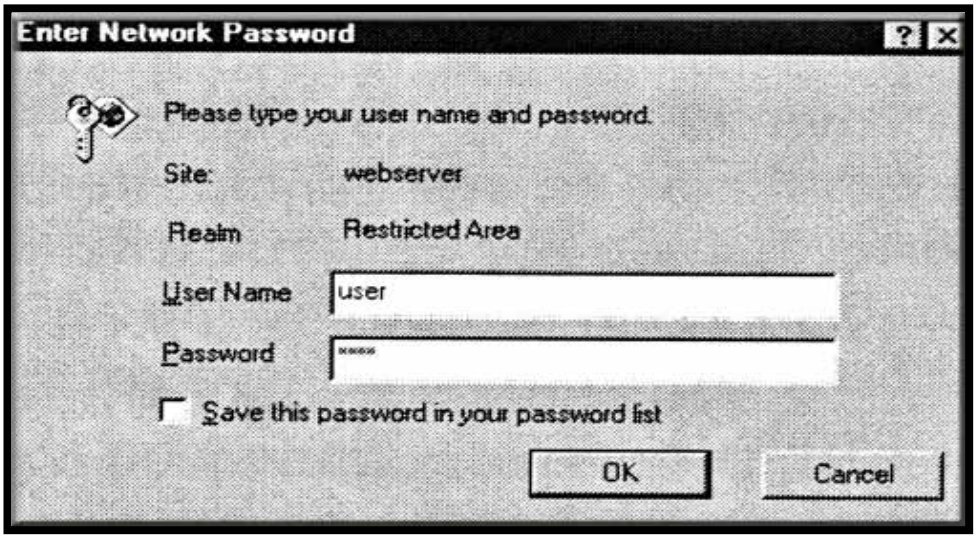
Οι κωδικοί πρόσβασης μπορεί επίσης να συλληφθούν ηλεκτρονικά. Τρέχοντας οι εισβολείς ένα πρόγραμμα που να συλλαμβάνει πληκτρολογήσεις σε ένα τερματικό ή χρησιμοποιώντας ένα πρόγραμμα υποκλοπής (sniffer) που να συλλαμβάνει την κίνηση του δικτύου, μπορούν και το κάνουν, να συλλαμβάνουν ζευγάρια από ονόματα σύνδεσης και κωδικούς πρόσβασης. Μπορείτε να περιορίσετε τις πιθανότατες να συλλαμβάνονται οι κωδικοί πρόσβασης κρυπτογραφώντας την κίνηση του δικτύου.

Παρόλα τα πιθανά κενά τους, οι κωδικοί πρόσβασης είναι ένας απλός και σχετικά αποτελεσματικός τρόπος πιστοποίησης των χρηστών σας. Παρέχουν ένα επίπεδο μυστικότητας, που μπορεί να μην είναι κατάλληλο για εθνική ασφάλεια, αλλά είναι ιδανικό για έλεγχο της κατάστασης της παράδοσης της παραγγελίας ενός πελάτη.

Χρησιμοποιώντας τον Έλεγχο Ταυτότητας

Οι μηχανισμοί ελέγχου ταυτότητας είναι ενσωματωμένοι στους πιο δημοφιλείς Web Server και Web διακομιστές. Οι Web διακομιστές μπορούν να απαιτούν όνομα χρήστη και κωδικό πρόσβασης για άτομα που ζητούν αρχεία από συγκεκριμένους καταλόγους του διακομιστή.

Όταν σας ζητείται ένα όνομα σύνδεσης και κωδικός πρόσβασης, ο browser θα παρουσιάσει ένα παράθυρο διαλόγου που θα μοιάζει με την Εικόνα 1.



Εικόνα 1

Οι Web browser κάνουν στους χρήστες έλεγχο ταυτότητας όταν προσπαθούν να επισκεφτούν ένα απαγορευμένο κατάλογο, σε ένα Web διακομιστή.

Ο Web διακομιστής Apache και ο IIS της Microsoft, σας επιτρέπουν πολύ εύκολα να προστατεύσετε όλο ή μέρος μιας τοποθεσίας με αυτόν τον τρόπο. Χρησιμοποιώντας PHP ή MySql, υπάρχουν και πολλοί άλλοι τρόποι που μπορούμε να επιτύχουμε το ίδιο αποτέλεσμα. Η χρήση της MySql. είναι γρηγορότερη από τον ενσωματωμένο έλεγχο ταυτότητας. Με την PHP, μπορείτε να παρέχετε πιο ευέλικτο έλεγχο ταυτότητας ή να παρουσιάσετε την αίτηση με ένα πιο ωραίο τρόπο.

1.6 Τα Βασικά Της Κρυπτογράφησης.

Ένας αλγόριθμος κρυπτογράφησης είναι μια μαθηματική διαδικασία, που μετασχηματίζει τις πληροφορίες σε μια μάλλον τυχαία συμβολοσειρά δεδομένων.

Τα δεδομένα με τα οποία ξεκινάτε συνήθως ονομάζονται απλό κείμενο, αν και δεν είναι σημαντικό για την διαδικασία τι πληροφορίες είναι αυτές - κείμενο ή κάποιου άλλου είδους δεδομένα. Παρόμοια, οι κρυπτογραφημένες πληροφορίες ονομάζονται κρυπτογραφημένο κείμενο, αλλά σπάνια δείχνουν σαν κείμενο. Η Εικόνα 2 δείχνει τη διαδικασία κρυπτογράφησης ως ένα απλό διάγραμμα ροής. Απλό κείμενο μπαίνει σε μια μηχανή κρυπτογράφησης, που μπορεί να είναι μια μηχανική συσκευή, όπως η μηχανή Engima στο Δεύτερο Παγκόσμιο Πόλεμο, αλλά τώρα είναι σχεδόν πάντα ένα πρόγραμμα υπολογιστή. Η μηχανή παράγει το κρυπτογραφημένο κείμενο.

Εικόνα 2

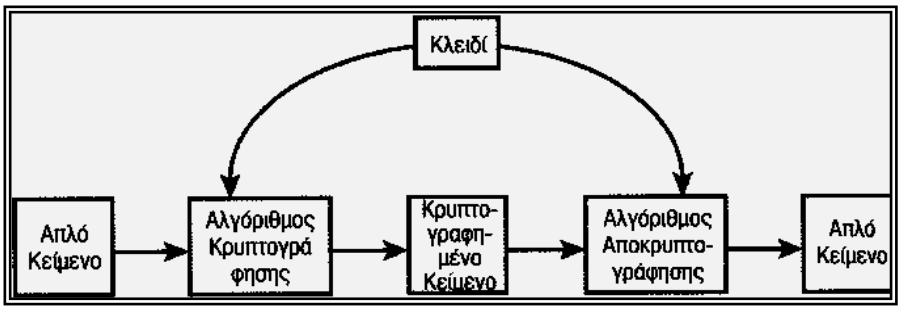
Η κρυπτογράφηση παίρνει απλό κείμενο και το μετασχηματίζει σε ένα τυχαίο κρυπτογραφημένο κείμενο.

Για να δημιουργήσουμε τον προστατευμένο κατάλογο, του οποίου η προτροπή ταυτότητας φαίνεται στην Εικόνα 1, χρησιμοποιήσαμε τον πιο βασικό τύπο πιστοποίησης του Apache. Αυτό κρυπτογραφεί τον κωδικό πρόσβασης πριν τον αποθηκεύσει. Δημιουργήσαμε χρήστη με κωδικό πρόσβασης password. Κρυπτογραφήθηκε και αποθηκεύθηκε ως aEDuA3X3H.mc2. Μπορείτε να δείτε ότι το απλό κείμενο και το κρυπτογραφημένο δεν έχουν καμία σχέση μεταξύ τους.

Αυτή η συγκεκριμένη μέθοδος κρυπτογράφησης δεν είναι αναστρέψιμη. Πολλοί πρόσβασης αποθηκεύονται χρησιμοποιώντας ένα αλγόριθμο κρυπτογράφησης μιας κατεύθυνσης. Για να δούμε αν είναι σωστός ένας κωδικός πρόσβασης, δεν χρειάζεται να αποκρυπτογραφήσουμε τον αποθηκευμένο κωδικό πρόσβασης. Μπορούμε αν κρυπτογραφήσουμε τον εισαχθέντα κωδικό πρόσβασης και να τον συγκρίνουμε με τον αποθηκευμένο.

Πολλές, αλλά όχι όλες οι διαδικασίες, μπορούν να αναστραφούν. Η διαδικασία αναστροφής ονομάζεται αποκρυπτογράφηση. Η Εικόνα 3 δείχνει μια αμφίδρομη διαδικασία κρυπτογράφησης

.



Εικόνα 3

Η κρυπτογράφηση παίρνει απλό κείμενο και το μετασχηματίζει σε τυχαία κρυπτογράφημένο κείμενο. Η αποκρυπτογράφηση παίρνει το κρυπτογραφημένο κείμενο και το μετατρέπει σε απλό κείμενο.

Η κρυπτογραφία έχει ιστορία σχεδόν 4000 χρόνων, αλλά εξελίχθηκε τον Δεύτερο Παγκόσμιο Πόλεμο. Η ανάπτυξη της από τότε, έχει ακολουθήσει ένα παρόμοιο μοτίβο με την υιοθέτηση των δικτύων υπολογιστών, που αρχικά χρησιμοποιήθηκαν μόνο από στρατιωτικές και οικονομικές υπηρεσίες, χρησιμοποιήθηκαν πιο ευρέως από εταιρείες στην δεκαετία του '70 και έγιναν παγκόσμια αποδεκτά την δεκαετία του 90. Τα τελευταία χρόνια η κρυπτογράφηση έχει προχωρήσει πολύ από αυτό που ήξεραν οι απλοί άνθρωποι από ταινίες του Δεύτερου Παγκοσμίου Πολέμου, σε κάτι για το οποίο διαβάζουμε στις εφημερίδες και χρησιμοποιούμε κάθε φορά που αγοράζουμε κάτι με τους Web Browser μας.

Υπάρχουν πολλοί διαφορετικοί αλγόριθμοι κρυπτογράφησης. Μερικοί, όπως ο DE5, χρησιμοποιούν ένα μυστικό ή ιδιωτικό κλειδί και μερικοί άλλοι, όπως ο RSA, χρησιμοποιεί ένα δημόσιο κλειδί και ένα διαφορετικό ιδιωτικό κλειδί.

Οι Αλγόριθμοι Κρυπτογράφησης.

Οι αλγόριθμοι κρυπτογράφησης χρησιμοποιούν:

Α) Μυστικό ή Ιδιωτικό Κλειδί π.χ. DES

Β) Δημόσιο και ένα διαφορετικό Ιδιωτικό Κλειδί π.χ. RSA

Θα αναφερθούμε στη συνέχεια σε μερικά θέματα σχετικά με παραπάνω**.**

Α) Κρυπτογράφηση Ιδιωτικού Κλειδιού.

Η κρυπτογράφηση Ιδιωτικού Κλειδιού βασίζεται σε πιστοποιημένα άτομα , που ξέρουν ή έχουν πρόσβαση σε ένα κλειδί. Αυτό το κλειδί πρέπει να διατηρείται Μυστικό.Ο αποστολέας με τον παραλήπτη έχουν το ίδιο κλειδί.

Ο πιο ευρέως χρησιμοποιούμενος αλγόριθμος μυστικού κλειδιού είναι ο Data Encryption Standard (DES). Αυτός ο συνδυασμός αναπτύχθηκε από την IBM στην δεκαετία του 70 και υιοθετήθηκε ως η αμερικανική τυποποίηση για εμπορική και εμπιστευτική επικοινωνία της κυβέρνησης. Οι ταχύτητες των υπολογιστών είναι πολύ μεγαλύτερες τώρα απ' ό,τι το 1970 και το DES έχει γίνει απαρχαιωμένο, τουλάχιστον από το 1998.

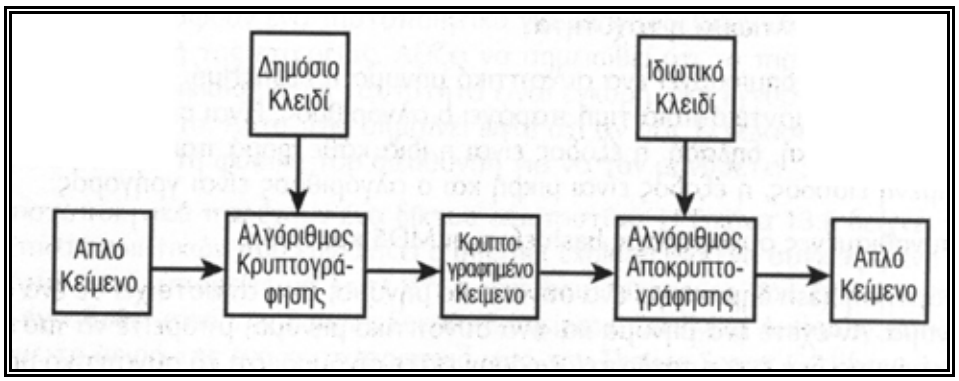
Άλλα πολύ γνωστά συστήματα μυστικών κλειδιών περιλαμβάνουν τα RC2,RC4,RC5,triple DES και IDEA Το Triple DES είναι αρκετά ασφαλές (για κάποιον περίεργο λόγο, το Triple DES είναι δύο φορές πιο ασφαλές από το DES. Αν θέλετε κάτι τρεις φορές πιο δυνατό, Θα μπορούσατε να γράψετε ένα πρόγραμμα που να χρησιμοποιεί ένα τετραπλό αλγόριθμο DES). Χρησιμοποιεί τον ίδιο αλγόριθμο με το DES, εφαρμόζεται τρεις φορές, με τρία διαφορετικά κλειδιά. Ένα απλό μήνυμα κειμένου κρυπτογραφείται με το κλειδί 1, αποκρυπτογραφείται με το κλειδί 2 και μετά κρυπτογραφείται με το κλειδί 3.

Ένα προφανές κενό κρυπτογράφησης είναι ότι, για να μπορείτε να στείλετε σε κάποιον ένα ασφαλές μήνυμα, χρειάζεστε έναν ασφαλή τρόπο να στείλετε το μυστικό κλειδί σε αυτόν. Αν έχετε ένα ασφαλή τρόπο να στείλετε ένα κλειδί, γιατί να μην του δώσετε και το μήνυμα με αυτόν τον τρόπο;

Ευτυχώς, υπήρξε κάτι καινούργιο το 1976, όταν οι Diffie και Hellman δημοσίευσαν το πρώτο συνδυασμό δημόσιου κλειδιού.

Β) Κρυπτογράφηση Δημόσιου Κλειδιού.

Η κρυπτογράφηση δημόσιου κλειδιού βασίζεται σε δύο διαφορετικά κλειδιά, ένα δημόσιο και ένα ιδιωτικό. Όπως φαίνεται στην Εικόνα 4, το δημόσιο κλειδί χρησιμοποιείται για να κρυπτογραφεί μηνύματα και το ιδιωτικό για να τα αποκρυπτογραφεί.



Εικόνα 4

Η κρυπτογράφηση του δημόσιου κλειδιού χρησιμοποιεί ξεχωριστά κλειδιά για κρυπτογράφηση και αποκρυπτογράφηση.

Το πλεονέκτημα αυτού του συστήματος είναι ότι το δημόσιο κλειδί, όπως αναφέρει και το όνομα του, μπορεί να διανεμηθεί δημόσια. Σε όποιον δίνετε το δημόσιο κλειδί, μπορεί να σας στείλει ένα ασφαλές μήνυμα. Εφόσον έχετε μόνο το ιδιωτικό σας κλειδί, τότε μόνο μπορείτε να αποκρυπτογραφήσετε το μήνυμα.

Ο πιο συνηθισμένος αλγόριθμος δημόσιου κλειδιού είναι ο RSA.

Είναι τεράστιο πλεονέκτημα το να υπάρχει η δυνατότητα να μεταδίδεται ένα δημόσιο κλειδί χωρίς να ανησυχείτε αν το δει κάποιος τρίτος, αλλά τα συστήματα με μυστικά κλειδιά εξακολουθούν να χρησιμοποιούνται ευρέως.

1.7 Ψηφιακές Υπογραφές

Οι ψηφιακές υπογραφές σχετίζονται με την κρυπτογράφηση των δημόσιων κλαδιών, αλλά αντιστρέφουν το ρόλο των δημόσιων και των ιδιωτικών κλειδιών. Ένας αποστολέας μπορεί να κρυπτογραφήσει και να υπογράψει ψηφιακά ένα μήνυμα με το μυστικό κλειδί. Όταν το μήνυμα ληφθεί, ο παραλήπτης μπορεί να το αποκρυπτογραφήσει με το δημόσιο κλειδί του αποστολέα. Καθώς ο αποστολέας είναι το μόνο άτομο που έχει πρόσβαση στο μυστικό κλειδί, ο παραλήπτης μπορεί να είναι σίγουρος από ποιον ήρθε το μήνυμα και ότι δεν έχει αλλάξει.

Οι ψηφιακές υπογραφές μπορεί να είναι πραγματικά χρήσιμες. Επιτρέπουν στον αποστολέα να είναι σίγουρος ότι το μήνυμα δεν έχει υποκλαπεί και κάνουν δύσκολο για τον αποστολέα να αρνηθεί ότι έστειλε το μήνυμα.

Είναι σημαντικό να ξέρετε ωστόσο, ότι αν και τα μηνύματα είναι κρυπτογραφημένα, μπορούν να διαβαστούν από οποιονδήποτε έχει το δημόσιο κλειδί. Αν και χρησιμοποιούνται ίδιες τεχνικές και κλειδιά, ο σκοπός της κρυπτογράφησης εδώ είναι να εμποδίσει την υποκλοπή και την άρνηση συμμετοχής, και όχι να εμποδίσει το διάβασμα.

Καθώς η κρυπτογράφηση του δημόσιου κλειδιού είναι σχετικά αργή για τα μεγάλα μηνύματα, συνήθως χρησιμοποιείται ένας άλλος τύπος αλγορίθμου, που ονομάζεται συνάρτηση Hash για να βελτιωθεί η ταχύτητα.

Οι πιο συνηθισμένες συναρτήσεις Hash είναι οι MD5 και SHA.

Μια συνάρτηση Hash δημιουργεί ένα συνοπτικό μήνυμα, που αντιστοιχεί σε ένα συγκεκριμένο μήνυμα. Αν έχετε ένα μήνυμα και ένα συνοπτικό μήνυμα, μπορείτε να πιστοποιήσετε ότι το μήνυμα δεν έχει υποκλαπεί, εφόσον είστε σίγουροι ότι το συνοπτικό μήνυμα δεν έχει υποκλαπεί.

Με αυτόν τον τρόπο, ο συνηθισμένος τρόπος δημιουργίας μιας ψηφιακής υπογραφής είναι να δημιουργείτε ένα συνοπτικό μήνυμα για ολόκληρο το μήνυμα, χρησιμοποιώντας μια γρήγορη συνάρτηση Hash και μετά να κρυπτογραφήσετε μόνο το σύντομο συνοπτικό μήνυμα χρησιμοποιώντας ένα αργό αλγόριθμο κρυπτογράφησης δημόσιου κλειδιού. Η υπογραφή μπορεί τώρα να σταλεί με το μήνυμα μέσω οποιασδήποτε κανονικής, ανασφαλούς μεθόδου.

1.8 Ψηφιακά Πιστοποιητικά

Είναι καλό να μπορείτε να πιστοποιήστε ότι ένα μήνυμα δεν έχει αλλάξει και ότι μια σειρά από μηνύματα προέρχονται από ένα συγκεκριμένο χρήστη ή υπολογιστή. Για εμπορικές συναλλαγές, θα ήταν ακόμα καλύτερο να μπορείτε να συνδέσετε αυτόν τον χρήστη ή διακομιστή με μια πραγματική νομική οντότητα, όπως ένα άτομο ή εταιρεία.

Ένα ψηφιακό πιστοποιητικό συνδυάζει ένα δημόσιο κλειδί και τις πληροφορίες ενός ατόμου ή οργανισμού ή εταιρείας, με μια υπογεγραμμένη ψηφιακή μορφή. Έχοντας ένα πιστοποιητικό και το δημόσιο κλειδί κάποιου άλλου, μπορείτε να στείλετε ένα κρυπτογραφημένο μήνυμα και έχετε και τις πληροφορίες αυτού του ατόμου, που ξέρετε ότι δεν έχουν αλλάξει.

Το πρόβλημα εδώ είναι ότι οι πληροφορίες είναι αξιόπιστες όσο το άτομο που τις υπέγραψε. Οποιοσδήποτε μπορεί να δημιουργήσει και να υπογράψει ένα πιστοποιητικό, λέγοντας ότι είναι κάποιος. Στις εμπορικές συναλλαγές, θα ήταν χρήσιμο να έχετε μια αξιόπιστη τρίτη εταιρεία που να πιστοποιεί την ταυτότητα των συμμετεχόντων και τις πληροφορίες που καταγράφονται στα πιστοποιητικά τους.

Αυτές οι τρίτες εταιρείες ονομάζονται Certifying Authorities (CA Αρχές πιστοποίησης). Αυτές δίνουν ψηφιακά πιστοποιητικά σε άτομα και εταιρείες, για ελέγχους ταυτότητας. Οι δύο πιο γνωστές είναι οι Verisign και Thawte , αλλά υπάρχουν και πολλές άλλες. Η VeriSign και η Thawte ανήκουν στην ίδια εταιρεία και υπάρχει λίγη πρακτική διαφορά μεταξύ τους.

Οι αρχές υπογράφουν ένα πιστοποιητικό για να αποδείξουν ότι έχουν δει την ταυτότητα του ατόμου ή της εταιρείας. Αξίζει να σημειωθεί ότι το πιστοποιητικό δεν είναι μια αναφορά ή μια δήλωση ότι η ταυτότητα είναι έγκυρη. Δεν εγγυάται ότι συνδιαλέγεστε με κάποιον αξιόπιστο. Αυτό που σημαίνει είναι ότι αν σας ξεγελάσει, έχετε μεγάλη πιθανότητα να μάθετε τη φυσική του διεύθυνση για να τον μηνύσετε.

Τα πιστοποιητικά παρέχουν ένα δίκτυο αξιοπιστίας. Η Εικόνα 5 δείχνει την διαδρομή των πιστοποιητικών που εμφανίζει ο Internet Explorer για ένα συγκεκριμένο πιστοποιητικό. Από αυτό, μπορείτε να δείτε ότι η www.equifaxsecure.com έχει ένα πιστοποιητικό που έχει δοθεί από την Equifax Secure E-Business Certifying Authority. Αυτή η εταιρεία, έχει με τη σειρά της ένα πιστοποιητικό από την Thawte Server Certifying Authority.

Εικόνα 5

Η διαδρομή πιστοποιητικού για το www.equifaxsecure.com το οποίο δείχνει το δίκτυο αξιοπιστίας που μας επιτρέπει να εμπιστευτούμε αυτή την τοποθεσία.

Η πιο συνηθισμένη χρήση των ψηφιακών πιστοποιητικών είναι το να παρέχουν ένα είδος αξιοπιστίας σε μια τοποθεσία ηλεκτρονικού εμπορίου. Με ένα πιστοποιητικό που δίνεται από μια γνωστή αρχή, οι Web Browser μπορούν να κάνουν SSL συνδέσεις στην τοποθεσία σας, χωρίς να εμφανίζουν παράθυρα διαλόγου προειδοποίησης. Οι WEB διακομιστές που επιτρέπουν SSL συνδέσεις, ονομάζονται συνήθως ασφαλείς Web διακομιστές.

1.9 Μέθοδοι πρόσβασης –Πιστοποίησης Χρηστών

1.9.1 Μέθοδοι Πρόσβασης.

