ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πτυχιακή Εργασία

<<Αξιολόγηση μέτρων ασφάλειας Web πληροφοριακών συστημάτων με την χρήση του IMB Rational Appscan>>

Φοιτητής:
Τσολακίδης Πασχάλης
Αρ.Μητρώου: 03/2286

Επιβλέπων Καθηγητής:
Ηλιούδης Χρήστος

ΘΕΣΣΑΛΟΝΙΚΗ 2010

ΠΡΟΛΟΓΟΣ

Ένας από τους κυριότερους τομείς της ασφάλειας των υπολογιστών είναι αυτός που σχετίζεται με τα Web πληροφοριακά συστήματα και ειδικότερα με τις ευπάθειες των διαδικτυακών τους εφαρμογών. Τα Web πληροφοριακά συστήματα βρίσκονται μόνιμα σε κοινή πρόσβαση, για την εξυπηρέτηση των χρηστών και έτσι είναι πιο επιρρεπή σε επιθέσεις. Οι ευπάθειες τέτοιων συστημάτων πρέπει να αντιμετωπίζονται άμεσα, καθώς η εκμετάλλευσή τους από έναν κακόβουλο χρήστη δύναται να έχει καταστροφικές συνέπειες.

Η ασφάλεια του τμήματος των διαδικτυακών εφαρμογών, των Web πληροφοριακών συστημάτων, αποτελεί σημαντική υπόθεση. Οι διαδικτυακές εφαρμογές ορίζονται ως πρωτεύοντας στόχος των κακόβουλων χρηστών καθώς το εβδομήντα πέντε τις εκατό (75%) των επιθέσεων γίνονται κατά του επιπέδου εφαρμογών (application layer). Σύμφωνα με μελέτες εννιά στις δέκα ιστοσελίδες περιέχουν ευπάθειες διαδικτυακών εφαρμογών. Επίσης λόγω του είδους των δεδομένων που διαχειρίζονται (προσωπικά στοιχεία, αριθμοί καρτών κ.τ.λ.) αποτελούν στόχους υψηλής αξίας για τους επιτιθέμενους. Όλα τα παραπάνω επισημαίνουν την ανάγκη για αντιμετώπιση αυτών των ευπαθειών. (Shawn Miller 2007)

Έρευνες έδειξαν έναν πολύ μεγάλο αριθμό από Web πληροφοριακά συστήματα, τα οποία είναι ευάλωτα σε επιθέσεις κατά διαδικτυακών εφαρμογών. Το μεγαλύτερο ποσοστό αυτών των επιθέσεων γίνεται μέσω των πρωτοκόλλων HTTP/S οι πόρτες των οποίων είναι συχνά εκτεθειμένες σε όλη την διαδικτυακή κοινότητα. Καθώς οι διαδικτυακές εφαρμογές έχουν γίνει τρομερά πολύπλοκες, ολοένα και περισσότερα προσωπικά, ιατρικά και οικονομικά δεδομένα ανταλλάσσονται και αποθηκεύονται. Οι χρήστες περιμένουν ή ακόμα απαιτούν τα στοιχεία αυτά να παραμένουν ασφαλή. Λαμβάνοντας αυτό υπόψη τους οι οργανισμοί οφείλουν να πάρουν δραστικά μέτρα για την ασφάλεια των εφαρμογών τους. (Danny Allan 2008)

ΠΕΡΙΛΗΨΗ

Αντικείμενο της πτυχιακής εργασίας είναι ο προσδιορισμός των ευπαθειών των Web πληροφοριακών συστημάτων και η δημιουργία ενός μεθοδολογικού πλαισίου αξιολόγησής τους. Στόχος της είναι να αναλυθούν και να κατηγοριοποιηθούν οι αδυναμίες που εμφανίζονται τόσο στην δικτυακή επικοινωνία (των επιπέδων TCP/IP) όσο και στις λειτουργικές οντότητες (DB – Application – Web server) ενός τέτοιου πληροφοριακού συστήματος και να προταθεί ένα πλαίσιο προσδιορισμού τους. Στα πλαίσια της πτυχιακής θα χρησιμοποιηθεί το λογισμικό IBM Rational AppScan και θα δημιουργηθούν σενάρια εφαρμογής του.

Η εργασία ξεκινάει στο πρώτο κεφάλαιο με μια περιγραφή των πληροφοριακών συστημάτων και της αρχιτεκτονικής τους. Αυτό βοηθάει τον αναγνώστη να κατανοήσει τον διαχωρισμό των ευπαθειών ενός τέτοιου συστήματος, που θα πραγματοποιηθεί στη συνέχεια της εργασίας. Έπειτα γίνεται μια ανάλυση των κυριότερων ευπαθειών και εξηγείται το πως γίνεται η εκμετάλλευση τους από τους κακόβουλους χρήστες. Στη συνέχεια στο δεύτερο κεφάλαιο γίνεται η δημιουργία του μεθοδολογικού πλαισίου με το οποίο αξιολογούμε και κατηγοριοποιούμε τις διάφορες ευπάθειες με βάση κάποια χαρακτηριστικά τους γνωρίσματα. Με αυτόν τον τρόπο έχουμε ένα μοντέλο το οποίο κρίνοντας αρκετούς παράγοντες ταξινομεί κάθε είδους ευαισθησία του συστήματος μας. Στο τρίτο κεφάλαιο γίνεται μια εισαγωγή του αναγνώστη στο πρόγραμμα της IBM το Rational Appscan, που πρόκειται για εφαρμογή η οποία εξετάζει άλλες εφαρμογές του διαδικτύου και μας αναφέρει τις ευπάθειες που ανακάλυψε σε αυτές. Το τέταρτο κεφάλαιο ασχολείται με διάφορα σενάρια εφαρμογής του Rational Appscan που υλοποιούνται για να δειχθούν τα συμπεράσματα και οι αναφορές που προκύπτουν από τις εξετάσεις. Τέλος στο πέμπτο κεφάλαιο παραθέτονται τα συμπεράσματα στα οποία κατέληξε η πτυχιακή αυτή εργασία σε ότι αφορά την αξιολόγηση των ευπαθειών που έχουν τα Web πληροφοριακά συστήματα.

ABSTRACT

The subject of this final thesis is the determination of the vulnerabilities that trouble the Web Information Systems and the creation of a methodological framework for their evaluation. The aim of this final thesis is to analyze and categorize the weaknesses that appear in both network communications (TCP/IP) and the functional parts (DB-Applications-Web Server) of the Web Information Systems and