Η απόδειξη της ταυτότητας κάθε χρήστη ονομάζεται έλεγχος ταυτότητας.Μια από τις πιο συνηθισμένες μεθόδους ελέγχου ταυτότητας που χρησιμοποιείται στο Web είναι η εισαγωγή ενός μοναδικού ονόματος σύνδεσης και ένας κωδικός πρόσβασης.

Με αυτόν τον τρόπο επιτρέπεται ή όχι η πρόσβαση σε συγκεκριμένες σελίδες ή πόρους.

Ένας δικτυακός τόπος κάνει έλεγχο των χρηστών με στόχο την πρόσβαση αυτών σε αυτόν με τον τρόπο που φαίνεται παρακάτω.

Υλοποίηση του Ελέγχου Πρόσβασης.

Ο απλός έλεγχος πρόσβασης δεν είναι δύσκολος. Ο κώδικας που φαίνεται τη Λίστα 14.1, παρέχει μια από τρεις πιθανές εξόδους. Αν το αρχείο φορτώνεται χωρίς παραμέτρους, θα εμφανίσει μια HTML φόρμα που ζητά ένα όνομα χρήστη και κωδικό πρόσβασης. Αυτού του είδους η φόρμα φαίνεται στην Εικόνα 6.

Εικόνα 6

Η HTML φόρμα μας, ζητά από τον επισκέπτη να εισάγει ένα όνομα χρήστη και κωδικό πρόσβασης προκειμένου να αποκτήσει πρόσβαση.

Αν οι παράμετροι δοθούν αλλά δεν είναι σωστές, θα εμφανιστεί ένα μήνυμα λάθους.Το μήνυμα λάθους το δικό μας φαίνεται στην Εικόνα 7.

Εικόνα 7

Όταν οι χρήστες εισάγουν λανθασμένες πληροφορίες, πρέπει να τους δείξουμε ένα μήνυμα λάθους. Σε μια πραγματική τοποθεσία, μπορείτε να εμφανίσετε κάποιο πιο φιλικό μήνυμα.

Αν δοθούν αυτές οι παράμετροι και είναι σωστές, θα εμφανιστούν τα μυστικά περιεχόμενα. Ο έλεγχος περιεχομένων μας φαίνεται στην Εικόνα 8.

Εικόνα 8

Όταν παρέχονται σωστές πληροφορίες, το script μας εμφανίζει τα περιεχόμενα.

Ο κώδικας για να δημιουργηθεί η λειτουργικότητα των Εικόνων 6, 7 και 8, φαίνεται στο Κώδικα 1.

Κώδικας 1
secret.php

PHP και HTML για Παροχή ενός Απλού Μηχανισμού Ελέγχου Ταυτότητας

```
<?
If ( ! isset ($ name)&& !isset($password))
{
// ο επισκέπτης πρέπει να δώσει όνομα και κωδικό πρόσβασης
?>
<h1>Please Log In/h1>
This page is secret.
<form method = post action = "secret.php">
<table border = 1>
<tr>
     <th> Username </th>
     <td> <input type = text name = name> </td>
</tr>
<tr>
   <th> Password </th>
```

```
   <td> <input type = password name = password> </td>
</tr>
 <tr>
   <td colspan =2 align = center>
   <input type = submit value = "Log In">
 </td>
 </tr>
 </table>
 </form>
else if ($name=="user"&&$password=="pass" )
{
// ο συνδυασμός ονόματος και κωδικού πρόσβασης είναι σωστός
echo "<h1>Here it is!</h1>";
echo "I bet you are glad you can see this secret page.";
}

else

{
// ο συνδυασμός ονόματος και κωδικού πρόσβασης είναι λάθος
echo "<h1>Go Away!</h1>" ;
echo "You are not authorized to view this resource.";
}
?>
```

Ο κώδικας 1 θα δώσει έναν απλό μηχανισμό ελέγχου ταυτότητας, που επιτρέπει στους πιστοποιημένους χρήστες να δουν μια σελίδα, αλλά έχει κάποια σημαντικά προβλήματα.

Αυτό το script.

- Έχει ένα όνομα χρήστη και κωδικό πρόσβασης που είναι γραμμένα μέσα στο script
- Αποθηκεύει το κωδικό πρόσβασης σαν απλό κείμενο
- Προστατεύει μόνο μια σελίδα
- Μεταδίδει το κωδικό πρόσβασης ως απλό κείμενο

Αυτά τα θέματα μπορούν να αντιμετωπιστούν με διάφορους τρόπους και με επιτυχία.

Στο παράδειγμα που είδαμε παραπάνω ο κωδικός πρόσβασης αποθηκεύεται στο ίδιο το αρχείο. Ακόμη αποθηκεύεται σαν απλό κείμενο. Στα επόμενα κεφάλαια θα δούμε και άλλους τρόπους ελέγχου ταυτότητας χρησιμοποιώντας Β.Δ. και κρυπτογράφηση.

1.10 Μέθοδοι Πιστοποίησης.

1.10.1 Μέθοδοι Πιστοποίησης Χρηστών.

Με τους παρακάτω τρόπους μπορούμε να επιτύχουμε Πιστοποίηση Των Χρηστών:

- Μέσω Script της Php
- Μέσω Β.Δ.
- Μέσω Password Files
- Μέσω απλών αρχείων configuration
- Μέσω διαφόρων πρωτοκόλλων π.χ. LDAP,KERBEROS
- Μέσω διάφορων πρωτοκόλλων που προσφέρουν Κρυπτογράφηση π.χ. S/KEY, PGP, SSL.

LDAP

Το X.509 σχεδιάστηκε για να παρέχει την υποδομή πιστοποίησης στις υπηρεσίες καταλόγου του X.500 (ldap). Η πρώτη έκδοση του X.509 δημοσιεύτηκε το 1988, καθιστώντας το έτσι την παλαιότερη πρόταση για μία παγκόσμια Υποδομή Δημοσίου Κλειδιού.

Το γεγονός αυτό σε συνδυασμό με την υποστήριξη του προτύπου από τον Διεθνή Οργανισμό Τυποποίησης (International Standards Organization - ISO) και την Διεθνή Ένωση Τηλεπικοινωνιών (Internation Telecommunications Union - ITU) έχουν οδηγήσει στην υιοθέτηση του X.509 από μεγάλο αριθμό οργανισμών και κατασκευαστών.

Η Visa και η Mastercard έχουν επιλέξει το X.509 για το Secure Electronic Transactions (SET) πρότυπο, και η Netscape υιοθέτησε το X.509 πρότυπο για την έκδοση των πιστοποιητικών που χρησιμοποιούνται στο Secure Sockets Layer πρωτόκολλο.

Η έκδοση 3 του X.509 επεκτείνει σε μεγάλο βαθμό την λειτουργικότητα του προτύπου και γι αυτό είναι ιδιαίτερα διαδεδομένο και χρησιμοποιείται σε πλοηγητές ιστοσελίδων (web browsers), εξυπηρετητές και προγράμματα λογισμικού για την διαχείριση του ηλεκτρονικού ταχυδρομείου (mail server/clients) κτλ από πολλές γνωστές εταιρίες ανάπτυξης λογισμικού.

Kerberos

Ένα άλλο σύστημα που χρησιμοποιείται πολύ, είναι η συνοδεία του ονόματος του χρήστη από έναν μυστικό κωδικό. Σε αυτό το εναλλακτικό σχήμα πιστοποίησης ταυτότητας υπάρχουν δύο, τουλάχιστον, διαφορετικά μειονεκτήματα. Πρώτον, αποτελεί χάσιμο χρόνο για τον χρήστη. Δεύτερον και σημαντικότερον, είναι ευάλωτο σε επιθέσεις παθητικού τύπου *(passive attacks)*, καθ' ότι ο κωδικός διανύει τη δίκτυο μη κρυπτογραφημένος. (Kerberos)

S/KEY'
S/KEY (Secure KEY), One-time Password System

Τα υπολογιστικά συστήματα καθημερινά απειλούνται από χιλιάδες εισβολείς που ανακαλύπτουν συνεχώς νέες, πιο εκλεπτυσμένες μεθόδους επίθεσης. Ένα από τα πιο συνηθισμένα είδη επίθεσης είναι η παράνομη καταγραφή της κυκλοφορίας σε καίρια σημεία του δικτύου και εκμετάλλευση των αποκτηθέντων πληροφοριών για την εξαγωγή μυστικών κωδικών για νόμιμους χρήστες. Η Bellcore έχει αναπτύξει ένα πρότυπο λογισμικό, το S/KEY, που αποτελεί ένα σύστημα παραγωγής κωδικών μίας χρήσης, για αντιμετωπίσει αυτό το είδος επίθεσης.

Το σύστημα S/KEY έχει αρκετά πλεονεκτήματα σε σχέση με άλλα συστήματα πιστοποίησης ταυτότητας . Κατ' αρχήν, ο κωδικός του χρήστη δεν ταξιδεύει ποτέ στο δίκτυο κατά την διάρκεια του login ή όταν εκτελούνται εντολές όπως οι passwd και η su του UNIX. Μυστικές, ευαίσθητες πληροφορίες δεν αποθηκεύονται πουθενά, ούτε στον υπολογιστή που χρησιμοποιεί ο χρήστης και οι αλγόριθμοι που εφαρμόζονται είναι ευρέως γνωστοί.

Η *Bellcore* πειραματίζεται με το σύστημα S/KEY εδώ και δύο χρόνια, το οποίο είναι διαθέσιμο στο Internet με anonymous ftp.

Υπάρχει μια μεγάλη ποικιλία απειλών που μπορούμε να σκεφτούμε για ένα δίκτυο. Αυτές διαχωρίζονται σε εσωτερικές απειλές και σε εξωτερικές. Το S/KEY έχει αναπτυχθεί για καταπολεμήσει τις εξωτερικές απειλές, δηλαδή τις προσπάθειες για εισχώρηση σε ένα σύστημα υπολογιστών από πηγές εκτός των ορίων του συστήματος. Δεν ασχολείται με τα επιπρόσθετα μέτρα ασφαλείας που πρέπει να ληφθούν υπόψη για να εμποδιστούν νόμιμοι χρήστες να αποκτήσουν παραπάνω δικαιώματα από αυτά που δικαιούνται. Προστατεύει τους κωδικούς των χρηστών από τις *passive attacks*, επιθέσεις κατά τις οποίες ο πιθανός εισβολέας παρακολουθεί τις συναλλαγές νόμιμων χρηστών και συλλέγει κωδικούς και άλλες χρήσιμες πληροφορίες, που θα χρησιμοποιήσει αργότερα.

Το S/KEY μπορεί εύκολα και γρήγορα να προστεθεί σε σχεδόν όλα τα UNIX συστήματα , χωρίς να απαιτεί επιπλέον hardware και χωρίς να αποθηκεύει ευαίσθητες πληροφορίες. Μπορεί να χρησιμοποιηθεί σε "χαζά τερματικά" *("dumb terminals")*, σε προσωπικούς υπολογιστές που έχουν εγκατεστημένα συμβατικά επικοινωνιακά προγράμματα, ή σε σταθμούς εργασίας *(workstations)*. Είναι συμβατό με πιθανή εφαρμογή βασισμένη σε *smart cards* ή *pocket calculators*.

Pretty Good Privacy (PGP)

Το Pretty Good Privacy ή PGP αποτελεί ένα κρυπτοσύστημα που δημιουργήθηκε από τον Phil Zimmerman και χρησιμοποιεί τους αλγόριθμους RSA και IDEA για την κρυπτογράφηση και υπογραφή μηνυμάτων της ηλεκτρονικής αλληλογραφίας.

Κάθε χρήστης του PGP διατηρεί μία λίστα με τα δημόσια κλειδιά των χρηστών με τους οποίους επικοινωνεί, η οποία καλείται *keyring*. Για την προστασία της λίστας, ο κάθε χρήστης την υπογράφει με το ιδιωτικό του κλειδί. Κάθε κλειδί που προστίθεται στην λίστα είναι δυνατό να φέρει έναν από τους εξής χαρακτηρισμούς:

- Απολύτως Έμπιστο (Completely Trusted)
- Μερικώς Έμπιστο (Marginally Trusted)
- Μη Έμπιστο (Untrusted)
- Αγνωστο (Unknown)

Το PGP επιτρέπει την ανταλλαγή keyrings, ενώ ο κάθε χρήστης έχει τη δυνατότητα να ρυθμίσει το επίπεδο εμπιστοσύνης για την αποδοχή ενός νέου κλειδιού. Δηλαδή, ο χρήστης μπορεί να θεωρήσει την οντότητα του κλειδιού έμπιστη, αν το κλειδί έχει ήδη υπογραφεί από δύο απολύτως έμπιστα (Completely Trusted) κλειδιά ή από τρία μερικώς έμπιστα (Marginally Trusted) κλειδιά.

Καθώς οι χρήστες ανταλλάσσουν keyrings σχηματίζουν έναν ιστό εμπιστοσύνης (web of trust). Κάθε χρήστης αποτελεί αρχή πιστοποίησης του εαυτού του και είναι υπεύθυνος για το μοντέλο εμπιστοσύνης που επιλέγει. Το απλό αυτό μοντέλο έχει επιτρέψει στο PGP να κερδίσει μία σχετικά μεγάλη αποδοχή στο Διαδίκτυο. Παρόλα αυτά, η Υποδομή Δημοσίου Κλειδιού του PGP δεν είναι κατάλληλη για εφαρμογές ηλεκτρονικού εμπορίου και για εφαρμογές που απαιτούν ισχυρή ταυτοποίηση.

Τα πιστοποιητικά του PGP δεν είναι επεκτάσιμα και περιέχουν μόνο μία διεύθυνση ηλεκτρονικής αλληλογραφίας, την τιμή ενός δημόσιου κλειδιού και ένα χαρακτηριστικό του βαθμού της εμπιστοσύνης. Καθώς η διεύθυνση ηλεκτρονικής αλληλογραφίας δεν μπορεί να αποτελέσει έναν ακριβή τρόπο του προσδιορισμού της ταυτότητας ενός χρήστη, το PGP δεν μπορεί να παρέχει ισχυρή ταυτοποίηση (strong authentication). Η έλλειψη επεκτασιμότητας των πιστοποιητικών του PGP

τα καθιστά ακατάλληλα για άλλες εφαρμογές εκτός της ηλεκτρονικής αλληλογραφίας.

Το PGP δεν υποστηρίζει κάποια μέθοδο επαλήθευσης και ανάκλησης των πιστοποιητικών. Οι διαδικασίες αυτές πραγματοποιούνται μόνο μέσω άμεσης επικοινωνίας των χρηστών.

SSL (Secure Sockets Layer)

Το πρωτόκολλο SSL σχεδιάστηκε αρχικά από την Netscape για να διευκολύνει την ασφαλή επικοινωνία μεταξύ Web διακομιστή και Web browser. Από τότε έχει υιοθετηθεί ως μη τυπική μέθοδος ανταλλαγής μυστικών πληροφοριών για τους browser και τους διακομιστές.

Η SSL έκδοση 2 και η 3 υποστηρίζονται καλά. Οι περισσότεροι Web browser, είτε περιλαμβάνουν SSL είτε τη δέχονται ως μια πρόσθετη λειτουργική μονάδα. Ο Internet Explorer και ο Netscape Navigator υποστηρίζουν και οι δύο SSL από την έκδοση 3.

Τα πρωτόκολλα δικτύου και τα προγράμματα που τα χειρίζονται, συνήθως διατάσσονται ως μια στοίβα από επίπεδα. Κάθε επίπεδο μπορεί να περάσει δεδομένα στο επάνω ή κάτω επίπεδο και να ζητήσουν υπηρεσίες από το επάνω ή το κάτω επίπεδο. Παρακάτω βλέπουμε μια τέτοια στοίβα πρωτοκόλλων.

HTTP FTP SMTP	**Επίπεδο Εφαρμογής**
TCP/UDP	**Επίπεδο Μεταφοράς**
IP	**Επίπεδο Δικτύου**
Διάφορα	**Επίπεδο Κύριου Υπολογιστή σε Δίκτυο**

Η στοίβα των πρωτοκόλλων που χρησιμοποιείται από ένα πρωτόκολλο επιπέδου εφαρμογής όπως το Hypertext Transfer protocol.

Όταν χρησιμοποιείτε HTTP για να μεταφέρετε πληροφορίες, το HTTP πρωτόκολλο καλεί το TCP (Transmission Control Protocol), που με τη σειρά του βασίζεται στο IP (Internet Protocol). Αυτό το πρωτόκολλο με τη σειρά του, χρειάζεται ένα κατάλληλο πρωτόκολλο για το υλικό του δικτύου που

χρησιμοποιείται για να παίρνει πακέτα από δεδομένα και να τα στέλνει σαν ηλεκτρικά σήματα στον προορισμό μας.

Το HTTP ονομάζεται πρωτόκολλο επιπέδου εφαρμογής. Υπάρχουν πολλά άλλα πρωτόκολλα επιπέδου εφαρμογής, όπως το FTP, SMTP και Telnet (όπως φαίνεται στην εικόνα) και άλλα, όπως το POP και το IMAP. Το TCP είναι ένα από τα δύο πρωτόκολλα επιπέδου μεταφοράς που χρησιμοποιούνται σε δίκτυα TCP/IP. Το IP είναι το πρωτόκολλο επιπέδου δικτύου. Το επίπεδο κύριου υπολογιστή σε δίκτυο είναι υπεύθυνο να συνδέει τον κύριο υπολογιστή μας σε ένα δίκτυο. Το πρωτόκολλο TCP/IP δεν καθορίζει τα πρωτόκολλα που χρησιμοποιούνται για αυτό το επίπεδο, καθώς χρειαζόμαστε διαφορετικά πρωτόκολλα για διαφορετικούς τύπους δικτύων.

Όταν στέλνουμε δεδομένα, τα δεδομένα στέλνονται μέσω της στοίβας από μια εφαρμογή, στα φυσικά μέσα του δικτύου. Όταν λαμβάνουμε δεδομένα, τα δεδομένα ταξιδεύουν από το φυσικό δίκτυο, μέσω της στοίβας, στην εφαρμογή.

Χρησιμοποιώντας SSL, προσθέτουμε ένα επιπλέον διαφανές επίπεδο σε αυτό το μοντέλο. Το επίπεδο SSL βρίσκεται μεταξύ του επιπέδου μεταφοράς και του επιπέδου εφαρμογής. Αυτό φαίνεται στο παρακάτω σχήμα . Το επίπεδο SSL τροποποιεί τα δεδομένα από την HTTP εφαρμογή μας, πριν τα μεταφέρει στο επίπεδο μεταφοράς, για να τα στείλει στον προορισμό τους.

SSL

HTTP	Handshake Protocol	Change Cipher	Alert Protocol	Επίπεδο Εφαρμογής
SSL	Record Protocol			Επίπεδο SSL
TCP				Επίπεδο Μεταφοράς
IP				Επίπεδο Δικτύου
Κύριος Υπολογιστής σε Δίκτυο				Επίπεδο Κύριου Υπολογιστή σε Δίκτυο

Το SSL προσθέτει ένα επιπλέον επίπεδο στη στοίβα του πρωτοκόλλου, όπως επίσης και πρωτόκολλα επιπέδου εφαρμογής για έλεγχο των δικών τους λειτουργιών.

v ΚΕΦΑΛΑΙΟ 2°

2.1 Εισαγωγή Στην PHP.

Πρώτα από όλα θα μιλήσουμε για το τι είναι η PHP. Η PHP, της οποίας τα αρχικά αντιπροσωπεύουν το "PHP: Hypertext Preprocessor" είναι μια ευρέως χρησιμοποιούμενη, ανοιχτού κώδικα, γενικού σκοπού scripting γλώσσα προγραμματισμού, η οποία είναι ειδικά κατάλληλη για ανάπτυξη εφαρμογών για το Web και μπορεί να ενσωματωθεί στην HTML.

Απλή απάντηση, αλλά τι σημαίνει; Ένα παράδειγμα:

Παράδειγμα. Ένα εισαγωγικό παράδειγμα

```
    <html>
   <head>

<title>Example</title>
   </head>
   <body>

       <?php
       echo "Hi, I'm a
PHP script!";
       ?>

   </body>
</html>
```

Παρατηρήστε πως αυτό είναι διαφορετικό από ένα script γραμμένο σε άλλες γλώσσες προγραμματισμού όπως η Perl ή η C : Αντί να γράφετε ένα πρόγραμμα με πολλές εντολές για να εξάγετε HTML, γράφετε ένα HTML script με κάποιο ενσωματωμένο κώδικα για να κάνει κάτι (σε αυτή την περίπτωση, να

εμφανίζει κάποιο κείμενο). Ο κώδικας PHP είναι εσώκλειστος σε ειδικά tags (ετικέτες) αρχής και τέλους που σας επιτρέπουν να μεταφέρεστε μέσα και έξω από το "PHP mode" (PHP τρόπο λειτουργίας).Αυτό που διαχωρίζει την PHP από κάτι σαν client-side Javascript είναι ότι ο κώδικας εκτελείται στον server (εξηπηρετητή). Αν είχατε ένα script σαν το παραπάνω στον server σας, ο client θα έπαιρνε τα αποτελέσματα της εκτέλεσης αυτού του script, χωρίς να υπάρχει κανένας τρόπος να καταλάβει τι κώδικας υπάρχει από κάτω. Μπορείτε ακόμη να ρυθμίσετε τον web server σας να χειρίζεται όλα τα HTML αρχεία σας με την PHP, και τότε πραγματικά δεν υπάρχει τρόπος ο χρήστης να καταλάβει τι έχετε κάτω από το μανίκι σας. Τα καλύτερο πράγμα στην PHP είναι ότι είναι εξαιρετικά απλή για ένα νεοφερμένο αλλά προσφέρει πολλά προηγμένα χαρακτηριστικά για ένα επαγγελματία προγραμματιστή. Μην τρομάζετε όταν διαβάζετε την μακροσκελή λίστα με τα χαρακτηριστικά της PHP. Μπορείτε να εξοικειωθείτε μέσα σε πολύ λίγο χρόνο και να αρχίσετε να γράφετε απλά script σε λίγες ώρες.

Αν και η ανάπτυξη της PHP εστιάζεται σε server-side scripting, μπορείτε να κάνετε πολύ περισσότερα με αυτή. Βέβαια τώρα μπαίνει και ένα άλλο ερώτημα όπως το τι μπορεί να κάνει η PHP. Η PHP μπορεί να κάνει οτιδήποτε και επικεντρώνεται κυρίως στο server-side scripting, έτσι μπορείτε να κάνετε οτιδήποτε ένα άλλο CGI πρόγραμμα μπορεί να κάνει, όπως να μαζέψει δεδομένα, να παράγει δυναμικό περιεχόμενο σελίδων, ή να στείλει και να πάρει cookies. Αλλά η PHP μπορεί να κάνει πολύ περισσότερα.

Υπάρχουν τρεις κύριοι τομείς που χρησιμοποιείται ένα PHP script.

- Server-side scripting. Αυτό είναι το πιο παραδοσιακό και το κύριο πεδίο για την PHP. Χρειάζεστε τρία πράγματα για να δουλέψει αυτό. Τον PHP μεταγλωττιστή (parser) (CGI ή server module), ένα webserver (εξηπηρετητή σελίδων) και ένα web browser ("φυλλομετρητή"). Πρέπει να τρέξετε τον webserver, με μια συνδεδεμένη εγκατάσταση της PHP. Μπορείτε να προσπελάσετε τα αποτελέσματα του PHP προγράμματος με ένα web browser, βλέποντας την σελίδα PHP μέσα από τον server.

- Command line scripting. Μπορείτε να φτιάξετε ένα PHP script για να το τρέχετε χωρίς server ή browser. Χρειάζεστε μόνο τον PHP μεταγλωττιστή για να την χρησιμοποιήσετε με αυτό τον τρόπο. Αυτός ο τύπος είναι ιδανικός για script που εκτελούνται συχνά με τη χρήση της cron (σε *nix ή Linux) ή με τον Task Scheduler (στα Windows). Αυτά τα script μπορούν επίσης να χρησιμοποιηθούν για απλές εργασίες επεξεργασίες κειμένου.

- Εγγραφή client-side GUI εφαρμογών (Γραφικά περιβάλλοντα χρηστών). Η PHP ίσως να μην είναι η πιο καλή γλώσσα για να γράψει κανείς παραθυριακές εφαρμογές, αλλά αν ξέρετε PHP πολύ καλά και θέλετε να χρησιμοποιήσετε κάποια προχωρημένα χαρακτηριστικά της PHP στις client-side εφαρμογές σας,

μπορείτε επίσης να χρησιμοποιήσετε το PHP-GTK για αυτού του είδους τα προγράμματα. Έχετε επίσης τη δυνατότητα να γράφετε cross-platform εφαρμογές με αυτό τον τρόπο. Το PHP-GTK είναι μια επέκταση της PHP και δεν συμπεριλαμβάνεται στην κύρια διανομή.Η PHP μπορεί να χρησιμοποιηθεί σε όλα τα κύρια λειτουργικά συστήματα, συμπεριλαμβανομένου του Linux, πολλών εκδοχών του Unix (HP-UX, Solaris και OpenBSD), Microsoft Windows, Mac OS X, RISC OS και πιθανώς σε άλλα. Η PHP υποστηρίζει επίσης τους Apache, Microsoft Internet Information Server, Personal Web Server, Netscape και iPlanet servers, Oreilly Website Pro server, Caudium, Xitami, OmniHTTPd, και πολλούς άλλους webserver. Για την πλειοψηφία των server η PHP έχει ένα module, για τους υπόλοιπους η PHP μπορεί να λειτουργήσει ως ένας CGI επεξεργαστής.

Έτσι με την PHP έχετε την ελευθερία επιλογής ενός λειτουργικού συστήματος και ενός web server. Επιπλέον, έχετε επίσης την ελευθερία να χρησιμοποιήσετε συναρτησιακό (procedural) ή αντικειμενοστραφή (object oriented) προγραμματισμό ή μια ανάμειξη τους. Αν και η παρούσα έκδοση δεν υποστηρίζει όλα τα πρότυπα χαρακτηριστικά, μεγάλες βιβλιοθήκες κώδικα και μεγάλες εφαρμογές (συμπεριλαμβανομένης και της βιβλιοθήκης PEAR) είναι γραμμένες μόνο με αντικειμενοστραφή κώδικα.

Με την PHP δεν είστε περιορισμένοι να εξάγετε HTML. Οι δυνατότητες της PHP συμπεριλαμβάνουν την εξαγωγή εικόνων, αρχείων PDF, ακόμη και ταινίες Flash (χρησιμοποιώντας τα libswf και Ming) παράγονται αμέσως. Μπορείτε επίσης να εξάγετε εύκολα οποιοδήποτε κείμενο όπως XHTML και οποιοδήποτε άλλο XML αρχείο. Η PHP μπορεί να δημιουργεί αυτόματα αυτά τα αρχεία και να τα αποθηκεύει στο σύστημα αρχείων, αντί να τα εκτυπώνει, αποτελώντας έτσι μια server-side cache για το δυναμικό σας περιεχόμενο.

Ένα από τα πιο δυνατά και σημαντικά χαρακτηριστικά της PHP είναι η υποστήριξη που έχει για ένα μεγάλο σύνολο βάσεων δεδομένων. Η συγγραφή μιας σελίδας που υποστηρίζει βάσεις δεδομένων είναι εξαιρετικά απλή. Οι εξής βάσεις δεδομένων υποστηρίζονται μέχρι στιγμής:

Adabas D	Ingres	Oracle (OCI7 and OCI8)
dBase	InterBase	Ovrimos
Empress	FrontBase	PostgreSQL
FilePro (read-only)	mSQL	Solid
Hyperwave	Direct MS-SQL	Sybase
IBM DB2	MySQL	Velocis

Informix ODBC Unix dbm

Έχουμε επίσης μια αφαιρετική επέκταση DBX βάσεων δεδομένων (DBX database abstraction extension) που σας επιτρέπει διάφανα να χρησιμοποιείτε οποιαδήποτε βάση δεδομένων υποστηρίζεται από αυτή την επέκταση. Επιπλέον η PHP υποστηρίζει το ODBC, το Open Database Connection standard (Ανοιχτό πρότυπο Σύνδεσης Βάσεων δεδομένων) έτσι μπορείτε να συνδεθείτε σε οποιαδήποτε βάση δεδομένων που υποστηρίζει αυτό το παγκόσμιο πρότυπο.

Η PHP έχει επίσης υποστήριξη για επικοινωνία με άλλες υπηρεσίες χρησιμοποιώντας πρωτόκολλα όπως LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (στα Windows) και αμέτρητα άλλα. Μπορείτε επίσης να ανοίξετε raw network sockets και να αλληλεπιδράσετε με οποιοδήποτε άλλο πρωτόκολλο. Η PHP έχει ακόμη υποστήριξη για την περίπλοκη ανταλλαγή δεδομένων WDDX μεταξύ σχεδόν όλων των Web programming γλωσσών. Μιλώντας για δια-επικοινωνία, η PHP υποστηρίζει instantiation αντικειμένων Java και τα χρησιμοποιεί διάφανα σαν αντικείμενα PHP. Μπορείτε επίσης να χρησιμοποιήσετε την CORBA επέκταση μας για να προσπελάσετε remote (απομακρυσμένα) αντικείμενα.

Η PHP έχει εξαιρετικά χρήσιμα χαρακτηριστικά επεξεργασίας κειμένων, από την POSIX επέκταση ή τις Perl regular expressions μέχρι XML parsing αρχείων. Για τη μεταγλώττιση και την πρόσβαση αρχείων XML, υποστηρίζουμε τα πρότυπα SAX και DOM. Μπορείτε να χρησιμοποιήσετε την XSLT επέκταση μας για να μετατρέπετε τα XML αρχεία σε άλλες μορφές.

Καθώς χρησιμοποιείτε την PHP στον τομέα του ecommerce, θα βρείτε τις Cybercash payment, CyberMUT, VeriSign Payflow Pro και CCVS συναρτήσεις χρήσιμες για τα online προγράμματα πληρωμής σας.

Τελευταίο αλλά σημαντικό, έχουμε πολλές άλλες ενδιαφέρουσες επεκτάσεις, τις mnoGoSearch search engine συναρτήσεις, πολλά εργαλεία συμπίεσης (gzip, bz2), μετατροπές ημερολογίου, μεταφράσεις.

Μερικά πλεονεκτήματα και χαρακτηριστικά της PHP αναφέρονται παρακάτω. Σήμερα η δημοτικότητα της PHP είναι τόσο μεγάλη, που πολλοί θέλουν να την μάθουν. Η PHP είναι μια τυπική λειτουργία που προσφέρεται από τις περισσότερες εταιρείες Web φιλοξενίας. Ωστόσο, είναι ενδιαφέρον να καταλάβετε γιατί τόσα άτομα επιλέγουν την PHP αντί για τις άλλες εναλλακτικές λύσεις.

Η Perl προσαρμόστηκε καλά για να παρέχει μια CGI λύση. Η Microsoft παρέχει τις Active Server Pages μαζί με τον Internet Information Server. Άλλα προϊόντα, όπως το Gold Fusion της Macromedia, αποτελούν μια εναλλακτική λύση. Το ServerWatch.com αναφέρει εκατοντάδες Web τεχνολογίες, από τις οποίες κάποιες κοστίζουν δεκάδες χιλιάδες δολάρια.

2.1.1 Τα Πλεονεκτήματα Της PHP.

Η σύντομη απάντηση είναι ότι η PHP είναι η καλύτερη. Είναι η γρηγορότερη στην κωδικοποίηση και εκτελείται γρηγορότερα. Ο ίδιος κώδικας PHP τρέχει όπως είναι σε διαφορετικούς Web διακομιστείς και σε διαφορετικά λειτουργικά συστήματα. Επιπλέον, η λειτουργικότητα της PHP είναι πρόσθετη σε άλλα περιβάλλοντα. Ακολουθεί μια λεπτομερής συζήτηση.

Η PHP είναι δωρεάν. Ο καθένας μπορεί να επισκεφθεί την PHP Web τοποθεσία **http://www.php.net** και να μεταφέρει τον πλήρη κώδικα προέλευσης, με άδεια χρήσης στυλ BSD **http://www.php.net.licence.** Είναι επίσης διαθέσιμα δυαδικά αρχεία για τα windows. Το αποτέλεσμα είναι ότι μπορείτε να αποκτήσετε εμπειρία πολύ εύκολα. Υπάρχει πολύ μικρός κίνδυνος όταν δοκιμάζετε την PHP και η άδειά της επιτρέπει να χρησιμοποιηθεί ο κώδικας για να αναπτυχθούν έργα, χωρίς πληρωμή δικαιωμάτων. Αυτό δε μοιάζει μα άλλα προϊόντα όπως το Cold Fusion της Allaire, που κοστίζει χιλιάδες δολάρια για να μπορεί το λογισμικό να μεταφραστεί και να εξυπηρετεί script. Ακόμα και οι γίγαντες της αγοράς, όπως Netscape και η IBM, αναγνωρίζουν τώρα τα πλεονεκτήματα της ελεύθερης διάθεσης του κώδικα προέλευσης **php.net** και να μεταφέρει τον πλήρη κώδικα προέλευσης, με άδεια χρήσης.

Η PHP τρέχει σε UNIX, Widows και Macintosh OS X. Η PHP έχει σχεδιασθεί να δουλεύει με τον Apache Web διακομιστή. Ο Apache, μια άλλη δωρεάν τεχνολογία είναι ο πιο δημοφιλής Web διακομιστής στο Internet και έρχεται με κώδικα προέλευσης για UNIX και Windows. Η PHP δουλεύει και με άλλους Web διακομιστές, όπως τον Internet Information Server της Microsoft. Τα script μπορούν να μετακινούνται μεταξύ διακομιστών χωρίς αλλαγή. Η PHP υποστηρίζει ISAPI για να παρέχει τα πλεονεκτήματα απόδοσης της σύνδεσης της με Microsoft Web διακομιστές.

Η PHP είναι τροποποιήσιμη. Η PHP έχει σχεδιασθεί να επιτρέπει μελλοντικές επεκτάσεις στην λειτουργικότητα της. Είναι γραμμένη σε C και παρέχει ένα καλά ορισμένο περιβάλλον προγραμματισμού(API). Οι ικανοί προγραμματιστές μπορούν να προσθέσουν εύκολα νέα λειτουργικότητα στην PHP. Το μεγάλο σύνολο συναρτήσεων που είναι διαθέσιμο στην PHP είναι απόδειξη ότι το κάνουν συχνά. Ακόμα και αν δεν ενδιαφέρεστε να αλλάξετε τον κώδικα προέλευσης, είναι ωραίο να ξέρετε ότι μπορείτε να τον δείτε. Αυτό ίσως να σας δώσει περισσότερη εμπιστοσύνη στην δύναμη της PHP.

Η PHP γράφθηκε για δημιουργία ιστοσελίδων. Η Perl, η C και η Java είναι πολύ καλές γενικές γλώσσες και είναι ικανές να καθοδηγούν Web εφαρμογές. Αυτό που θυσιάζουν, δυστυχώς αυτές οι εναλλακτικές λύσεις, είναι η ευκολία

στην επικοινωνία με το Web. Οι PHP εφαρμογές μπορούν να αναπτυχθούν γρήγορα και εύκολα επειδή ο κώδικας ενσωματώνεται στις ίδιες τις ιστοσελίδες.

Η υποστήριξη για την PHP είναι δωρεάν και εύκολα διαθέσιμη. Τα ερωτήματα στις ταχυδρομικές λίστες συνήθως απαντώνται μέσα σε λίγα λεπτά, ένα προσαρμοσμένο σύστημα παρακολούθησης λαθών στην PHP τοποθεσία δείχνει κάθε πρόβλημα μαζί με τη λύση του. Πολλές τοποθεσίες, όπως η phpbuilder.com και η zend.com προσφέρουν περιεχόμενα για PHP προγραμματιστές.

Η PHP είναι δημοφιλής και οι εταιρείες παροχής υπηρεσιών την βρίσκουν μια καλή λύση για να επιτρέπουν στους πελάτες τους να κωδικοποιούν Web εφαρμογές χωρίς το κίνδυνο πού έχουν τα CGI script. Προγραμματιστές σε όλο τον κόσμο προσφέρονται να προγραμματίσουν σε PHP. Διάφορες τοποθεσίες γραμμένες σε PHP θα έχουν την επιλογή να μετακινούνται από ένα υπολογιστή σε ένα άλλο υπολογιστή, όπως επίσης και την επιλογή να τους προσθέτουν οι προγραμματιστές επιπλέον λειτουργικότητα.

Οι ικανότητες προγραμματισμού που έχουν αναπτυχθεί σε άλλες δομημένες γλώσσες, μπορούν να εφαρμοσθούν και στην PHP, η οποία εμπνέεται από την Perl και την C. Οι έμπειροι προγραμματιστές της Perl και της C μαθαίνουν PHP πολύ γρήγορα και παρόμοια και οι προγραμματιστές που μαθαίνουν PHP ως πρώτη γλώσσα μπορεί να εφαρμόσουν τις γνώσεις τους όχι μόνο στην Perl και στην C αλλά και σε άλλες παρόμοιες γλώσσες με την C, όπως και στην Java.

2.2 Διασυνδέσεις Με Εξωτερικα Συστήματα.

Αρχικά η PHP έγινε διάσημη για την διασύνδεσή της με πολλές διαφορετικές βάσεις δεδομένων, αλλά επίσης υποστηρίζει και άλλα εξωτερικά συστήματα. Η υποστήριξη γίνεται με την μορφή λειτουργικών μονάδων, που ονομάζονται επεκτάσεις. Αυτές μεταγλωττίζονται κατευθείαν στην PHP ή φορτώνονται δυναμικά. Τακτικά, προστίθενται νέες επεκτάσεις στη PHP. Οι επεκτάσεις παρέχουν ομάδες από συναρτήσεις για χρήση αυτών των εξωτερικών συστημάτων. Όπως αναφέρθηκε και παραπάνω μερικά από αυτά είναι και βάσεις δεδομένων. Η PHP προσφέρει συναρτήσεις για να μιλά εγγενώς με τις περισσότερες βάσεις δεδομένων και παρέχει πρόσβαση σε προγράμματα οδήγησης ODBC. Άλλες επεκτάσεις σας δίνουν τη δυνατότητα να στέλνετε μηνύματα χρησιμοποιώντας ένα συγκεκριμένο πρωτόκολλο δικτύου. Όπως το LDAP ή το IMAP.

Η Pspell είναι ένα σύστημα για έλεγχο ορθογραφίας. Μια επέκταση παρέχει υποστήριξη για αριθμούς με άπειρη ακρίβεια. Υπάρχει μια επέκταση για διάφορα

συστήματα ημερολογίων. Μια επέκταση παρέχει υποστήριξη για βάσεις δεδομένων στυλ DBM. Μπορείτε να χρησιμοποιήσετε τα πρωτόκολλα SNMP,IMAP και LDAP. Οι βάσεις δεδομένων Interbase και Informix υποστηρίζονται εγγενώς, όπως και η mSQL, MYSQL, MS SQL, Sybase, Oracle και PostgreSQL. Μπορείτε επίσης να αναλύσετε XML ή να δημιουργήσετε WDDX πακέτα. Μπορείτε ακόμα να εξάγετε μετά πληροφορίες για τις ψηφιακές εικόνες σας χρησιμοποιώντας την επέκταση EXIF.

2.3 Πληροφορίες Για Το Πως Δουλέυει Η PHP Με Ένα WEB Διακομιστή.

Η κανονική διαδικασία που ακολουθεί ένας Web διακομιστής όταν παραδίδει μια σελίδα σε ένα browser είναι ο εξής. Όλα ξεκινούν όταν ο browser ζητά μια ιστοσελίδα. Ανάλογα με το URL, ο browser βρίσκει την διεύθυνση του Web διακομιστή, προσδιορίζει την σελίδα που θέλει και δίνει όποιες άλλες πληροφορίες ζητά ο Web διακομιστής. Μερικές από αυτές τις πληροφορίες είναι για τον ίδιο τον browser, όπως το όνομα του (Mozilla), η έκδοση του (4,08), ή το λειτουργικό σύστημα (Linux). Άλλες πληροφορίες που δίνονται στον Web διακομιστή μπορεί να είναι το κείμενο που πληκτρολόγησε ο χρήστης σε πεδία μιας φόρμας.

Αν η αίτηση είναι για ένα HTML αρχείο, ο Web διακομιστής θα βρει απλώς το αρχείο, θα πει στον browser να περιμένει κάποιο HTML κείμενο και μετά θα στείλει τα περιεχόμενα του αρχείου. Ο browser παίρνει τα περιεχόμενα και αρχίζει να εμφανίζει την σελίδα, ανάλογα με τον HTML κώδικα , αν έχετε προγραμματίσει σε HTML για κάποιο χρονικό διάστημα, αυτό θα σας είναι σαφές.

Ελπίζουμε να έχετε επίσης κάποια εμπειρία με CGI script. Όταν ένας Web διακομιστής παίρνει μια αίτηση για ένα CGI, δεν μπορεί απλώς να στείλει τα περιεχόμενα του αρχείου. Πρέπει να εκτελέσει πρώτα το script. Το script θα δημιουργήσει κάποιο HTML κώδικα, που μετά θα σταλεί στον browser. Σε σχέση με τον browser, απλώς παίρνει HTML.

Όταν ζητείται μια PHP σελίδα, γίνεται η επεξεργασία της όπως γίνεται για ένα CGI script, τουλάχιστον όσον αφορά στο γεγονός ότι το script δεν στέλνεται απλώς στον browser. Περνά πρώτα από την PHP μηχανή , που δίνει το HTML κείμενο στον Web διακομιστή.

2.4 Απαιτήσεις Υλικού Και Λογισμικού.

Ένα μεγάλο πλεονέκτημα του λογισμικού Ανοικτού Κώδικα (Open Source) είναι ότι έχει την δυνατότητα προσαρμογής σε νέα περιβάλλοντα. Αυτό ισχύει και για την PHP. Αν και αρχικά είχε στόχο να γίνει μια λειτουργική μονάδα για τον Apache Web διακομιστή, η PHP από τότε έχει γίνει πιο αφηρημένη σε σχέση με την διασύνδεση με τον Web διακομιστή. Το νέο επίπεδο αφαιρετικότητας επέτρεψε να γραφεί μια ISAPI λειτουργική μονάδα, με την οποία μπορεί να δουλεύει εξίσου καλά και με τον Internet Information Server της Microsoft. Σε σχέση με τις απαιτήσεις υλικού, προσωπικά έχω δει την PHP να τρέχει σε υπολογιστές 100-MHz Pentium με Slack ware Linux και Windows NT, αντίστοιχα. Η απόδοση ήταν θαυμάσια για ένα προσωπικό περιβάλλον ανάπτυξης. Πρέπει να έχει βοηθήσει το γεγονός ότι οι μηχανές των PHP 3και 4 αναπτύχθήκαν σε Intel 486 CPU. Μια τοποθεσία που αναμένεται να λαμβάνει χιλιάδες αιτήσεις την ημέρα, θα χρειαστεί φυσικά, γρηγορότερο υλικό. Αν και απαιτούνται περισσότεροι πόροι σε μια PHP τοποθεσία από ότι σε μια απλή HTML τοποθεσία, οι απαιτήσεις δεν είναι δραματικά διαφορετικές. Ανεξάρτητα από το παράδειγμά μου δεν είναι κανείς περιορισμένος σε υλικό Intel. Η PHP δουλεύει εξίσου καλά σε PowerPC, Sparc και σε άλλες 32-bit CPU ή σε καλύτερες.

Όταν επιλέγετε ένα λειτουργικό σύστημα, έχετε την γενική επιλογή μεταξύ Windows και ενός λειτουργικού συστήματος βασισμένου στο UNIX. Η PHP θα τρέξει και σε παλαιότερα συστήματα των Windows, αν και αυτά τα λειτουργικά συστήματα δεν είναι κατάλληλα για Web διακομιστές με πολλή κίνηση. Θα τρέξει επίσης σε Windows 2000 και Windows XP. Σε UNIX λειτουργικά συστήματα, η PHP δουλεύει καλά με Linyx και Solaris, όπως και με άλλα συστήματα. Αν έχετε επιλέξει ένα σύστημα βασισμένο σε PPC, όπως ένα Macintosh, μπορείτε να επιλέξετε το Linux PPc, μια έκδοση του Linux. Ο Chad Cunningham έδωσε διορθώσεις για να μπορεί να γίνει μεταγλώττιση της PHP στο OS X ητς Apple. Υπάρχει ακόμη υποστήριξη για το OS/2 της IBM και το Novell Netware.

Η PHP δουλεύει πιο καλά με τον Apache Web διακομιστή. Αλλά τώρα δουλεύει πολύ καλά και με το IIS. Μεταγλωττίζεται επίσης ως μια λειτουργική μονάδα για τον fhttpd Web διακομιστή. Μπορείτε να κάνετε την PHP να δουλεύει σχεδόν με οποιονδήποτε Web διακομιστή χρησιμοποιώντας την CGI έκδοση, αλλά δεν σας συστήνω αυτή την διαμόρφωση για παραγωγικές Web διακομιστές.

¨ ***Εγκατάσταση Στον APACHE Για UNIX.***

Αν χρησιμοποιείτε Linux, μπορείτε να βρείτε εύκολα ένα RPM για τον Apache και την PHP,αλλά αυτή η εγκατάσταση μπορεί να μην περιλαμβάνει κάθε PHP λειτουργία που θέλετε. Προτείνω αυτή την διαδρομή για μια εύκολη εκκίνηση. Μπορείτε πάντα να μεταγλωττίσετε τον Apache και την PHP εκ του μηδενός αργότερα. Η PHP θα μεταγλωττιστεί στα περισσότερα λειτουργικά συστήματα που βασίζονται στο UNIX, συμπεριλαμβανομένου του Solaris και του Linux. Αν έχετε κάνει μεταγλωττίσεις λογισμικού που έχετε βρει στο internet , δεν θα έχετε πρόβλημα με αυτήν την εγκατάσταση. Αν δεν έχετε εμπειρία με την εξαγωγή αρχείων από ένα αρχείο tar και με την εκτέλεση αρχείων make, μπορείτε να βασιστείτε στον διαχειριστή του συστήματος σας ή σε κάποιον πιο έμπειρο. Θα χρειαστείτε δικαιώματα ρίζας για να εγκαταστήσετε πλήρως την PHP.

Το πρώτο βήμα είναι να μεταφέρετε τα tar αρχεία και να τα αποσυμπιέσετε. Μεταφέρετε τις νεώτερες εκδόσεις από την PHP τοποθεσία **http://www.php.net/downloads.php** και την Apache 2 τοποθεσία **http://httpd.apache.org/** . Την ώρα που γραφόταν το βιβλίο , το Apache 2 θεωρείτο σταθερό. Η υποστήριξη για το mod_php στον Apache δεν είναι πλήρης. Οι παρακάτω οδηγίες υποθέτουν ότι ο Apache 1.3 και ο Apache 2 μπορεί να απαιτούν μερικές αλλαγές.

Μετά τη αποσυμπίεση του tar αρχείου, το πρώτο βήμα είναι να διαμορφώσετε τον Apache. Αυτό γίνεται εκτελώντας το script διαμόρφωσης μέσα στον κατάλογο Apache. Το παράδειγμα μας δείχνει μια ελάχιστη διαμόρφωση.

Κώδικας 2.1
Διαμορφώνοντας τον Apache.

```
./configure\

--server-uid=nobody\

--enable-module=so\
```

Το script θα εξετάσει το σύστημά σας και θα προετοιμάσει ένα αρχείο make για τον Apache. Αυτό κάνει τον Apache να μπορεί να χρησιμοποιηθεί με κοινόχρηστες βιβλιοθήκες, μια εκ των οποίων θα είναι η PHP. Θα πρέπει μετά το βήμα διαμόρφωσης να εκτελέσετε τα δυαδικά αρχεία στην προεπιλεγμένη θέση. Ίσως να θέλετε να ελέγξετε τον Apache ξεκινώντας τον με το script /usr/local/apache/bin/apachectl.

Μετά, θα διαμορφώσετε και θα μεταγλωττίσετε την PHP. Το παράδειγμα 1.2 δείχνει μια εντολή για διαμόρφωση της PHP με μερικές επεκτάσεις, που εκτελούνται μέσα στον κατάλογο του κώδικα προέλευσης της PHP. Μετά δώστε την εντολή make install. Στις περισσότερες περιπτώσεις, η PHP θα μπορέσει να βρει τις βιβλιοθήκες που χρειάζεται για τις επεκτάσεις. Στο κώδικα 2.2 χρησιμοποιούμε τις βιβλιοθήκες της MySQL που έχω στο /usr/libs, αντί για τις

MySQL βιβλιοθήκες που περιλαμβάνονται στην PHP. Μπορείτε ακόμη να πάρετε πληροφορίες εκτελώντας το ./configure –help. Το make θα δημιουργήσει την PHP βιβλιοθήκη και το make install θα τοποθετήσει την PHP λειτουργική μονάδα στον κατάλογο που έχει ο Apache για τις λειτουργικές μονάδες. Εγκαθιστά επίσης τις τελευταίες κλάσεις PEAR, που είναι ένα σύνολο με τυπικό κώδικα PHP.

Κώδικας 2.2
Διαμορφώνοντας την PHP

```
./configure \
--with-apxs=/usr/local/apache/bin/apxs \
--with-zlib \
--with-bz2 \
--with-openssl \
--with-gd \
--enable-exif \
--with-jpeg-dir=/usr \
--with-freetype-dir \
--with-t1lib \
--enable-gd-native-ttf \
--wkth-mysql=/usr
```

Για επιπλέον επιλογές διαμόρφωσης, η PHP χρησιμοποιεί ένα αρχείο που ονομάζεται php.ini. Αυτό το αρχείο θα πρέπει να βρίσκεται στο /usr/local/lib έτσι αντιγράψτε το από το κατάλογο της PHP με τον κώδικα προέλευσης.(κώδικας 2.3):

Κώδικας 2.3
Αντιγράφοντας την PHP.

```
Cp php.ini-dist /usr/local/lib/php.ini
```

Ίσως να μην χρειαστεί να τροποποιήσετε αυτό το αρχείο. Ελέγχει κάποια θέματα της PHP, όπως υποστήριξη για συμπεριφορά ιστορικού. Το τελευταίο βήμα είναι να βεβαιωθείτε ότι ο Apache καταλαβαίνει τα PHP script. Κάπου στο αρχείο διαμόρφωσης του Apache, το httpd.conf, θα χρειαστείτε μια οδηγία Addtype που αντιστοιχεί τα script που τελειώνουν με .php με το application/x-httpd-php. Θα χρειαστεί επίσης να φορτώσετε την PHP λειτουργική μονάδα. Αν οι γραμμές του κώδικα 2.4 δεν υπάρχουν στο http.conf, προσθέστε τις.

Κώδικας 2.4
Ενεργοποιώντας την Php για τον Apache.

```
LoadModule php5_module  libexec/libphp5.so

AddType application/x-httpd-php .php

AddModule mod_php5.c
```

Αυτό αναγκάζει όλα τα αρχεία με επέκταση .php να εκτελούνται ως PHP script. Ίσως να θέλετε να εισάγετε το index.php ως προκαθορισμένο έγγραφο. Όταν ξεκινήσει ο apache διακομιστής, θα επεξεργαστεί τα PHP script. Η τεκμηρίωση του Apache έχει οδηγίες για την αυτόματη εκκίνηση του Apache. Αν ο Apache έτρεχε προηγουμένως ,θα πρέπει να τον επανεκκινήσετε και να μην χρησιμοποιήσετε απλώς την εντολή kill –HUP.

¨ *Εγκατάσταση Στον APACHE Για WINDOWS.*

Η μεταγλώττιση της PHP για Windows δεν είναι μια συνηθισμένη εργασία. Οι χρήστες των Windows γενικά χρησιμοποιούν δυαδικά αρχεία, που είναι διαθέσιμα στην PHP Web τοποθεσία. Το ίδιο ισχύει για τον Apache. Και τα δυο πακέτα περιλαμβάνουν αυτοματοποιημένα προγράμματα εγκατάστασης, που κάνουν την εγκατάσταση πολύ εύκολη. Η εγκατάσταση του Apache με αυτό τον τρόπο είναι θαυμάσια. Προτιμώ να εγκαθιστώ την PHP μη αυτόματα, χρησιμοποιώντας το συμπιεσμένο αρχείο, επειδή παρέχει μεγαλύτερη ευελιξία.

Αποσυμπιέστε το αρχείο της PHP σε ένα κατάλογο. Μετά, αντιγράψτε το αρχείο php.ini-dist στο κατάλογο ρίζα στο σύστημά σας, που είναι πιθανόν ο κατάλογος c:/windows και μετονομάστε το αρχείο σε php.ini.. Όταν καλείται η PHP, κοιτάζει πρώτα σε αυτό τον κατάλογο για να βρει το php.ini. Αν και δεν χρειάζεται, ίσως να θέλετε να το τροποποιήσετε για να αλλάξετε τις παραμέτρους διαμόρφωσης, όπως την αυτόματη φόρτωση επεκτάσεων.

Το επόμενο βήμα είναι να βεβαιωθείτε ότι τα απαιτούμενα DLL αρχεία είναι στην διαδρομή σας. Ένας τρόπος είναι να αντιγράψετε τα απαιτούμενα αρχεία στον κατάλογο του συστήματος σας, όπως στον c:/windows\system32. εναλλακτικά, μπορείτε να κάνετε κλικ στο εικονίδιο system στο control panel (πίνακα ελέγχου) και να προσθέσετε τον PHP κατάλογο στην διαδρομή του συστήματος. Ο Web διακομιστής θα πρέπει να μπορεί να βρίσκει το php4ts.dll, που είναι η ρίζα του καταλόγου εγκατάστασης της PHP.

Μετά διαμορφώστε τον Apache ώστε να φορτώνει την λειτουργική μονάδα της PHP. Τροποποιήστε το httpd.conf και προσθέστε τις γραμμές του κώδικα 2.5. αυτές οι γραμμές φορτώνουν την λειτουργική μονάδα και συσχετίζουν την επέκταση .php με τα PHP script. Το τελευταίο βήμα είναι η επανεκκίνηση του Apache.

Κώδικας 2.5
Ενεργοποιώντας την Php για τον Apache στα Windows.

```
LoadModule php5_moodule c:/php/sapi/php5apache.dll

   AddType application/x-httpd-php .php

AddModule mod_php5.c
```

Τροποποίηση Script.

Τα PHP script είναι απλώς αρχεία κειμένου και μπορείτε να τα τροποποιήσετε και να τα δημιουργήσετε με τον ίδιο τρόπο με τα HTML αρχεία. Φυσικά, μπορείτε χρησιμοποιήσετε telnet στον Web διακομιστή σας και να αρχίσετε να δημιουργήσετε αρχεία με το vi. Ή μπορείτε να δημιουργήσετε αρχεία με το Notepad και να χρησιμοποιήσετε FTP για να τα μεταφέρεται ένα προς ένα. Αλλά αυτές δεν είναι ιδανικές εμπειρίες. Μια βολική λειτουργία των νεότερων επεξεργαστών είναι η ενσωματωμένη δυνατότητα για FTP. Αυτοί οι επεξεργαστές μπορούν να ανοίγουν αρχεία σε ένα απομακρυσμένο Web διακομιστή, σαν να βρίσκονταν σε μια τοπική μονάδα. Ένα κλικ τα αποθηκεύει ξανά στον απομακρυσμένο Web διακομιστή. Μια άλλη λειτουργία που θα σας αρέσει είναι η επισήμανση της σύνταξης. Αυτή χρωματίζει τις PHP λέξεις κλειδιά για να μπορείτε να διαβάζετε γρηγορότερα τον κώδικα.

Όλοι έχουν ένα επεξεργαστή που προτιμούν να χρησιμοποιήσουν με τα PHP script. Συνήθως προτιμείται ο Ultraedit, το Dreamweaver της macromedia ή το homesite για να τροποποιούν τα PHP script. Οι χρήστες των Macintosh χρησιμοποιούν και προτιμούν το Bbedit.

Σε λειτουργικό σύστημα UNIX, ίσως να προτιμάτε το emacs ή το vi, φυσικά. Μπορείτε να σκεφθείτε να χρησιμοποιήσετε και το nEdit. Μια λειτουργική μονάδα για την PHP είναι διαθέσιμη στον κατάλογο contrib. Το θέμα του ποιος

επεξεργαστής είναι καλύτερος εμφανίζεται συχνά στην ταχυδρομική λίστα για την PHP. Το διάβασμα των μηνυμάτων μπορεί να είναι διασκεδαστικό και πληροφοριακό.

Αν και συνεχίζω να χρησιμοποιώ ένα επεξεργαστή κειμένου για δημιουργία PHP εφαρμογών, πολλοί προτιμούν ένα ολοκληρωμένο περιβάλλον ανάπτυξης, που είναι γνωστό με την συντόμευση IDE (integrated development environment). Υπάρχουν αρκετά IDE που έχουν σχεδιασθεί ειδικά για την PHP. το PHPEdit είναι ένα τέτοιο παράδειγμα.

¨ ***Αλγόριθμοι.***

Όποτε χρησιμοποιούμε ένα υπολογιστή, του λέμε να εκτελέσει κάποια ενέργεια. Όταν σύρετε ένα εικονίδιο στο κάδο ανακύκλωσης στην επιφάνεια εργασίας, ζητάτε από τον υπολογιστή να αφαιρέσει το αρχείο από τον σκληρό σας δίσκο. Όταν γράφετε ένα HTML αρχείο, λέτε στον υπολογιστή σας τον σωστό τρόπο να εμφανίσει κάποιες πληροφορίες. Υπάρχουν συνήθως πολλά συνεχόμενα βήματα σε κάθε διαδικασία που εκτελεί ο υπολογιστής. Ίσως πρώτα να καθαρίζει την οθόνη με το χρώμα που καθορίσατε στην ετικέτα body. Μετά ίσως αρχίσει να γράφει κάποιο κείμενο, σε ένα συγκεκριμένο χρώμα και γραμματοσειρά. Καθώς χρησιμοποιείτε ένα υπολογιστή, δεν ξέρετε κάθε μικρό βήμα που εκτελεί, αλλά απλώς του δίνετε μια λίστα με οδηγίες που θα πρέπει να εκτελέσει με την σειρά.

Οι οδηγίες για να φτιάξετε ένα γλυκό ονομάζονται συνταγή. Οι οδηγίες για να δημιουργήσετε μια ταινία ονομάζονται σενάριο. Οι οδηγίες για ένα υπολογιστή ονομάζονται πρόγραμμα. Κάθε μια είναι γραμμένη στην δική της γλώσσα, που είναι μια υλοποίηση ενός αφηρημένου συνόλου από οδηγίες. Η πληροφορική δανείστηκε την ονομασία από τα μαθηματικά και ονομάζει τις οδηγίες ένα αλγόριθμο.

Ίσως αυτή την στιγμή να έχετε στο μυαλό σας ένα αλγόριθμο που θα θέλατε να υλοποιήσετε. Ίσως να θέλετε να εμφανίζετε πληροφορίες σε ένα Web browser οι οποίες αλλάζουν συχνά. Φανταστείτε κάτι απλό, όπως την εμφάνιση της σημερινής ημερομηνίας. Θα μπορούσατε να τροποποιείτε ένα απλό HTML αρχείο μια φορά την ημέρα. Θα μπορούσατε ακόμα να γράψετε ένα σύνολο από εντολές για να σας υπενθυμίζουν κάθε σας βήμα. Αλλά δεν μπορείτε να εκτελέσετε αυτή την εργασία μόνο με την HTML. Δεν υπάρχει ετικέτα που να σας αντιστοιχεί στην σημερινή ημερομηνία.

Η PHP είναι μια γλώσσα που σας επιτρέπει να εκφράζετε αλγορίθμους για να δημιουργείτε HTML αρχεία. Με τη PHP, μπορείτε να γράφετε οδηγίες για εμφάνιση σε ένα αρχείο που ονομάζεται script. Η γλώσσα του script είναι η PHP, μια γλώσσα που μπορείτε να καταλάβετε και εσείς και ο υπολογιστής σας.

2.5 Πως Δείχνουν Τα Script Της PHP.

Η PHP είναι μια ετικέτα μέσα σε ένα HTML αρχείο. Όπως και όλες οι HTML ετικέτες, ξεκινά με ένα σύμβολο 'μικρότερο από' ή αριστερή γωνιώδη αγκύλη (<) και τελειώνει με ένα σύμβολο 'μεγαλύτερο από' ή δεξιά γωνιώδη αγκύλη (>). Για να την ξεχωρίζει από τις άλλες ετικέτες, η PHP ετικέτα έχει ένα λατινικό ερωτηματικό (?) μετά την αριστερή γωνιώδη αγκύλη και πριν την δεξιά γωνιώδη αγκύλη. Όλο το κείμενο από έξω από την PHP ετικέτα απλώς περνά στον browser. Το κείμενο μέσα στην ετικέτα είναι PHP κώδικας και αναλύεται.

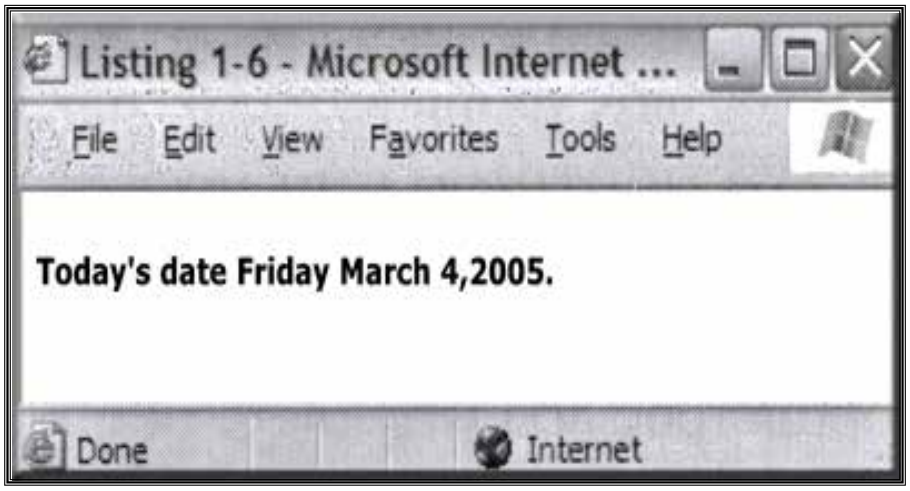
Για να μπορεί η PHP να χειρίζεται την XML και κάποιους άλλους ιδιότροπους επεξεργαστές, όπως το Front Page της Microsoft, προσφέρει τρεις άλλους τρόπους να σημειώνετε τον κώδικα. Το php μετά το αριστερό λατινικό ερωτηματικό κάνει τον κώδικα PHP φιλικό σε αναλυτές XML. Εναλλακτικά, μπορείτε να χρησιμοποιήσετε μια ετικέτα script σαν να γράφετε Javascript. Τέλος μπορείτε να χρησιμοποιήσετε ετικέτες που μοιάζουν με ASP, χρησιμοποιώντας το <% για να ξεκινάτε μπλοκ με κώδικα. Επειδή μπορούμε να διαμορφώσουμε την PHP χρησιμοποιούμε την απλή μέθοδο <? και όταν πρόκειται για κώδικα τον οποίο μοιράζεστε με άλλους είναι καλύτερα να χρησιμοποιήσετε το <? php για αριστερή ετικέτα όπως και στα παραδείγματα.

Το παράδειγμα 1.6 δείχνει μια κανονική HTML σελίδα με μια αξιοσημείωτη διαφορά : τον PHP κώδικα μεταξύ των <?php και the ?>. όταν αυτή η σελίδα αναλύεται από την λειτουργική μονάδα της PHP, θα αντικαταστήσει τον PHP κώδικα με την σημερινή ημερομηνία. Ίσως να διαβάσετε κάτι όπως Monday march 17, 2003.

Κώδικας 2.6
Τυπώνοντας την σημερινή ημερομηνία.

```
<html>

<head>

<title>example 1-6</title>

</head>

<body>
```

```
today’s date : </php print (date (“l f d, y “));?>

</body>

</html>
```



Εικόνα 2.1

Έξοδος από τη λίστα της εικόνας 2.1

Ο κενός χώρος, δηλαδή τα κενά, οι στηλοθέτες και οι αλλαγές γραμμών, αγνοούνται από την PHP. Αν χρησιμοποιηθούν διακριτικά, μπορούν να βελτιώσουν την αναγνωσιμότητα του κώδικα σας. Το παράδειγμα 2.7 είναι λειτουργικά ίδιο με το προηγούμενο παράδειγμα αν και μπορείτε να παρατηρήσετε πιο εύκολα ότι περιέχει κώδικα PHP.

Κώδικας 2.7
Αναμορφοποίηση για λόγους αναγνωσιμότητας.

```
<html>

<head>

<title>example 1-7</title>

</head>

<body>

today;s date :

<?php

/* τυπώνει την σημερινή ημερομηνία */
```

```
print (date (“l f d, y”));

?>

</body>

</html>
```

Ίσως επίσης να παρατηρήσατε την γραμμή του κώδικα του παραδείγματος 1 που ξεκινά από μια κάθετο και ακολουθείται από ένα αστερίσκο. Αυτό είναι σχόλιο και οτιδήποτε ανάμεσα των /* και */ είναι ισοδύναμο με κενά και αγνοείται. Τα σχόλια μπορούν να χρησιμοποιηθούν για να τεκμηριώνετε τον τρόπο που δουλεύει ο κώδικά σας. Ακόμη και αν συντηρείτε μόνοι σας τον κώδικα θα βρείτε τα σχόλια απαραίτητα για όλα τα ascript έκτος από τα πιο απλά. Ακόμη τα σχόλια βοηθάνε ακόμη και κάποιον που δεν ξέρει καλά την γλώσσα να κατανοήσει πως δουλεύει ο κώδικας.

Εκτός από τις εντολές που ανοίγουν και κλείνουν τα σχόλια, η PHP παρέχει δυο τρόπους να δημιουργείτε μια γραμμή σχολίου. Οι διπλές κάθετοι ή το # κάνουν τον αναλυτή να αγνοεί οτιδήποτε υπάρχει ως το τέλος της γραμμής.

Αφού αγνοήσει τα κενά και τα σχόλια του παραδείγματος 2.7 ο PHP αναλυτής συναντά την πρώτη λέξη : print. Αυτή είναι μια από τις συναρτήσεις της PHP. Μια συνάρτηση συγκεντρώνει κώδικα σε μια μονάδα που μπορείτε να καλέσετε με το όνομά της. Η συνάρτηση print στέλνει κείμενο στον browser. Τα περιεχόμενα των παρενθέσεων θα υπολογισθούν και αν παράγουν έξοδο, η print θα την περάσει στον browser.

Που τελειώνει η γραμμή; αντίθετα με την Basic και την Javascript, που χρησιμοποιούν μια αλλαγή γραμμής για να υποδείξουν το τέλος μιας γραμμής, η PHP χρησιμοποιεί ένα ερωτηματικό. Σε αυτό το θέμα η PHP έχει εμπνευσθεί και από την C.

Τα περιεχόμενα της γραμμής μεταξύ των print και ; είναι μια κλήση στη συνάρτηση με το όνομα date. Το κείμενο μεταξύ της αριστερής και δεξιάς παρένθεσης είναι η παράμετρος που περνά στην date. Η παράμετρος τώρα λέει στην date σε ποια μορφή θέλετε να εμφανίζεται η ημερομηνία. Σε αυτή την περίπτωση έχουμε χρησιμοποιήσει του κωδικούς για τα ονόματα των ημερών της εβδομάδας, το πλήρες όνομα του μήνα και το τετραψήφιο έτος. Η σημερινή ημερομηνία μορφοποιείται και περνά ξανά στην συνάρτηση print.

Η συμβολοσειρά των χαρακτήρων που ξεκινά και τελειώνει με διπλά εισαγωγικά ονομάζεται σταθερή συμβολοσειρά ή κυριολεκτική συμβολοσειρά. Η PHP ξέρει ότι όταν τα εισαγωγικά περικλείουν χαρακτήρες, πρέπει να τους αντιμετωπίσει ως κείμενο. Χωρίς τα εισαγωγικά, η PHP θα υπέθετε ότι ονομάζατε μια συνάρτηση ή κάποιο άλλο μέρος της ίδιας της γλώσσας. Με άλλα λόγια , το πρώτο εισαγωγικό λέει στην PHP να «μείνει μακριά», μέχρι να βρει ένα άλλο εισαγωγικό.

Παρατηρήστε ότι η print είναι πλήρως πληκτρολογημένη σε πεζά γράμματα αλλά η date έχει ένα αρχικό κεφαλαίο γράμμα. Το έκανα αυτό για να δείξω ότι η PHP δείχνει επιείκεια στα ονόματα των ενσωματωμένων συναρτήσεών της. Οι Print,PRINT και PrInT είναι όλες έγκυρες κλήσεις στη ίδια συνάρτηση. Ωστόσο, για λόγους αναγνωσιμότητας, συνηθίζεται να γράφονται οι ενσωματωμένες συναρτήσεις της PHP χρησιμοποιώντας μόνο πεζά γράμματα.

2.6 Αποθηκεύοντας Δεδομένα.

Συνήθως είναι απαραίτητο να αποθηκεύεται πληροφορίες για μετέπειτα χρήση. Η PHP , όπως και οι περισσότερες γλώσσες προγραμματισμού, προσφέρουν την ιδέα των μεταβλητών. Οι μεταβλητές δίνουν ένα όνομα στις πληροφορίες που θέλετε να αποθηκεύσετε και να χειριστείτε. Το παράδειγμα 1.8 επεκτείνει το παράδειγμα μας χρησιμοποιώντας μεταβλητές.

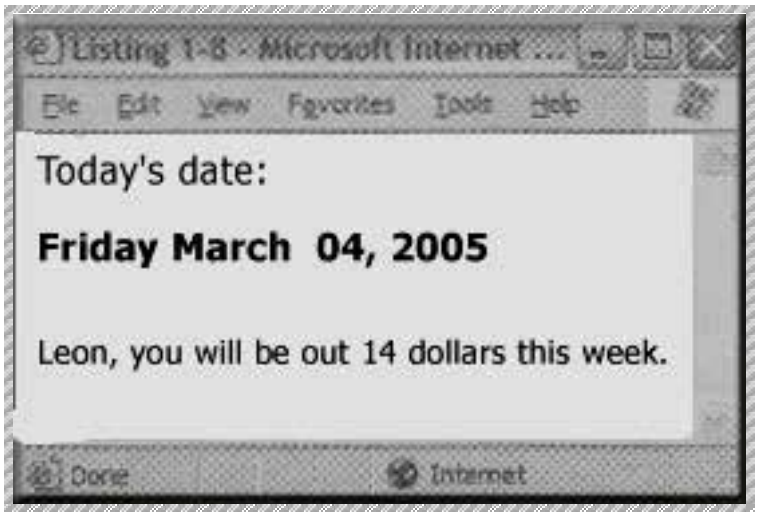
Το πρώτο μπλοκ του PHP κώδικα δίνει τιμές σε μερικές μεταβλητές. Οι τέσσερις μεταβλητές είναι Yourname, CostOfLunch, Today και DaysBuyingLunch. Η PHP ξέρει ότι είναι μεταβλητές γιατί έχουν στην αρχή τους ένα δολάριο ($). Την πρώτη φορά που χρησιμοποιείτε μια μεταβλητή σε ένα PHP script, δεσμεύεται κάποια μνήμη για αποθήκευση των πληροφοριών που θέλετε να αποθηκεύσετε. Δεν χρειάζεται να πείτε στην PHP τι είδους πληροφορίες περιμένετε να αποθηκευθούν στην μεταβλητή. Η PHP μπορεί να το καταλάβει από μόνη της.

Το script πρώτα βάζει ένα χαρακτήρα στη μεταβλητή YourName. Όπως είπα νωρίτερα, η PHP ξέρει ποια δεδομένα είναι κείμενο, επειδή βάζω εισαγωγικά γύρω του. Παρόμοια βάζω την σημερινή ημερομηνία σε μια μεταβλητή που ονομάζεται Today. Σε αυτή την περίπτωση, η PHP ξέρει ότι θα βάλει κείμενο στην μεταβλητή, επειδή η συνάρτηση date επιστρέφει κείμενο. Αυτό το είδος των δεδομένων ονομάζεται συμβολοσειρά, που είναι συντόμευση του πλήρους ονόματος που είναι « συμβολοσειρά χαρακτήρων ». Ένας χαρακτήρας είναι ένα γράμμα, ένας αριθμός, ή κάποιο άλλο σημάδι που εισάγετε πληκτρολογώντας ένα πλήκτρο στο πληκτρολόγιο σας. Παρατηρήστε ότι υπάρχει ένα (=) ίσον που χωρίζει την μεταβλητή από την τιμή που τη δίνετε. Αυτός είναι ο τελεστής εκχώρησης. Οτιδήποτε είναι στα δεξιά του ίσον, μπαίνει στην μεταβλητή που είναι αριστερά του ίσον.

Κώδικας 2.8

Δίνοντας τιμές σε μεταβλητές.

```
<?php
$YourName = “leon”;
$Today = date (“l F d, y”);
$CostOfLunch = 3.50;
$DaysBuyingLunch = 4;
?>
<html>
<head>
<title>example 1.8</title>
</head>
Today;s date :
<?php  /* τυπώνει την σημερινή ημερομηνία*/
print(“<h3>$Today</h3>\n”);
/* τυπώνει μήνυμα για το κόστος του γεύματος */
print (“$YourName, you will be out “);
print($CostOfLunch * $DaysBuyingLunch);
print(“dollars this week.<br>\n”);
?>
</body>
</html>
```



Εικόνα 2.2

Έξοδος από το κώδικα της εικόνας 2.2

Η τρίτη και η τέταρτη εκχώρηση βάζουν αριθμητικά δεδομένα σε μεταβλητές. Η τιμή 3.5 είναι αριθμός κινητής υποδιαστολής, ή κλασματικός αριθμός. Η PHP καλεί αυτό τον τύπο διπλό αριθμό (double), δείχνοντας έτσι την κληρονομικότητάς της από την C. Η τιμή 4 στην επόμενη εκχώρηση είναι ένας ακέραιος ή ένας ολόκληρος αριθμός.

Μετά την εκτύπωση κάποιου HTML κώδικα, ανοίγει ένα άλλο μπλοκ με κώδικα PHP. πρώτα το script τυπώνει την σημερινή ημερομηνία ως επικεφαλίδα τρίτου επιπέδου. Παρατηρήστε ότι το script περνά κάποιες νέες πληροφορίες στην συνάρτηση print και αυτές θα σταλούν στον browser.

Σε σχέση με τις μεταβλητές η PHP δεν δείχνει επιείκεια με τα κεφαλαία-πεζά γράμματα. Το Today και το today είναι δυο διαφορετικές μεταβλητές. Αφού η PHP δεν απαιτεί να δηλώνετε τις μεταβλητές πριν τις χρησιμοποιήσετε, μπορείτε κατά λάθος να πληκτρολογήσετε today όταν εννοείτε το Today και δεν θα δημιουργηθεί λάθος εξ ορισμού. Εν οι μεταβλητές είναι κενές ενώ έπρεπε να έχουν τιμή, ελέγξτε τα κεφαλαία πεζά. Μπορείτε επίσης να βρείτε τέτοια λάθη διαμορφώνοντας την PHP έτσι ώστε να σας προειδοποιεί για μη αρχικοποιημένες μεταβλητές.

Το script μετά τυπώνει το leon, you will be out 14 dollars this week. Η γραμμή που τυπώνει το σύνολο θα πρέπει να το υπολογίσει με πολλαπλασιασμό χρησιμοποιώντας τον τελεστή *.

2.7 Λαμβάνοντας Την Είσοδο Του Χρήστη.

Ο χειρισμός μεταβλητών που μπορείτε να ορίσετε μέσα στο script σας είναι κάπως ενδιαφέρων, αλλά όχι κάτι το εντυπωσιακό. Τα script γίνονται πιο χρήσιμα όταν χρησιμοποιούν είσοδο από τον χρήστη. Όταν καλείται την PHP από μια HTML φόρμα, τα πεδία της φόρμας μετατρέπονται σε μεταβλητές. Το παράδειγμα 2.9 είναι μια φόρμα που καλεί το παράδειγμα 2.10, μια ακόμα τροποποίηση του script του παραδείγματός μας.

Κώδικας 2.9
HTML φόρμα πληροφορίας γεύματος.

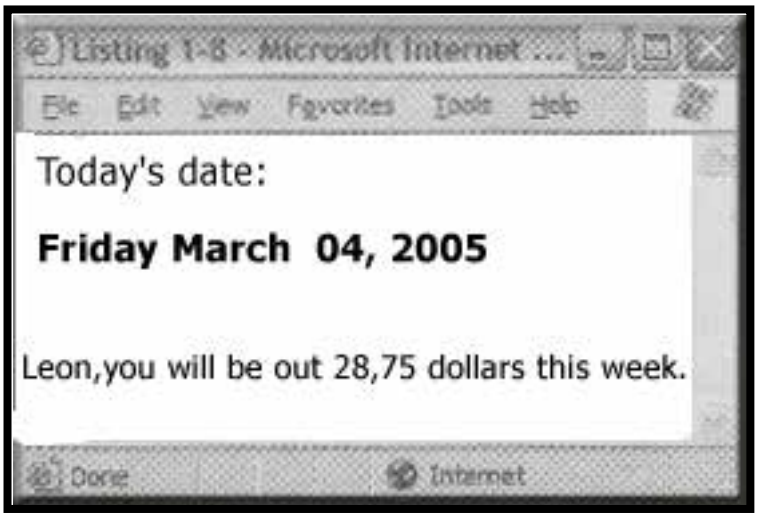
```
<html>

<head>

<title> example 1-9</title>

</head>

<body>

<form action=”1-10.php” method=“post”>

Your name :

<input type=”text” name=”yourName”><br>

costoflunch :

<input type=”text” name=”Costoflunch”<br>

days buying lunch :

<input type=”text” name=”Daysbuyinglunch”><br>

<input type=”submit” value=”Compute”>

</form>

</body>

</html>
```

Το παράδειγμα 2.9 είναι μια τυπική HTML φόρμα. Αν έχετε χρησιμοποιήσει CGI, θα σας φανεί οικεία. Υπάρχουν τρία πεδία της φόρμας που ταιριάζουν με τις μεταβλητές του προηγούμενου παραδείγματός μας. Αντί απλώς να βάζουμε δεδομένα στις μεταβλητές, παρέχουμε μια φόρμα και χρησιμοποιούμε τις πληροφορίες που πληκτρολογεί ο χρήστης. Όταν ο χρήστης πατήσει το κουμπί submit, το script που αναφέρεται στην ιδιότητα Action θα λάβει τα τρία πεδία της φόρμας και η PHP θα τα μετατρέψει σε μεταβλητές.

Κώδικας 2.10
Υπολογίζοντας το κόστος ενός γεύματος από μια φόρμα.

```
<?php

$Today = date (“l F d, y”);

?>

<html>

<head>

<title>example</title>

</head>

<body>

today’s date :

<?php

/* τυπώνει την ημερομηνία */

print (“<h3>$Today</h3>\n”);

/* τυπώνει μήνυμα για το κόστος του γεύματος */

print ($_Request[ ‘yourname’] . “, you will be out”);

print ($_Request[costoflunch’] *

$_Request[‘daysbuyinglunch’]);

print(“dollars this week.<br>\n”);

?>
```

```
</body>

</html>
```



Εικόνα 2.3

Έξοδος από το κώδικα 1.10

Παρατηρήστε ότι στο πρώτο τμήμα του PHP script, έχω διαγράψει τις γραμμές που ορίζουν τις μεταβλητές, εκτός από τη σημερινή ημερομηνία. Είδατε ότι, αντί να χρησιμοποιήσω το $CostOfLunch, χρησιμοποίησα το $_Request['CostOfLunch'];. Η PHP συλλέγει όλες τις μεταβλητές που στέλνονται από φόρμες και cookies σε μια συλλογή που ονομάζεται _Request. Το τεχνικό όνομα για αυτό το είδος των δεδομένων είναι πίνακας (array).

Προσπαθήστε να πειραματιστείτε με τα script δίνοντας άσχετες πληροφορίες στα πεδία της φόρμας. Κάτι που θα πρέπει να παρατηρήστε είναι ότι αν βάλετε λέξεις εκεί που το script περιμένει αριθμό, η PHP φαίνεται ότι τους δίνει μηδενικές τιμές. Οι μεταβλητές ορίζονται με μια συμβολοσειρά κειμένου και όταν το script προσπαθεί να τις χειριστεί σαν αριθμό, η PHP προσπαθεί να μετατρέψει τις πληροφορίες. Δίνοντας το 10 Little Indians για κόστος του γεύματος, αυτό θα μεταφρασθεί ως 10.

2.8 Επιλέγοντας Μεταξύ Εναλλακτηκών Επιλογών.

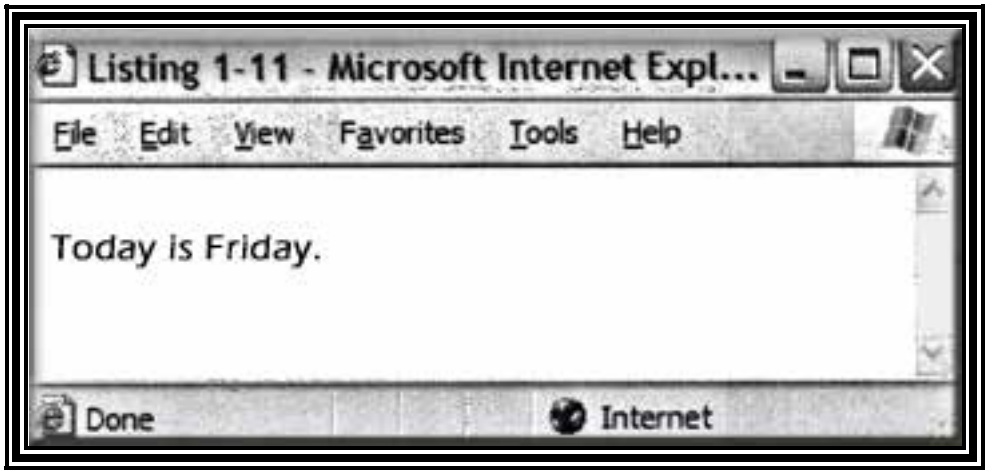
Η PHP σας επιτρέπει να ελέγχετε συνθήκες και να εκτελείται κάποιο κώδικα, ανάλογα με τα αποτελέσματα του ελέγχου. Η απλούστερη μορφή είναι η

εντολή if. Το παράδειγμα 2.11 δείχνει πως μπορείτε να προσαρμόσετε τα περιεχόμενα μιας σελίδας ανάλογα με την τιμή της μεταβλητής.

Η μεταβλητή Today ορίζεται με το όνομα της σημερινής ημέρας της εβδομάδας. Η εντολή if αξιολογεί την παράσταση μέσα στις παρενθέσεις ως true ή false. Ο τελεστής = συγκρίνει την αριστερή πλευρά με την δεξιά πλευρά. Αν το Today περιέχει την λέξη Friday, εκτελείται το μπλοκ του κώδικα που περικλείεται σε άγκιστρα ({ και}). Σε άλλες περιπτώσεις, εκτελείται το μπλοκ του κώδικα που σχετίζεται με την εντολή else.

Κώδικας 2.11

```
<html>

<head>

<title>example1-11</title>

</head>

<body>

<h1>

<?php

/* λαμβάνει την σημερινή ημερομηνία*/

$Today = date(“l”);

if ($Today == “Friday”)

{      print (“thank goodness it’s Friday!”);

}

else

{    print (“today is$Today.”);

}

?>

</h1>

</body>

</html>
```



Εικόνα 2.4

Έξοδος του κώδικα 2.11

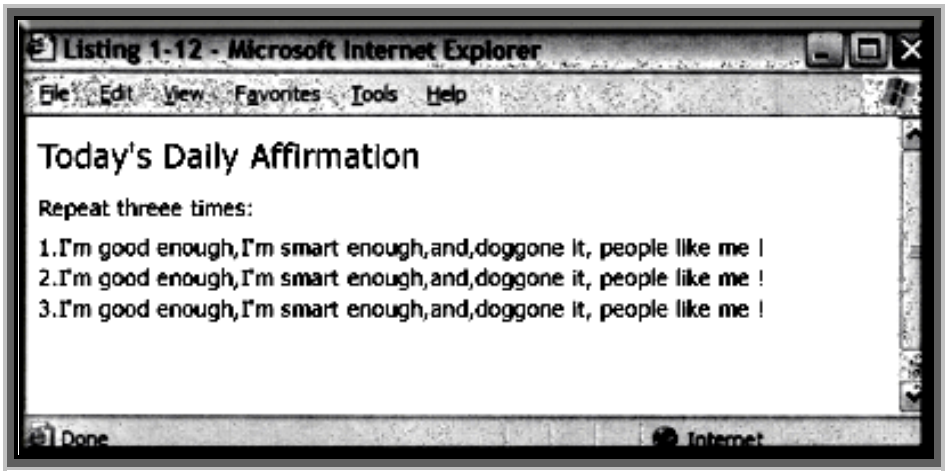
2.9 Επαναλαμβανόμενος Κώδικας.

Ο τελευταίος τύπος λειτουργικότητας σε αυτή την σύντομη εισαγωγή είναι ο βρόχος. Οι βρόχοι σας επιτρέπουν να επαναλαμβάνετε την εκτέλεση του κώδικα. Το παράδειγμα 2.12 είναι ένα παράδειγμα ενός βρόχου for. Η εντολή for περιμένει τρεις παραμέτρους, χωρισμένες με ερωτηματικά. Η πρώτη παράμετρος εκτελείται μια φορά πριν ξεκινήσει ο βρόχος. Συνήθως αρχικοποιεί μια μεταβλητή. Η δεύτερη παράμετρος κάνει ένα έλεγχο. Αυτός είναι συνήθως ένας έλεγχος ως προς την μεταβλητή που είναι η πρώτη παράμετρος. Η Τρίτη παράμετρος εκτελείται κάθε φορά που φθάνει το τέλος του βρόχου.

Κώδικας 2.12
Δήλωση της σημερινής ημέρας.

```
<html>

<head>

<title>example 1-12</title>

</head>

<body>
```

```
<h1>today’s daily affirmation</h1>

repeat three times : <br>

<?php

for($count = 1;  $count <=3; $count++)

{

print (“<b>$count</b> I’m good enough,”);

print (“I’m smart enough,”);

print (“and, doggone it, people like me!<br>\n”);

}

?>

</h1>

</body>

</html>
```



Εικόνα 2.5

Έξοδος από το κώδικα 2.12

Ο βρόχος for στο παράδειγμα 2.12 θα εκτελεσθεί τρεις φορές. Η αρχικοποίηση του κώδικα ορίζει την μεταβλητή count σε ένα. Μετά ο κώδικας

συγκρίνει την τιμή του count με το τρία. Αφού το ένα είναι μικρότερο ή ίσο με τρία, εκτελείται ο κώδικας μέσα σε βρόχο. Παρατηρήστε ότι το script τυπώνει την τιμή του count. Όταν τρέξετε αυτό το script, θα δείτε ότι το count από ένα έως τρία. Ο λόγος είναι ότι το τρίτο μέρος της εντολής for προσθέτει ένα στο count κάθε φορά που περνά από τον βρόχο. Ο τελεστής ++ αυξάνει την μεταβλητή αμέσως στα αριστερά του.

Την πρώτη φορά που το count περνά από τον βρόχο είναι 1 και όχι 2. αυτό γίνεται επειδή η αύξηση του count συμβαίνει όταν φθάσουμε στο δεξιό άγκιστρο. Μετά την τρίτη φορά που περνά το count από το βρόχο, θα αυξηθεί σε τέσσερα, αλλά σε αυτό το σημείο τα τέσσερα δεν θα είναι μικρότερο ή ίσο με το τρία, οπότε ο βρόχος θα τερματίσει. Η εκτέλεση συνεχίζεται στην εντολή που ακολουθεί το μπλοκ με τον κώδικα του βρόχου.

v **ΚΕΦΑΛΑΙΟ 3°**

Σε αυτό το κεφάλαιο θα αναφερθούμε με περισσότερα παραδείγματα σε διάφορα παραδείγματα ασφαλούς πιστοποίησης των χρηστών μέσω Scripts της Php.Θα αναλύσουμε τον τρόπο λειτουργίας κώδικα ή διάφορων Scripts της Php.

Όπως είδαμε στο πρώτο κεφάλαιο οι κωδικοί πρόσβασης μπορούν να αποθηκευτούν στο ίδιο το Script αλλά υπάρχουν και άλλοι τρόποι που μπορούμε να πετύχουμε ανάλογα αποτελέσματα. Μάλιστα υπάρχουν άλλοι τρόποι που μας προσφέρουν περισσότερη ασφάλεια.

Οι κωδικοί πρόσβασης και τα ονόματα των εκάστοτε χρηστών μπορούν να αποθηκευτούν σε μια Βάση Δεδομένων (Data Base).Μια τέτοια βάση για παράδειγμα που η Php μπορεί να συνεργαστεί και να έχουμε και τα ανάλογα αποτελέσματα είναι η MySql.Με τον τρόπο αυτό έχουμε πολλά πλεονεκτήματα.

3.1 Αποθηκεύοντας Κωδικούς Πρόσβασης.

Υπάρχουν πολλά καλύτερα μέρη να αποθηκεύετε ονόματα χρηστών και κωδικούς πρόσβασης από το ίδιο το script. Μέσα στο script, είναι δύσκολο να τροποποιήστε τα δεδομένα. Είναι δυνατόν, αλλά είναι κακή ιδέα να γράψετε ένα script που να τροποποιεί τον εαυτό του. Θα σήμαινε να έχετε ένα script στον διακομιστή σας, που εκτελείται στον διακομιστή σας, αλλά γράφεται ή τροποποιείται από άλλους. Η αποθήκευση των δεδομένων σε ένα άλλο αρχείο του διακομιστή θα σας επιτρέψει να γράψετε πιο εύκολα ένα πρόγραμμα για να προσθέτετε και να αφαιρείτε χρήστες και να αλλάζετε τους κωδικούς πρόσβασης.

Μέσα σε ένα script ή σε άλλα δεδομένα, υπάρχει ένα όριο στον αριθμό των χρηστών που μπορείτε να έχετε, χωρίς να επηρεάσετε σοβαρά την ταχύτητα του script. Αν σκέφτεστε να αποθηκεύσετε και να κάνετε αναζητήσεις σε ένα μεγάλο αριθμό στοιχείων ενός αρχείου, θα πρέπει να σκεφτείτε να χρησιμοποιήσετε μια βάση δεδομένων, όπως συζητήθηκε νωρίτερα. Ως απλό κανόνα, αν θέλετε να αποθηκεύσετε και να ψάξετε μια λίστα από περισσότερα από 100 στοιχεία, θα πρέπει να είναι σε μια βάση δεδομένων και όχι σε ένα απλό αρχείο.

Η χρήση μιας βάσης δεδομένων για την αποθήκευση ονομάτων χρηστών και κωδικών πρόσβασης δεν θα κάνει το script πολύ πιο πολύπλοκο, αλλά θα σας επιτρέψει να πιστοποιείτε πολλούς διαφορετικούς χρήστες, γρήγορα. Θα σας επιτρέψει επίσης να γράψετε εύκολα ένα script για να προσθέτετε νέους χρήστες, να διαγράφετε χρήστες και να επιτρέψετε σε χρήστες να αλλάζουν το κωδικό πρόσβασης τους.

Ένα script που να πιστοποιεί τους επισκέπτες μιας σελίδας από μια βάση δεδομένων, δίνεται παρακάτω.

Κώδικας 3.1
secretdb.php

Έχουμε Χρησιμοποιήσει MySQL για να Βελτιώσουμε τον Απλό Μηχανισμό μας Ελέγχου Ταυτότητας.

```
if(!isset($name)&&!isset($password))
 {
Ι/ο επισκέπτης πρέπει να δώσει όνομα και  κωδικό πρόσβασης
<h1>Please Log
This page is secret.
<form method = post action =  "secretdb.php">
<table border = 1>
```

```
<tr>
<th> User-name </th>
<td> <input type = text name = name> </td>
 </tr>
 <tr>
<th> Password </th>
<td> <input type = password name = password>
 </td>
 </tr>
<tr>
<td colspan =2 align = center>
<input type = submit value = "Log In">
</td>
 </tr>
</table>
</form>
<?
{
else
}
// σύνδεση στην mysql
$mysql = mysql_connect( 'localhost', 'webauth', 'webauth1 );
if ( !$mysql)
{
echo 'Cannot connect to database.'; exit;
}
// επιλογή της κατάλληλης βάσης δεδομένων
$mysql = mysql_select_db( 'auth' );
if ( !$mysql)
{
echo 'Cannot select database.';
exit;
}
// ερώτηση στην βάση δεδομένων αν υπάρχει αντίστοιχη εγγραφή


Squery = "select count(*) from auth where
name = '$name' and
pass = 'Spassword' " ;
$result = mysql_query( $query );
if (!$result)
{
echo 'Cannot run query.';
exit;
}
$count = mysql_result( Sresult, 0, 0 );
if ( $count > 0 )
{

//ο συνδυασμός ονόματος και κωδικού πρόσβασης είναι σωστός
echo "<h1>Here it is!</h1>";
echo "I bet you are glad you can see this secret page.";
```

```
}
else
{

// ο συνδυασμός ονόματος και  κωδικού πρόσβασης δεν είναι  σωστός
echo  "<h1>Go Away!</h1>";
echo  "You are not authorized to view this resource.";
}
}
?>
```

Το Script (Κώδικας 3.1) πιστοποιεί τους επισκέπτες μιας σελίδας χρησιμοποιώντας μια Βάση Δεδομένων. Με τον τρόπο αυτό βελτιώνουμε τον απλό Μηχανισμό Ελέγχου Ταυτότητας.

3.2 Χρησιμοποιώντας Βασικό Έλεγχο Ταυτότητας στην PHP.

Τα script της PHP ισχύουν γενικά για όλες τις πλατφόρμες, αλλά η χρήση βασικού ελέγχου ταυτότητας βασίζεται σε μεταβλητές του επιβάλλοντος, που έχουν ορισθεί στον διακομιστή. Για να μπορεί ένα script ελέγχου ταυτότητας HTTP να τρέξει στον Apache χρησιμοποιώντας PHP ως μια Apache λειτουργική μονάδα ή στον 1IS χρησιμοποιώντας PHP ως μια ISAPI λειτουργική μονάδα, πρέπει να εντοπίσετε τον τύπο του διακομιστή και να συμπεριφερθείτε ανάλογα. Το script 3.2 τρέχει και στους δύο διακομιστές.

Κώδικας 3.2
http.php

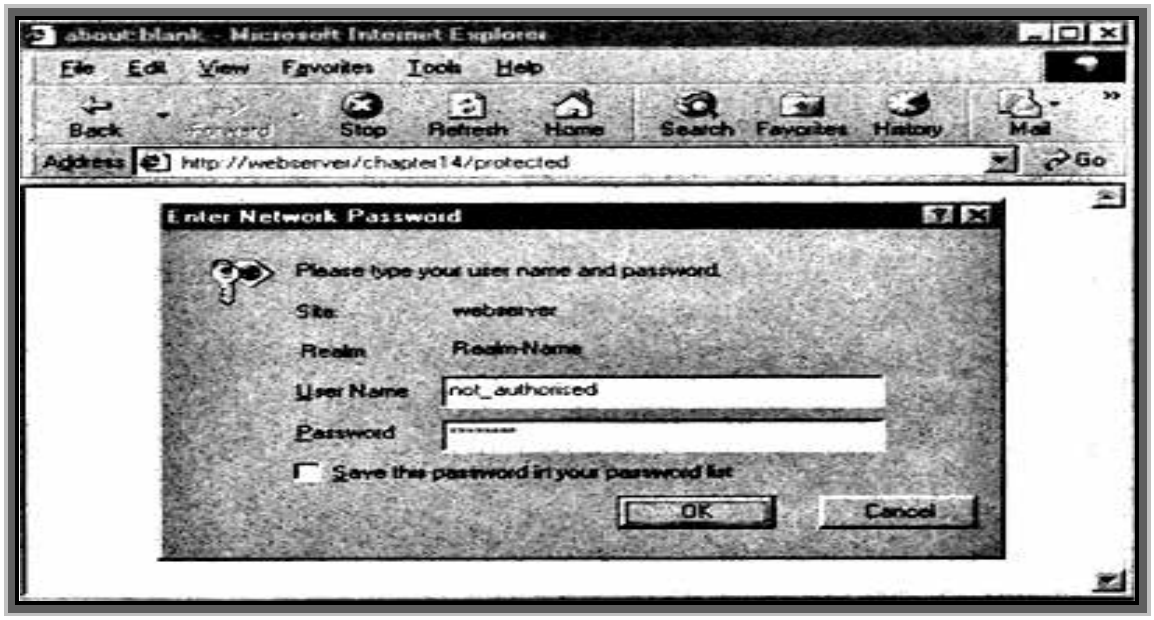
Η PHP Μπορεί να Ξεκινήσει Βασικό Έλεγχο Ταυτότητας http.

```
<?

//αν χρησιμοποιούμε IIS, πρέπει να ορίσουμε $PHP_AUTH_USER και
$PHP_AUTH_PW

 if (substr($SERVER_SOFTWARE, 0, 9) == "Microsoft" &&
!iSset($PHP_AUTH_USER) &&
!isset($PHP_AUTH_PW) &&
substr($HTTP_AUTHORIZATION,   0,   6)   ==  "Basic  "
)
{
```

```
list($PHP_AUTH_USER,   $PHP_AUTH_PW)   =
explode(":",   base64_decode(substr($HTTP_AUTHORIZATION,   6)));
}
// αντικαταστήσετε αυτή την εντολή if με ένα ερώτημα βάσης
δεδομένων ή κάτι παρόμοιο if($PHP_AUTH_USER!="user"||
$PHP_AUTH_PW!="pass")
{

// ο επισκέπτης δεν έχει δώσει ακόμα πληροφορίες, ή ο
// ο συνδυασμός ονόματος και κωδικού πρόσβασης δεν είναι σωστός

header('WWW-Authenticate: Basic realm="Realm-Name"');
 if (substr($SERVER_SOFTWARE, 0, 9) == "Microsoft")
header("Status: 401 Unauthorized");

else
header("HTTP/1.0 401 Unauthorized");
echo "<h1>Go Away!</h1>";
echo "You are not authorized to view this resource.";
{
else
}
// ο επισκέπτης έχει δώσει σωστές πληροφορίες
echo "<h1>Here it is!</h1>";
echo "<p>l bet you are glad you can see this secret page.";
     }
     ?>
```

Ο κώδικας 3.2 ενεργεί με ένα πολύ παρόμοιο τρόπο με τη προηγούμενη λίστα. Αν ο χρήστης δεν έχει δώσει ακόμα πληροφορίες ελέγχου ταυτότητας, θα του ζητηθούν. Αν έχει δώσει λανθασμένες πληροφορίες, παίρνει ένα μήνυμα απόρριψης. Αν παρέχει ένα σωστό ζευγάρι ονόματος - κωδικού πρόσβασης, του παρουσιάζονται τα περιεχόμενα της σελίδας.

Ο χρήστης θα δει ένα περιβάλλον λίγο διαφορετικό από τις προηγούμενες λίστες. Δεν παρέχει μια HTML φόρμα πληροφορίες σύνδεσης. Ο browser του χρήστη θα του εμφανίσει ένα παράθυρο διαλόγου. Μερικά άτομα το βλέπουν αυτό σαν βελτίωση και άλλοι προτιμούν να έχουν πλήρη έλεγχο πάνω στο οπτικό μέρος του περιβάλλοντος. Το παράθυρο διαλόγου σύνδεσης που παρέχει ο Internet Explorer, φαίνεται στην Εικόνα 3.1.



Εικόνα 3.1

Ο browser του χρήστη είναι υπεύθυνος για την εμφάνιση του παραθύρου διαλόγου όταν χρησιμοποιείται HTTP έλεγχος ταυτότητας.

Επειδή ο έλεγχος ταυτότητας βοηθειέται από λειτουργίες που είναι ενσωματωμένες στο browser, οι browser επιλέγουν να είναι λίγο διακριτικοί στο τρόπο που γίνεται ο χειρισμός των αποτυχημένων προσπαθειών ελέγχου ταυτότητας. Ο Internet Explorer επιτρέπει στον χρήστη να δοκιμάσει να πιστοποιηθεί τρεις φορές, πριν εμφανίσει την σελίδα απόρριψης. Ο Netscape Navigator θα επιτρέψει στον χρήστη να δοκιμάσει απεριόριστο αριθμό φορών, εμφανίζοντας μεταξύ των προσπαθειών το παράθυρο διαλόγου "Authorization failed. Retry?" (ο έλεγχος ταυτότητας απέτυχε - θα ξαναπροσπαθήσετε;). Ο Netscape εμφανίζει τη σελίδα απόρριψης μόνο αν ο χρήστης κάνει κλικ στο Cancel.

Όπως και με τον κώδικα 3.1 και 3.2, θα μπορούσαμε να συμπεριλάβουμε αυτόν τον κώδικα σε σελίδες που θέλαμε να προστατεύσουμε ή να τον προσαρτούμε αυτόματα σε κάθε αρχείο ενός καταλόγου.

3.3 Κρυπτογράφηση Κωδικών Πρόσβασης.

Ανεξάρτητα από το αν αποθηκεύουμε τα δεδομένα μέσα σε μια βάση δεδομένων ή σε ένα αρχείο, δεν πρέπει να αποθηκεύουμε τους κωδικούς πρόσβασης ως απλό κείμενο. Ένας αλγόριθμος hash μιας κατεύθυνσης, μπορεί να παρέχει λίγη περισσότερη ασφάλεια, με λίγη επιπλέον προσπάθεια.

Η PHP συνάρτηση crypt () παρέχει μια hash συνάρτηση κρυπτογράφησης, μιας κατεύθυνσης. Το πρωτότυπο της συνάρτησης είναι

```
string crypt   (string str  [,   string salt])
```

Αν δοθεί η συμβολοσειρά str, η συνάρτηση θα επιστρέψει μια ψευδο-τυχαία συμβολοσειρά. Για παράδειγμα, αν δοθεί η συμβολοσειρά "pass", το salt "xx", crypt () επιστρέφει "xxkTImYjlikoII". Αυτή η συμβολοσειρά δεν μπορεί να αποκρυπτογραφηθεί και μετατρέπεται ξανά σε "pass", ακόμα και από τον δημιουργό της, καθώς δεν φαίνεται πολύ χρήσιμη με την πρώτη ματιά. Η ιδιότητα που κάνει την crypt ( ) χρήσιμη είναι ότι η έξοδος της είναι προσδιορίσιμη. Αν δοθεί η ίδια συμβολοσειρά και το salt, η crypto θα επιστρέψει το ίδιο αποτέλεσμα κάθε φορά που τρέχει.

Αντί να έχετε PHP κώδικα όπως:

```
if(  $username ==  "user"  && $password ==  "pass"   )
{
//OK,   οι  κωδικοί  πρόσβασης ταιριάζουν
μπορούμε να έχουμε κώδικα όπως
if(  $username ==   'user'   && crypt(Spassword,'xx')  ==
'xxkTImYjlikoII'   )
{
//OK,  οι  κωδικοί πρόσβασης ταιριάζουν
}
```

Δεν χρειάζεται να ξέρουμε πώς ήταν το 'xxkTImYjlikoir' πριν χρησιμοποιήσουμε την crypt () σε αυτό. Πρέπει μόνο να ξέρουμε αν ο κωδικός πρόσβασης που πληκτρολογήθηκε είναι ο ίδιος με αυτόν που αρχικά έτρεξε η crypt ().

Όπως αναφέρθηκε ήδη, είναι κακή ιδέα να γράφετε μέσα στον κώδικα τα ονόματα χρηστών και τους κωδικούς πρόσβασης. Θα πρέπει να χρησιμοποιήσετε ένα ξεχωριστό αρχείο ή βάση δεδομένων για να τα αποθηκεύσετε.

Αν έχουμε μια MySQL βάση δεδομένων για να αποθηκεύουμε τα δεδομένα πιστοποίησης, θα μπορούσαμε να χρησιμοποιήσουμε την PHP συνάρτηση crypt() ή την MySQL συνάρτηση PASSWORD (). Αυτές οι συναρτήσεις δεν παράγουν την

ίδια έξοδο, αλλά εξυπηρετούν τον ίδιο σκοπό. Η crypt () και η PASSWORD () παίρνουν μια συμβολοσειρά και εφαρμόζουν ένα hash αλγόριθμο, μη αναστρέψιμο.

Για να χρησιμοποιήσουμε την PASSWORD (), θα μπορούσαμε να ξαναγράψουμε το SQL ερώτημα του Κώδικα 3.1

```
select count(*)  from auth where
name =   '$name'
and pass = password('$password')
```

Αυτό το ερώτημα θα μετρήσει τον αριθμό των γραμμών του πίνακα auth, που έχουν τιμή ονόματος ίση με τα περιεχόμενα του $ name και μια τιμή pass ίση με την έξοδο που δίνεται από την PASSWORD () όταν εφαρμοστεί στα περιεχόμενα του $password. Υποθέτοντας ότι εξαναγκάζουμε τους χρήστες να έχουν μοναδικά ονόματα χρηστών, το αποτέλεσμα αυτού του ερωτήματος θα είναι Ο ή 1.

3.3.1 PGP Και GPG

Όπως αναφέραμε και παραπάνω μπορούμε να χρησιμοποιήσουμε κρυπτογράφηση για να έχουμε πιο ασφαλείς συναλλαγές και πιο ασφαλή πρόσβαση σε διάφορες εφαρμογές του Web.

Μπορούμε να χρησιμοποιήσουμε πολλά συστήματα κρυπτογράφησης για να το πετύχουμε αυτό. Δύο από τα πιο διαδεδομένα είναι το **PGP και GPG.** Το GPG το χρησιμοποιούμε σε εφαρμογές LINUX.Το PGP παρέχει βασική τυποποίηση για να έχουμε κρυπτογραφημένο Ηλεκτρονικό Ταχυδρομείο .Το PGP δεν είναι ένα δωρεάν πρόγραμμα.

Η δωρεάν έκδοση μπορεί να χρησιμοποιηθεί για μη επαγγελματική χρήση.

Τώρα, αφού εγκαταστήσουμε το GPG στον Linux σταθμό μας μπορούμε να δούμε πώς στέλνουμε ένα κρυπτογραφημένο Ηλεκτρονικό ταχυδρομείο.

Θα το δοκιμάσουμε δημιουργώντας ένα αρχείο που περιέχει κάποιο κείμενο και αποθηκεύοντας το ως text. txt. πληκτρολογώντας την παρακάτω εντολή:

```
gpg -a –recipient 'Luke Welling <luke@tangledweb.com.au>' –encrypt test.txt
```
(τροποποιήστε την για να χρησιμοποιήσετε το όνομα του κλειδιού σας), θα πρέπει να πάρετε την προειδοποίηση
```
gpg: Warning: using insecure memory!
```
και δημιουργήστε ένα αρχείο που ονομάζεται text.txt.asc. Αν ανοίξετε το text.txt.asc, θα πρέπει να δείτε ένα κρυπτογραφημένο μήνυμα, όπως το εξής:
```
-----BEGIN PGP MESSAGE-----
Version: GnuPG v1.0.3 (GNU/Linux)
Comment: For info see http://www.gnupg.org
hQEOAODU7hVGgdtnEAQAhr4HgR7xpIBsK9CiELQw85+k1QdQ+p/FzqL8tICrQ+B3
OGJTEehPUDErwqUw/uQLTdsOr1oPSrIAZ7c6GVkhOYEVBj2MskT81IIBvdo950yH
K9PUCvg/rLxJ1kxe4Vp8QFET5E3FdII/ly8VP5gSTE7gAgmOSbFf3S91PqwMyTkD
/2oJEvL6e3cP384sOi81rBbDbOUAAhCj jXt2DX/uX9q6P18QW56UICUOn4DPaW1G
/gnNZCkcVDgLcKfBjbkB/TCWWhpA7o7kX4CIcIh7KHMHY4RKdnCWQf271oE+8i9
cJRSCMsFIoI6MMNRCQHY6p9bfxL2uE39IRJrQbe6xoEeOnkBOuTYxiLOTG+FrNrE
tvBVMSOnsHu7HJey+oY4Z833pk5+MeVwYumJwlvHjdZxZmV6wz46G02XGT17b28V
wSBnWOoBHSZsPvkQXHTOq65EixP8y+YJvBN3z4pzdHOXa+NpqbH7q3+xXmd30hDR
+u7t6MxTLDbgC+NR
=gfQu
```

```
-----END PGP MESSAGE-----
```

Θα πρέπει να μπορείτε να μεταφέρετε αυτό το αρχείο στο σύστημα όπου δημιουργήσατε αρχικά το κλειδί και να τρέξετε:

```
gpg -d test.txt.asc
```

για να δείτε το αρχικό σας κείμενο ξανά.

Για να τοποθετήσετε το κείμενο σε ένα αρχείο, αντί να το εξάγετε στην οθόνη, μπορείτε να χρησιμοποιήσετε τη σήμανση -o και να καθορίσετε ένα αρχείο εξόδου, όπως το εξής:

```
gpg  -do test.out test.txt.asc
```

Αν έχετε διαμορφωμένο το GPG ώστε ο χρήστης του PHP script σας να το τρέχει από τη γραμμή εντολών, έχετε κάνει τον περισσότερο δρόμο. Αν κάτι δεν πάει καλά, δείτε το διαχειριστή του συστήματος ή την τεκμηρίωση του GPG.

Ο Κώδικας 3.3 και 3.4 επιτρέπουν σε άτομα να στέλνουν κρυπτογραφημένο ηλεκτρονικό ταχυδρομείο χρησιμοποιώντας PHP που καλεί GPG.

Κώδικας 3.3
privatejmail.php

Η HTML φόρμα μας για να στέλνουμε κρυπτογραφημένο ηλεκτρονικό ταχυδρομείο.

```
<html>
<body>
<h1>Send Me Private Mail</h1>

<?
// ίσως να πρέπει να αλλάξετε αυτή την γραμμή, αν δεν την
χρησιμοποιείτε
// οι προεπιλεγμένες θύρες, 80 για κανονική κίνηση και 443 για SSL
if ($HTTP_SERVER_VARS[ "SERVER_PORT" ] !=443)
 echo "<p><font color = red>
WARNING: you have not connected to this page using SSL.
Your message could be read by others. </font></p>" ;
?>

<form method = post action = send_private_mail.php><br>
Your email address :<br>
<input type = text name = from size = 38><br>
Subject :<br>
<input type = text name = title size = 38><br>
Your message :<br>
<textarea name = body cols = 30 rows = 10>
</textarea><br>
<input type = submit value = "Send!">
</form>
</body>
</html>
```

Κώδικας 3.4
send_private_mail.php

Το PHP Script μας για να καλούμε GPG και να στέλνουμε Κρυπτογραφημένο Ηλεκτρονικό Ταχυδρομείο.

```
$to_email =  "luke@localhost";

//  Πείτε στο gpg πού να βρει το δακτύλιο των κλειδιών
```

```
//  Σε αυτό το σύστημα,  ο αρχικός κατάλογος του χρήστη nobody
είναι ο /tmp/putenv("GNUPGHOME=/tmp/.gnupg");

//δημιουργία μοναδικού ονόματος αρχείου
 $infile = tempnam("", "pgp");
$outfile = $inf ile. " .asc" ;

//γράψιμο του κειμένου του χρήστη στο αρχείο
 $fp = fopen ($inf ile, "w");
fwrite($fp, $body);
 fclose($f p) ;

//διαμόρφωση της εντολής μας
Scommand = "/usr/local/bin/gpg -a \ \
— recipient 'Luke Welling <lukePtangledweb.com.au>[1] \ \
   —  encrypt -o Soutfile $infile";

// εκτέλεση της gpg εντολής μας
system($command, Sresult);

//διαγραφή μη κρυπτογραφημένου προσωρινού αρχείου
unlink($infile) ;

if ($result==0)
{
$fp = fopen (Soutfile, "r");
 if (!$fp| |filesize ($outf ile)==0)
{
$result = -1 ;
}
 else
{
//διάβασμα του κρυπτογραφημένου αρχείου
Scontents = fread ($fp, filesize (Soutfile));
 // διαγραφή κρυπτογραφημένου προσωρινού unlink($outf ile) ;
mail($to_email, $title, $contents, "From: $from\ n");
 echo "<h1>Message Sent</h1>
<p>Your message was encrypted and sent.
<p>Thank you. " ;
}
}
if ($result!=0)
}
echo "<h1>Error:</h1>

<p>Your message could not be encrypted,so has not been sent.
<p>Sorry.";
}
?>
```

Για να δουλέψει αυτός ο κώδικας, θα πρέπει να αλλάξετε μερικά πράγματα. Το ηλεκτρονικό ταχυδρομείο θα σταλεί στην διεύθυνση στο $ to_email.

Η γραμμή putenv("GNUPGHOME=/tmp/.gnupg");

Θα πρέπει να αλλάξει ανάλογα με τη θέση του δακτυλίου κλειδιών GPG. Στο σύστημα μας, ο Web διακομιστής τρέχει ως χρήστης nobody και έχει αρχικό κατάλογο /tmp/.

Χρησιμοποιούμε τη συνάρτηση tempnam() για να δημιουργήσουμε ένα μοναδικό, προσωρινό όνομα αρχείου. Μπορείτε να καθορίσετε τον κατάλογο και ένα πρόθεμα ονόματος αρχείου. Πρόκειται να δημιουργήσουμε και να διαγράψουμε αυτά τα αρχεία σε ένα δευτερόλεπτο, έτσι δεν είναι σημαντικό το πώς θα το ονομάσουμε. Καθορίζουμε ένα πρόθεμα 'gpg', αλλά επιτρέπουμε στην PHP να χρησιμοποιήσει τον προσωρινό κατάλογο του συστήματος.

® Η εντολή:

```
$command = "/usr/local/bin/gpg -a ".
"—recipient 'Luke Welling <luke@tangledweb.com.au>' ". "—encrypt -o
$outfile Sinfile";
```

Διαμορφώνει την εντολή και τις παραμέτρους που θα χρησιμοποιηθούν για να κληθεί το gpg. Θα πρέπει να τροποποιηθεί ανάλογα με τις ανάγκες σας. Όπως και όταν το χρησιμοποιήσαμε στην γραμμή εντολών, θα πρέπει να πείτε στο GPG ποιο κλειδί να χρησιμοποιήσει για να κρυπτογραφήσει το μήνυμα.

® Η εντολή:

```
system($command, Sresult);
```

εκτελεί τις εντολές που αποθηκεύονται στο $command και αποθηκεύει την τιμή επιστροφής στο $result.

Θα μπορούσαμε να αγνοήσουμε την τιμή επιστροφής, αλλά μας επιτρέπει να έχουμε μια εντολή if και να πούμε στον χρήστη ότι κάτι πήγε λάθος.

Όταν τελειώσουμε με τα προσωρινά αρχεία που χρησιμοποιούμε, τα διαγράφουμε χρησιμοποιώντας τη συνάρτηση unlink(). Αυτό σημαίνει ότι το μη κρυπτογραφημένο ηλεκτρονικό ταχυδρομείο του χρήστη μας αποθηκεύεται στο διακομιστή, για σύντομο χρόνο.

3.4 Τι Είναι Ο Έλεγχος Συνόδων Λειτουργίας.

Μπορεί να έχετε ακούσει ότι "το HTTP είναι ένα πρωτόκολλο χωρίς κατάσταση" που εννοεί αυτό είναι ότι το πρωτόκολλο δεν έχει ένα ενσωματωμένο τρόπο να δια την κατάσταση μεταξύ δυο συναλλαγών. Όταν ένας χρήστης ζητά μια σελίδα, ακολουθούμενη από μια άλλη, το HTTP δεν παρέχει ένα τρόπο να μας πει ότι οι δυο αιτήσεις ήρθαν από τον ίδιο χρήστη.

Η ιδέα του ελέγχου συνόδων λειτουργίας είναι για να μπορούμε να παρακολουθήσουμε χρήστη στη διάρκεια μιας συνόδου λειτουργίας του, σε μια Web τοποθεσία.

Αν μπορούμε να το κάνουμε αυτό, μπορούμε εύκολα να υποστηρίξουμε σύνδεση ενός χρήστη και εμφάνιση περιεχομένων σύμφωνα με το επίπεδο πιστοποίησης ή των προσωπικών προτιμήσεων του. Μπορούμε να παρακολουθήσουμε τη συμπεριφορά του χρήστη. Μπορούμε για παράδειγμα να χειριστούμε καλάθια αγορών.

Σε προηγούμενες εκδόσεις της PHP, ο έλεγχος συνόδων λειτουργίας υποστηριζότα την PHPLib, την PHP Base Library, που εξακολουθεί να είναι ένα χρήσιμο εργαλείο.

3.5 Βασική Λειτουργικότητα Συνόδων Λειτουργίας.

Οι σύνοδοι λειτουργίας στην PHP καθοδηγούνται από ένα μοναδικό κωδικό συν κρυπτογραφικά τυχαίο αριθμό. Ο κωδικός της συνόδου δημιουργείται από την PHPαποθηκεύεται στην πλευρά του πελάτη κατά την διάρκεια της συνόδου. Μπορεί να αποθηκευτεί είτε στον υπολογιστή ενός χρήστη σε ένα cookie είτε να περάσει μέσω των URL

Ο κωδικός συνόδου ενεργεί ως ένα κλειδί που μας επιτρέπει να εγγράφουμε συγκεκριμένες μεταβλητές, στις μεταβλητές συνόδων λειτουργίας. Τα περιεχόμενα αυτών των μεταβλητών αποθηκεύονται στον διακομιστή. Ο κωδικός συνόδου είναι η μόνη ορατή πληροφορία στην πλευρά του πελάτη. Αν, στην διάρκεια μια συγκεκριμένης σύνδεσης τοποθεσία μας, ο κωδικός συνόδου είναι ορατός είτε μέσω ενός cookie είτε μέσω URL μπορούμε να έχουμε πρόσβαση στις μεταβλητές συνόδου που είναι αποθηκευμένες στον διακομιστή για αυτή τη σύνοδο. Εξ ορισμού, οι μεταβλητές συνόδου αποθηκεύονται σε επίπεδα αρχεία στον διακομιστή. (Μπορείτε να το αλλάξετε αυτό και να χρησιμοποιήσετε μια βάση δεδομένων, αν θέλετε να γράψετε τη δική σας

συνάρτηση, αλλά περισσότερα(για αυτό στην ενότητα "Διαμόρφωση Ελέγχου Συνόδου Λειτουργίας").

Έχετε πιθανόν δει Web τοποθεσίες που αποθηκεύουν ένα κωδικό συνόδου στο URL. Αν υπάρχει μια συμβολοσειρά με τυχαία δεδομένα στο URL σας, είναι πιθανόν κάποια μορφή ελέγχου συνόδου.

Τα cookie είναι μια διαφορετική λύση στο πρόβλημα της διατήρησης της κατάστασης στην διάρκεια διαφόρων συναλλαγών, ενώ δίνουν ένα καθαρό URL.

Τι Είναι Ένα Cookie;

Ένα **cookie** είναι μια μικρή πληροφορία, που μπορεί να αποθηκεύσει ένα script στον υπολογιστή του πελάτη. Μπορείτε να ορίσετε ένα cookie σε έναν υπολογιστή του χρήστη'' στέλνοντας μια επικεφαλίδα HTTP, που περιέχει δεδομένα στην παρακάτω μορφή:

```
Set-Cookie:          NAME=VALUE;          [expires=04r£;    ]
[path=/»/177/; ] [aomai.n=DOMAIN_NAME\]   [secure]
```

Αυτό θα δημιουργήσει ένα cookie που ονομάζεται NAME, με τιμή VALUE. Οι άλλες παράμετροι είναι προαιρετικές. Το πεδίο expires ορίζει μια ημερομηνία πέρα από την οποίαν το cookie δεν είναι έγκυρο. (Σημειώστε ότι, αν δεν οριστεί η ημερομηνία λήξης, το cookie θα είναι μόνιμο, εκτός και αν διαγραφεί μη αυτόματα από εσάς ή το χρήστη).Μαζί, το path και το domain, μπορούν να χρησιμοποιηθούν για να καθοριστεί το URL ή τα URL με τα οποία σχετίζεται το cookie. Η λέξη κλειδί secure σημαίνει ότι το cookie δεν θα σταλεί μέσω απλής HTTP σύνδεσης.

Όταν ένας browser συνδέεται σε ένα URL, ψάχνει πρώτα τα cookie που είναι αποθηκευμένα τοπικά. Αν κάποια από αυτά είναι σχετικά με το URL στο οποίο συνδέεται, θα μεταδοθούν ξανά στον διακομιστή.

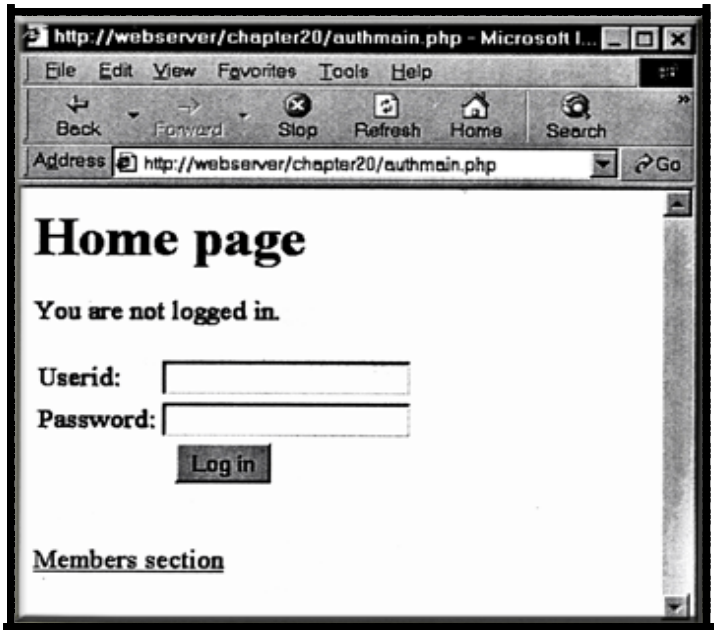
3.6 Χειρισμός Ελέγχου Ταυτότητας με Έλεγχο Συνόδων Λειτουργίας.

Τέλος, θα δούμε κάποιο πιο προχωρημένο παράδειγμα χρήσης ελέγχου συνόδων λειτουργίας.

Πιθανόν, η πιο συνηθισμένη χρήση του ελέγχου συνόδων λειτουργίας είναι να παρακολουθεί τους χρήστες αφού πιστοποιηθούν μέσω ενός μηχανισμού σύνδεσης. Σε αυτό το παράδειγμα, θα συνδυάσουμε πιστοποίηση από μια MySQL βάση δεδομένων με χρήση συνόδων λειτουργίας, για να παρέχουμε αυτή τη λειτουργικότητα.

Το παράδειγμα αποτελείται από τρία απλά script. Το πρώτο, το authmain.php, παρέχει μια φόρμα σύνδεσης και πιστοποίησης για μέλη της Web τοποθεσίας μας. Η δεύτερη εμφανίζει πληροφορίες μόνο για τα μέλη που έχουν συνδεθεί με επιτυχία. Η Τρίτη logout. php, αποσυνδέει ένα μέλος.

Αυτή είναι η αρχική εικόνα που εμφανίζεται από το **authmain.php**.



Εικόνα 3.2

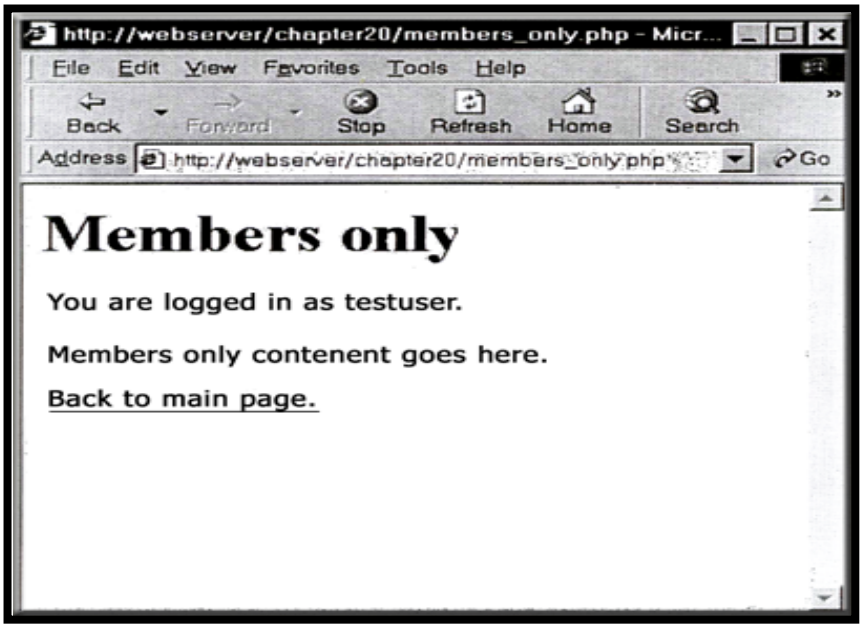
Επειδή ο χρήστης δεν έχει ακόμα συνδεθεί, δείξτε του μια σελίδα σύνδεσης.

Αυτή η σελίδα δίνει στον χρήστη ένα μέρος να συνδεθεί.Αν προσπαθήσει να έχει πρόσβαση στη σελίδα στην ενότητα members χωρίς να συνδεθεί πρώτα, θα πάρει το μήνυμα που φαίνεται στην παρακάτω εικόνα 3.3

Εικόνα 3.3

Οι χρήστες που δεν έχουν συνδεθεί, δεν μπορούν να δουν τα περιεχόμενα της τοποθεσίας και θα εμφανιστεί αντίθετα αυτό το μήνυμα.

Ωστόσο, αν ο χρήστης συνδεθεί πρώτα και μετά προσπαθήσει να δει τη σελίδα Members, θα δει την έξοδο που φαίνεται στην Εικόνα 3.4.



Εικόνα 3.4

Αφού ο χρήστης συνδεθεί, θα μπορεί να έχει πρόσβαση σε περιοχές μελών.

Ας δούμε τον κώδικα αυτής της εφαρμογής.

Το μεγαλύτερο μέρος του κώδικα είναι στο authmain. php. Αυτό το script μπορείτε να το Κώδικα 3.5.

Κώδικας 3.5
authmain.php

Το Κύριο Μέρος της Εφαρμογής Πιστοποίησης.

```
<?
session_start();

if ($userid && $password)
{

// αν ο χρήστης μόλις προσπάθησε να συνδεθεί

$db_conn = mysql_connect("localhost" "webauth" "webauth");
mysql_select_db("auth" $db_conn);
$query = "select * from auth "
      ."where name='$userid' "
      ." and pass=password('Spassword')";
$result = mysql_query($query $db_conn);
if (mysql_num_rows($result) >0 )

{
// αν είναι στην βάση δεδομε'νων, κάνε εγγραφή του user id
$valid_user = Suserid;
session_register("valid_user");
}
}
?>
<html>


<body>

<h1>Home page</h1>
}
if (session_is_registered( "valid_user" ) )
{
   echo "You are logged in as: $valid_user <br>";
   echo "<a href=\ "logout. php\ ">Log out</a><br>";
}
else
{
if (isset($userid) )
```

```
{
/ / αν δεν προσπάθησε και απέτυχε να συνδεθούν
    echo "Could not log you in";
}
else
}
/ / δεν προσπάθησαν να συνδεθούν ακόμα ή αποσυνδέθηκαν
echo "You are not logged in.<br>";
}
/ / παροχή φόρμας σύνδεσης
echo "<form method=post action=\ "authmain.php\ ">";
echo "<table>";
echo "<tr><td>Userld:</td>" ;
echo "<td><input type=text name=userid></td></tr>" ;
echo "<tr><td>Password:</td>" ;
echo "<td><input type=password name=password></td></tr>" ;
echo "<tr><td colspan=2 align=center>" ;
echo "<input type=submit value=\ "Log in\ "></td></tr>";
echo "</table></form>" ;
}
?>
<br>
<a href="members_only .php">Members section</a>
</body>
</html>
```

Υπάρχει κάποια λίγο περίπλοκη λογική σε αυτό το script, επειδή εμφανίζει τη φόρμα σύνδεσης και είναι επίσης και η ενέργεια της φόρμας.

Οι δραστηριότητες του script περιστρέφονται γύρω από την μεταβλητή συνόδου $valid_user. Η κύρια ιδέα είναι ότι, αν κάποιος συνδεθεί με επιτυχία, θα εγγράψουμε με μια μεταβλητή συνόδου που ονομάζεται $valid_user, που θα περιέχει τον κωδικό χρήστη.

Το πρώτο που κάνουμε στο script είναι να καλέσουμε την session_start (). Αυτή θα φορτώσει την μεταβλητή συνόδου $valid_user, αν έχει εγγραφεί.

Στο πρώτο πέρασμα του script, καμία από τις συνθήκες δεν ισχύει και ο χρήστης θα αποτύχει στο τέλος του script, όπου θα του πούμε ότι δεν είναι συνδεδεμένος και θα πρέπει να του δώσουμε μια φόρμα για να το κάνει:

```
echo   "<form method=post action=\   "authmain.php\  ">";
echo   "<table>";
echo   "<tr><td>Userid:</td>";
echo   "<td><input type=text name=userid></td></tr>";
echo   "<tr><td>Password:</td>";
echo   "<td><input type=password name=password></td></tr>";
echo   "<tr><td colspan=2 align=center>";
echo   "<input type=submit value=\   "Log in\   "></td></tr>";
```

```
echo   "</table></form>";
```

Όταν ο χρήστης πατήσει το κουπί Submit της φόρμας, ξανακαλείται αυτό το script και ξεκινάμε ξανά από την αρχή. Αυτή τη φορά, θα έχουμε κωδικό χρήστη και κωδικό πρόσβασης για πιστοποίηση, αποθηκευμένα στο Suserid και $password.Αν οριστούν αυτές οι μεταβλητές, πηγαίνουμε στο τμήμα πιστοποίησης:

```
if   ($userid && Spassword)
{
//  αν ο χρήστης μόλις προσπάθησε να συνδεθεί
$db_conn = mysql_connect("localhost"  "webauth"  "webauth");
mysql_select_db("auth"  $db_conn);
Squery =  "select * from auth  "
    ."where name='$userid'   "
    ."  and pass=password('Spassword')";
Sresult = mysql_query(Squery $db_conn);
```

Συνδεόμαστε σε μια MySQL βάση δεδομένων και ελέγχουμε το userid και το password. Αν υπάρχει ένα αντίστοιχο ζευγάρι στην βάση δεδομένων, εγγράφουμε την μεταβλητή $valid_user, που περιέχει το userid αυτού του χρήστη, ώστε να ξέρουμε ποιος είναι συνδεδεμένος για να τον παρακολουθούμε.

```
if  (mysql_num_rows($result)  >0  )
{
//αν είναι στην βάση δεδομένων, κάνε εγγραφή του user id
$valid_user = $userid;
session_register("valid_user");
}
}
```

Επειδή ξέρουμε ποιος είναι, δεν χρειάζεται να του δείξουμε ξανά τη φόρμα σύνδεσης. Αντίθετα, του λέμε ότι ξέρουμε ποιος είναι και του δίνουμε την επιλογή να αποσυνδεθεί:

```
if  (session_is_registered("valid_user"))
{
echo  "You are logged in as:  $valid_user <br>";
echo  "<a href=\   "logout.php\   ">Log out</a><br>";
}
```

Αν προσπαθήσουμε να τον συνδέσουμε και αποτύχουμε για κάποιο λόγο, θα έχουμε ένα userid αλλά όχι μια μεταβλητή $valid_user, έτσι θα πρέπει να του εμφανίσουμε ένα μήνυμα λάθους:

```
if   (isset($userid))
{
// αν δεν προσπάθησε και απέτυχε να συνδεθούν
echo  "Could not log you in";
}
```

Επειδή το $valid user είναι μια εγγεγραμμένη μεταβλητή συνόδου, δεν μπορεί να επικαλυφθεί από την προσπάθεια να πάρει διαφορετική τιμή στο URL, όπως στο παρακάτω:

```
members_only.php?valld_user=testuser
```

Αυτό είναι το κύριο script. Τώρα ας δούμε τη σελίδα Members. Ο κώδικας αυτού του script φαίνεται στο Κώδικα 3.6

Κώδικας 3.6
members_only.php

Ο Κώδικας για την Ενότητα Μελών της Web Τοποθεσίας μας, Ελέγχει για Έγκυρους Χρήστες

```
<?
session_start();
echo "<h1>Members only</h1>";

// ελέγχει την μεταβλητή συνόδου

if (session_is_registered("valid_user"))
{
echo "<p>You are logged in as $valid_user.</p>";
echo "<p>Members only content goes here</p>";
}
}
else
{
   echo "<p>You are not logged in.</p>";
   echo "<p>0nly logged in members may see this page.</p>";
}

  echo "<a href=\ "authmain.php\ ">Back to main page</a>";
```

```
?>
```

Αυτός ο κώδικας είναι πολύ απλός. Αυτό που κάνει είναι να ξεκινά μια σύνοδο και να ελέγχει αν η τρέχουσα σύνοδος περιέχει ένα εγγεγραμμένο χρήστη, χρησιμοποιώντας τη συνάρτηση session_registered_user ( ) . Αν ο χρήστης είναι συνδεδεμένος, του δείχνουμε τα περιεχόμενα του μέλους, διαφορετικά του λέμε ότι δεν είναι πιστοποιημένος.

Τέλος, έχουμε το script logout. php, που αποσυνδέει ένα χρήστη από το σύστημα. Ο κώδικας αυτού του script φαίνεται στο Κώδικα 3.7

Κώδικας 3.7
logout.php

Αυτό το Script Ακυρώνει την Εγγραφη της Μεταβλητής Συνόδου και Καταστρέφει τη Σύνοδο.

```
<?
session_start( ) ;

$old_user = $valid_user;  // το αποθηκεύει για να ελέγξει
αν*είχε* συνδεθεί
$result = session_unregister("valid_user" ) ;
session_destroy( ) ;
?>
<html>
<body>
<h1>Log out</h1>
<?
if ( !empty($old_user) )
{
if ($result)
{
  // αν συνδέθηκε και δεν αποσυνδέθηκε
    echo "Logged out.<br>";
}
else
{
 //  συνδέθηκε  και  δεν μπορούσε να αποσυνδεθεί
 echo   "Could not  log you out.<br>";

else
}
 //αν δεν συνδέθηκε αλλά ήρθε κάπως σε αυτή την σελίδα
 echo  "You were not logged in,   and so have not been logged
out.<br>";

}
?>
```

Ο κώδικας είναι πολύ απλός, αλλά κάνουμε και κάποια εντυπωσιακά πράγματα, Ξεκινάμε μια σύνοδο, αποθηκεύουμε το παλιό όνομα χρήστη, ακυρώνουμε την εγγραφή της έγκυρης μεταβλητής χρήστη και καταστρέφουμε την συνοδό. Μετά εμφανίζουμε στο χρήστη ένα μήνυμα ότι θα είναι διαφορετικός αν αποσυνδέθηκε, δεν μπορούσε να αποσυνδεθεί ή δεν συνδέθηκε αρχικά.

Αυτό το απλό σύνολο από script αποτελεί τη βάση για πολλή από τη δουλειά που θα κάνουμε στα επόμενα κεφάλαια.

3.7 Χειρισμός Ελέγχου Ταυτότητας Χρήστη

Υπάρχουν βασικά στοιχεία για την λειτουργική μονάδα του ελέγχου της ταυτότητας του χρήστη. Αυτά είναι η εγγραφή , η σύνδεση και η αποσύνδεση, η αλλαγή κωδικών πρόσβασης και η επαναφορά κωδικών πρόσβασης. Παρακάτω θα δούμε την εγγραφή ,τη σύνδεση και την αποσύνδεση ενός χρήστη στις οποίες θα αναφερθούμε και πιο αναλυτικά.

¨ *3.7.1 Εγγραφή.*

Για να εγγραφεί ένας χρήστης, θα πρέπει να πάρουμε τα στοιχεία του μέσω μιας φόρμας και θα τον εισάγουμε στη βάση δεδομένων.

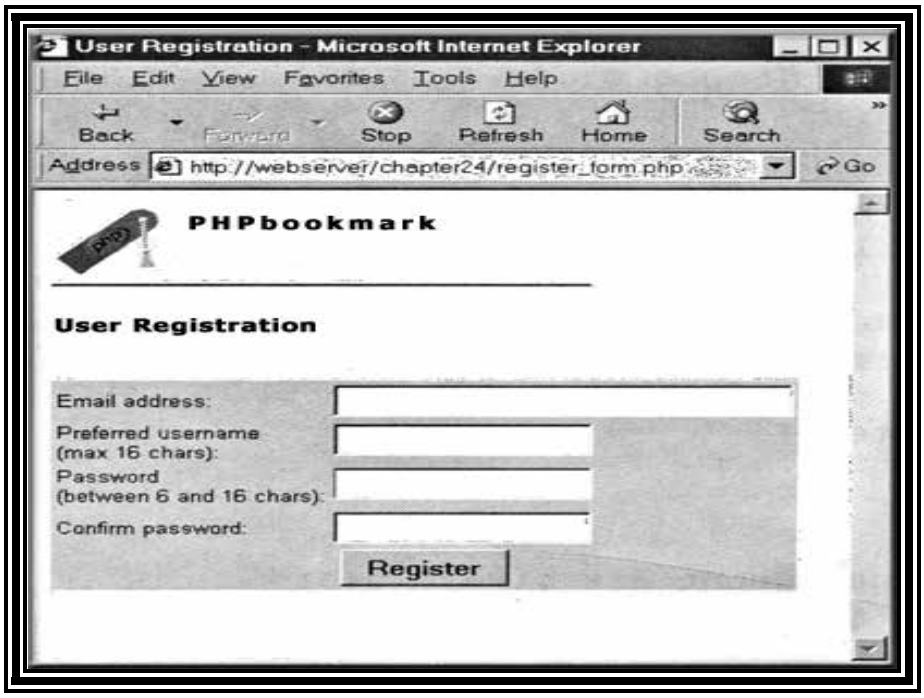
Όταν ένας χρήστης κάνει κλικ στη σύνδεση "Not a member?" (δεν είστε μέλος;) στη σελίδα login.php, θα μεταφερθεί σε μια φόρμα εγγραφής που δημιουργείται από το register_form.php. Αυτό το script φαίνεται στο Κώδικα 3.8.

Κώδικας 3.8
register_form.php

Αυτή η Φόρμα Δίνει στους Χρήστες την Ευκαιρία να Εγγραφούν στο PHPBookmarks.

```
<?
require_once("bookmark_fns.php");
do_html_header("User Registration");
```

```
display_registration_form();
,
do_html_footer();
?>
```



Εικόνα 3.5

Η φόρμα εγγραφής ανακτά τις λεπτομέρειες που χρειαζόμαστε για τη βάση δεδομένων.Βάζουμε τους χρήστες να πληκτρολογήσουν δύο φορές τους κωδικούς πρόσβασης, στην περίπτωση που κάνουν λάθος.

Και πάλι, μπορείτε να δείτε ότι αυτή η σελίδα είναι σχετικά απλή και ότι απλώς καλεί συναρτήσεις από τη βιβλιοθήκη εξόδου στο ou.tputjris.php. Η έξοδος αυτού του script φαίνεται στην Εικόνα 3.5.

Η γκρι φόρμα σε αυτή τη σελίδα είναι η έξοδος της συνάρτησης display_registration_form(), που περιλαμβάνεται στο output_fns.php. Όταν ο χρήστης κάνει κλικ στο κουμπί Register, μεταφέρεται στο script register_new.php. Αυτό το script φαίνεται στο Κώδικα 3.9.

Κώδικας 3.9
register_new.php

Αυτό το Script Επικυρώνει τα Δεδομένα του Νέου Χρήστη και τα Τοποθετεί στη Βάση Δεδομένων.

```
<?
//  συμπερίληψη των αρχείων της συναρτήσεων της εφαρμογής
require_once("bookmark_fns.php");

// έναρξη συνόδου που μπορεί να χρειασθεί αργότερα
// ξεκινήστε το τώρα επειδή πρέπει να πάει πριν τις επικεφαλίδες
session_start();

// έλεγχος συμπληρωμένης φόρμας
if ( !f illed_out($HTTP_POST_VARS) )
{
do_html_header ( " Problem : " ) ;
echo   "You have not filled the form out correctly
. "   and try again. " ;
do_html_f ooter ( ) ;
exit;
}

//η διεύθυνση ηλεκτρονικού ταχυδρομείου δεν είναι έγκυρη
if ( !valid_email($email) )
}
do_html_header( "Problem: " ) ;
echo "That is not a valid email address.
. " and try again. " ;
do_html_footer( ) ;
exit;
}

// οι κωδικοί πρόσβασης δεν είναι ίδιοι
if (Spasswd != $passwd2)
{
do_html_heading ( "Problem : " ) ;
echo "The passwords you entered do not match - please go back"
. " and try again. " ;
do_html_f ooter () ;
exit;


}

// έλεγχος αν το μήκος του κωδικού πρόσβασης είναι σωστό
// είναι εντάξει αν ο χρήστης το κόψει, αλλά θα υπάρξει
// πρόβλημα αν είναι πολύ μεγάλος
if (strlen($passwd)<6 || strlen($passwd) >16)
{
do_html_header ( "Problem : " ) ;
echo "Your password must be between 6 and 16 characters.
."Please go back and try again.";
```

```
do_html_footer() ;
exit;

}

//  προσπάθεια για εγγραφή
$reg_result = register($username,  $email,  $passwd);
if   ($reg_result ==  "true")

{
// εγγραφή μεταβλητής συνόδου
$valid_user = Susername;
session_register("valid_user");

// παρέχει σύνδεση στην σελίδα μελών
do_html_header("Registration successful");
echo "Your registration was successful. Go to the members page
."to start setting up your bookmarks!";
do_HTML_URL("member.php", "Go to members page");
}
else
}
// διαφορετικά,   δίνει σύνδεση προς τα πίσω και να προσπαθήσουν
ξανά
do_html_header(" Problem: ") ;
echo $reg_result;
do_html_footer() ;
exit;
}
//  τέλος σελίδας
do_html_footer() ;
?>
```

Αυτό είναι το πρώτο script που είναι κάπως πιο πολύπλοκο από όσα είδαμε σε αυτήν την εφαρμογή.

Το script ξεκινά συμπεριλαμβάνοντας τα αρχεία συναρτήσεων της εφαρμογής και ξεκινά μια συνοδό λειτουργίας (session). (Αν ο χρήστης έχει εγγραφεί, θα χρησιμοποιήσουμε το όνομα χρήστη ως μεταβλητή συνόδου λειτουργίας).

Στη συνέχεια, επικυρώνουμε τα δεδομένα που εισάγει ο χρήστης. Υπάρχουν διάφορες συνθήκες που πρέπει να ελέγξουμε. Είναι οι εξής:

- Ελέγχουμε αν η φόρμα έχει συμπληρωθεί. Αυτό το ελέγχουμε καλώντας τη συνάρτηση f illed_out ( ) ως εξής:

```
if   ( !filled_OUt($HTTP_POST_VARS) )
```

Αυτή τη συνάρτηση την έχουμε γράψει μόνοι μας. Βρίσκεται στη βιβλιοθήκη συναρτήσεων του αρχείου data_valid_fns.php. Θα δούμε τη συνάρτηση αυτή σε λίγο.

- Ελέγχουμε αν είναι έγκυρη η διεύθυνση ηλεκτρονικού ταχυδρομείου που δόθηκε από το χρήστη. Αυτό γίνεται με τον εξής τρόπο:

```
if  (valid_email($email) )
```

Και πάλι, αυτή είναι μια συνάρτηση που έχουμε γράψει εμείς και βρίσκεται στη βιβλιοθήκη data_valid_fns.php.

- Ελέγχουμε αν οι δύο κωδικοί πρόσβασης που έδωσε ο χρήστης είναι ίδιοι, ως εξής:

```
if  ($passwd   != $passwd2)
```

- Ελέγχουμε αν ο κωδικός πρόσβασης έχει το κατάλληλο μέγεθος, ως εξής:

```
if   (strlen($passwd)<6  ||   strlen($passwd)   >16)
```

Στο παράδειγμα μας, ο κωδικός πρόσβασης θα πρέπει να έχει τουλάχιστον 6 χαρακτήρες, για να είναι πιο δύσκολο να τον μαντέψει κανείς και είναι μικρότερος από 16 χαρακτήρες, για να χωράει στη βάση δεδομένων.

Οι συναρτήσεις επικύρωσης των δεδομένων που χρησιμοποιήσαμε εδώ, οι filled_out ( )και valid_email(), εμφανίζονται στο Κώδικα 3.10 και στο Κώδικα 3.11, αντίστοιχα.

Κώδικας 3.10
Συνάρτηση filled_out() από το data_valid_fns.php

Αυτή η Συνάρτηση Ελέγχει αν η Φόρμα Έχει Συμπληρωθεί.

```
function f illed_out ($form_vars)
{
// έλεγχος ότι κάθε μεταβλητή έχει μια τιμή
foreach ($form_vars as $key => $value)
}
if  (!isset($key) ($value ==  ""))
return false;
}
return true;
}
```

Κώδικας 3.11
Συνάρτηση valid_email() από το data_valid_fns.php

Αυτή η Συνάρτηση Ελέγχει αν Είναι Έγκυρη μια Διεύθυνση Ηλεκτρονικού Ταχυδρομείου.

```
function valid_email($address)
{
//  έλεγχος αν η διεύθυνση ηλεκτρονικού ταχυδρομείου είναι
έγκυρη
if   (ereg(""[a-zA-ZO-9_]+@[a-zA-ZO-9\  -]+\ .[a-zA-ZO-9\-\
.]+$",Saddress))
return true;
else
return false;
}
```

Η συνάρτηση filled_out () περιμένει να της περάσει ένας πίνακας μεταβλητών γενικά οι πίνακες αυτοί θα είναι οι πίνακες $HTTP_POST_VARS ή $HTTP_GET_VARS. Θα ελέγξει αν έχουν συμπληρωθεί όλοι και θα επιστρέψει την τιμή true αν έχουν συμπληρωθεί και false αν δεν έχουν συμπληρωθεί.

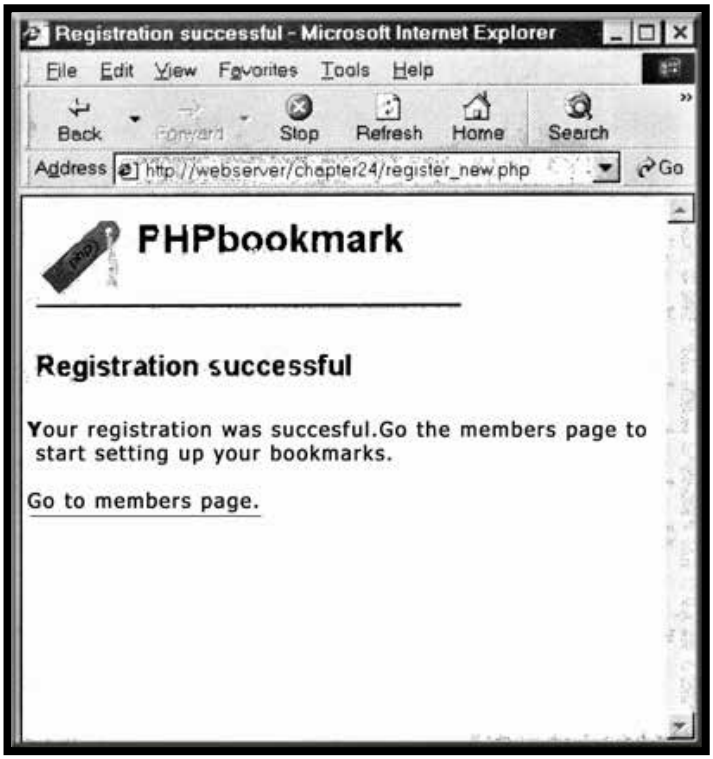
Η συνάρτηση valid_email() χρησιμοποιεί την κανονική παράσταση που για τον έλεγχο της εγκυρότητας ηλεκτρονικών διευθύνσεων. Επιστρέφει true, αν μια διεύθυνση φαίνεται έγκυρη και false στην αντίθετη περίπτωση.

Αφού ελέγξουμε την εγκυρότητα των δεδομένων που έδωσε ο χρήστης, μπορούμε να εγγράψουμε το χρήστη. Αν δείτε ξανά τη Λίστα 3.9, θα δείτε ότι αυτό γίνεται με τον εξής τρόπο:

```
$reg_result = register($username,$email, $passwd);
if   ($reg_result ==  "true")
{
// εγγραφή μεταβλητής συνόδου
$valid_user = Susername;
session_register("valid_user");

// παρέχει σύνδεση στην σελίδα μελών
do_html_header("Registration successful");
echo "Your registration was successful. Go to the members
     ."to start setting up your bookmarks!";
do_HTML_URL("member.php", "Go to members page");
}
```

Όπως μπορείτε να δείτε, καλούμε τη συνάρτηση register () με το όνομα χρήστη, την διεύθυνση ηλεκτρονικού ταχυδρομείου και τον κωδικό πρόσβασης που έδωσε ο χρήστης .Αν είναι επιτυχής η εγγραφή, εγγράφουμε το όνομα του

χρήστη ως μια μεταβλητή συνόδου λειτουργίας και παρέχουμε στο χρήστη μια σύνδεση για τη βασική σελίδα των μελών. Αυτό είναι το αποτέλεσμα που φαίνεται στην Εικόνα 3.6.



Εικόνα 3.6

Η εγγραφή ήταν επιτυχή -ο χρήστης μπορεί τώρα να πάει στη σελίδα των μελών.

Η συνάρτηση register() βρίσκεται στη συμπεριλαμβανόμενη βιβλιοθήκη που ονομάζεται user_auth_fns.php. Αυτή η συνάρτηση φαίνεται στο Κώδικα 3.12.

Κώδικας 3.12
Συνάρτηση register ( ) από το user_auth_fns.php

Αυτή η Συνάρτηση Επιχειρεί να Τοποθετήσει τις Πληροφορίες του Νέου Χρήστη στη Βάση Δεδομένων.

```
function register($username, $email, Spassword)
// εγγραφή νέου ατόμου στην βάση δεδομένων
// επιστροφή true ή μήνυμα λάθους
{
// σύνδεση με την βάση δεδομένων
$conn = db_connect();
if (!$conn)
return "Could not connect to database server - please try
later.";

// έλεγχος αν το όνομα χρήστη είναι μοναδικό
```

```
$result = mysql_query("select * from user where
username='$username'");
if (!$result)
return "Could not execute query";
if (mysql_num_rows($result)>0)
return "That username is taken - go back and choose another
one.";
// αν είναι εντάξει, το βάζει στην βάση δεδομένων
$result = mysql_query("insert into user values
('$username', password('Spassword'), '$email')");
if (!$result)
return "Could not register you in database - please try again
later.";
return true;

}
```

Δεν υπάρχει κάτι καινούργιο σε αυτή τη συνάρτηση - συνδέεται με τη βάση δεδομένων που δημιουργήσαμε νωρίτερα. Αν το όνομα χρήστη που επιλέχθηκε ανήκει σε κάποιον άλλο χρήστη ή αν η βάση δεδομένων δεν μπορεί να ενημερωθεί, θα επιστραφεί false. Αλλιώς, θα ενημερωθεί η βάση δεδομένων και θα επιστρέψει true. Κάτι που πρέπει να τονίσουμε είναι ότι εκτελούμε τη σύνδεση με την βάση δεδομένων με μια συνάρτηση που έχουμε γράψει, η οποία ονομάζεται db_connect(). Αυτή η συνάρτηση απλώς παρέχει μια θέση που περιέχει το όνομα χρήστη και τον κωδικό πρόσβασης για τη σύνδεση με τη βάση δεδομένων. Με τον τρόπο αυτό, αν αλλάξουμε τον κωδικό πρόσβασης στη βάση δεδομένων, θα χρειαστεί να αλλάξουμε ένα μόνο αρχείο της εφαρμογής μας. Η συνάρτηση φαίνεται στο Κώδικα 3.13

Κώδικας 3.13

Συνάρτηση db_connect() από το dbjns.php

Αυτή η Συνάρτηση Συνδέεται με τη MySQL βάση δεδομένων.

```
function db_connect()
$result = mysql_pconnect("localhost"
if (!$result)
return false;
if (!mysql_select_db("bookmarks"))
return false;
return $result;
"bm_user",   "password");
please try later.";
username='$username1
choose another one. ";
```

Όταν εγγραφούν οι χρήστες, μπορούν να συνδέονται και να αποσυνδέονται χρησιμοποιώντας τις κανονικές σελίδες σύνδεσης και αποσύνδεσης. Θα τις δημιουργήσουμε στη συνέχεια.

¨ *3.7.2 Σύνδεση.*

Αν οι χρήστες συμπληρώσουν τις πληροφορίες στη φόρμα login.php και την στείλουν, θα μεταφερθούν στο script που ονομάζεται member.php. Αυτό το script θα τους συνδέσει, αν έρχονται από αυτή τη φόρμα. Επίσης θα εμφανίσει σχετικούς σελιδοδείκτες στους χρήστες που είναι συνδεδεμένοι. Είναι το κέντρο της υπόλοιπης εφαρμογής. Αυτό το script φαίνεται στο Κώδικα 3.14

Κώδικας 3.14
member.php

Αυτό το script Είναι το Κέντρο της Εφαρμογής.

```
<?

// συμπερίληψη αρχείων αυτής της εφαρμογής
require_once("bookmark_fns.php");
session_start();

if ($username && Spasswd)
// μόλις προσπάθησαν να συνδεθούν
}
if (login($username, $passwd))
}
// αν είναι στην βάση δεδομένων, εγγράφεται ο κωδικός
χρήστης
$valid_user = Susername;
session_register("valid_user");
}
else
}
// ανεπιτυχής σύνδεση
do_html_header("Problem:");
echo "You could not be logged in.
You must be logged in to view this page.";
do_html_url("login.php", "Login");
do_html_footer();
exit;
}
```

```
}
do_html_header("Home");
check_valid_user();
// λήψη των σελιδοδεικτών που έχει αποθηκεύσει αυτός ο
χρήστης
if ($url_array = get_user_urls($valid_user));
display_user_urls($url_array);

// δίνει επιλογές μενού
display_user_menu();
do_html_footer();
```

Επεξήγηση του παραπάνω Script

Πρώτον, ελέγχουμε αν ο χρήστης έρχεται από την πρώτη σελίδα - αν δηλαδή, έχει μόλις συμπληρώσει τη φόρμα σύνδεσης - και προσπαθούμε να τον συνδέσουμε με τον παρακάτω τρόπο:

```
if ($username && $passwd)
// μόλις προσπάθησαν να συνδεθούν
}
if (login($username, Spasswd))
}
// αν είναι στην βάση δεδομένων, εγγράφεται ο κωδικός χρήστης
$valid_user = $username;
session_register("valid_user");
}
```

Μπορείτε να δείτε ότι προσπαθούμε να συνδέσουμε το χρήστη χρησιμοποιώντας μια συνάρτηση που ονομάζεται login(). Αυτή τη συνάρτηση την έχουμε ορίσει στη βιβλιοθήκη user auth_fns.php και θα δούμε τον κώδικα της σε λίγο.

Αν η σύνδεση γίνει με επιτυχία, εγγράφουμε αυτήν την σύνοδο λειτουργίας, όπως κάναμε πριν, αποθηκεύοντας το όνομα χρήστη στη μεταβλητής συνόδου λειτουργίας $valid_user.

Αν όλα πάνε καλά, εμφανίζουμε στο χρήστη τη σελίδα των μελών:

```
do_html_header("Home");
check_valid_user();
```

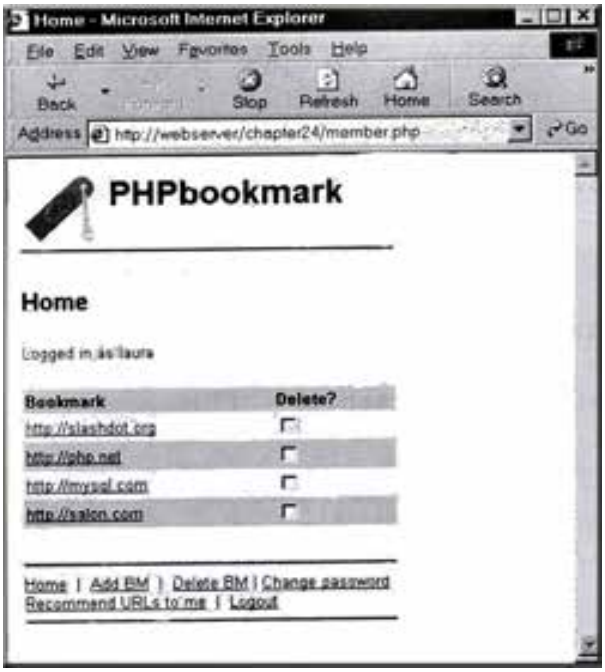
```
// λήψη των σελιδοδεικτών που έχει αποθηκεύσει αυτός ο χρήστης
if ($url_array = get_user_urls($valid_user));
display_user_urls($url_array);

// δίνει επιλογές μενού
display_user_menu();
do_html_footer();
```

Αυτή η σελίδα δημιουργείται ξανά χρησιμοποιώντας τις συναρτήσεις εξόδου. Θα παρατηρήσετε ότι χρησιμοποιούμε διάφορες νέες συναρτήσεις. Αυτές είναι η check_valid_user(), από το user_auth_fns.php, η get_user_urls() από το url_fns.php και display_user_urls () από το output_f ns. php. Η συνάρτηση check_valid_user () ελέγχει αν ο τρέχων χρήστης έχει μια εγεγραμμένη σύνοδο λειτουργίας. Αυτό απευθύνεται στους χρήστες που δεν έχουν μόλις συνδεθεί, αλλά βρίσκονται στη μέση της συνόδου λειτουργίας.

Η συνάρτηση get_user_urls () παίρνει τους σελιδοδείκτες ενός χρήστη από τη βάση δεδομένων και η display_user_urls() εμφανίζει τους σελιδοδείκτες σε μορφή πίνακα στο browser. Θα δούμε τη συνάρτηση check_valid_user() σε λίγο και τις άλλες δύο στην ενότητα για την αποθήκευση και την ανάκτηση σελιδοδεικτών.

Το script member.php σταματά τη σελίδα εμφανίζοντας ένα μενού με τη συνάρτηση display_user_menu( ).

Στην παρακάτω εικόνα φαίνεται ένα παράδειγμα αυτού που εμφανίζει το **member.php.**



Εικόνα 3.7

Κώδικας 3.15
Η Συνάρτηση login από το user_auth_fns.php

Αυτή η Συνάρτηση Ελέγχει τα Στοιχεία του Χρήστη σε Σχέση με τη Βάση Δεδομένων.

```
function login($username,   Spassword)
//  ελέγχει  αν το όνομα χρήστη και ο κωδικός πρόσβασης είναι
στην βάση δεδομένων
// αν είναι, επιστρέφει true
// διαφορετικά επιστρέφει false
{
// σύνδεση με την βάση δεδομένων
$conn = db_connect();
if (!$conn)
return 0;


// ελέγχει αν το όνομα χρήστη είναι μοναδικό

$result = mysql_query("select * from user
where username='$username'
and passwd = password('Spassword')");

if (!$result)

return 0;
if (mysql_num_rows($result)>0)
return 1;
else
return 0;
}
```

Όπως μπορείτε να δείτε, αυτή η συνάρτηση συνδέεται με τη βάση δεδομένων και ελέγχει αν υπάρχει χρήστης με αυτό το συνδυασμό ονόματος χρήστη και κωδικού πρόσβασης. Θα επιστρέψει true αν ο συνδυασμός υπάρχει και false αν δεν υπάρχει ή αν τα στοιχεία πιστοποίησης του χρήστη δεν μπορούν να ελεγχθούν.

Η συνάρτηση **check valid user ( )** δεν συνδέεται με τη βάση δεδομένων ξανά, αλλά ελέγχει αν ο χρήστης έχει μια εγγεγραμμένη σύνοδο λειτουργίας, δηλαδή αν έχει ήδη συνδεθεί.

Η συνάρτηση αυτή φαίνεται στο Κώδικα 3.15

Κώδικας 3.15
Η Συνάρτηση check_valid_user() από το user_auth_fns.php

Αυτή η Συνάρτηση Ελέγχει αν ο Χρήστης Έχει μια Έγκυρη Σύνοδο λειτουργίας.

```
function check_valid_user()
// βλέπει αν κάποιος είναι συνδεδεμένος και ειδοποιεί αν
όχι

{
global $valid_user;
if (session_is_registered( "valid_user" ) )
{
echo "Logged in as $valid_user. " ;
echo "<br>";
else
}
//  δεν είναι  συνδεδεμένος
do_html_heading( "Problem: " ) ;
echo  "You are not logged in.<br>";
do_html_url( " login. php" ,   "Login" ) ;
do_html_f ooter ( ) ;
exit;

}
}
```

Αν ο χρήστης δεν έχει συνδεθεί, η συνάρτηση θα του πει ότι πρέπει να συνδεθεί για να δει αυτή τη σελίδα και θα του δώσει μια σύνδεση προς τη σελίδα που γίνεται η σύνδεση.

3.7.3 Αποσύνδεση.

Μπορεί να έχετε παρατηρήσει ότι υπάρχει μια σύνδεση που ονομάζεται "Logout" στο μενού της Εικόνας 3.7 Αυτή είναι μια σύνδεση προς το script **logout.php**. Ο κώδικας αυτού του script φαίνεται στο Κώδικα 3.16.

Κώδικας 3.16
logout.php

Αυτό το script τερματίζει μια Σύνοδο Λειτουργίας.

```
<?
```

```
// συμπερίληψη αρχείων συναρτήσεων για αυτή την εφαρμογή
require_once( "bookmark_fns.php" ) ;
session_start() ;
$old_user = $valid_user;  // αποθήκευση για να ελέγξουμε αν είχαν
συνδεθεί
$result_unreg = session_unregister( "valid_user" ) ;
$result_dest = session_destroy() ;
// έναρξη της html εξόδου
do_html_header(" Logging Out");
if (! empty ($old_user))
{
if ($result_unreg && $result_dest)
{
//αν ήταν συνδεδεμένος και τώρα αποσυνδέθηκε
echo "Logged out.<br>";
do_html_url( " login. php" , "Login") ;
}
else
{
//αν ήταν συνδεδεμένος και δεν μπορεί να αποσυνδεθεί
echo "Could not log you out.<br>";
}
}
else
{
//αν δεν ήταν συνδεδεμένος, αλλά ήρθε με κάποιο τρόπο σε αυτή την
σελίδα
echo "You were not logged in, and so have not been logged
out.<br>";
do_html_url( "login. php" , "Login" ) ;
}
do_html_footer() ;
?>
```

Βιβλιογραφία

1) **Leon Atkinson and Zeev Suraski ,"Πλήρης οδηγός της Php 5",** Εκδόσεις Μ.Γκιούρδας.

2) **Leon Atkinson ,"Πλήρης οδηγός της Php 4" ,** Εκδόσεις Μ.Γκιούρδας

3) **Luke Welling and Laura Thomson , " Ανάπτυξη Web Εφαρμογών με Php και MySql"** , Εκδόσεις Μ.Γκιούρδας.

4) **Smievski and Hughes, "Php Οδηγός Προγραμματισμού" ,** Εκδόσεις Μ.Γκιούρδας.

5) **Wankyu Choi, Allan Kent, Chris Lea, Ganesh Prasad, Chris Ullman, Jon Blank, Sean Cazzell , "Beginning PHP 4" ,** Εκδόσεις Wrox

Ιστοσελίδες

1) http://openwebmail.mirror.luxadmin.org/manual/el/introduction.php

2) http://grjava.com/gr/php scripts

3) http: / /www. freewebmasterhelp .com / tutorials / php

4) http:/ /martin, f2o.org/php /login

5) http://www.signal42.com/php login scripts with mysqlasp

6) http://www.devshed.eom/c/a/PHP/Creating-a-Secure-PHP-Login-Script/ http: / /www. webscriptsdirectory.com/PHP/User-Authentication/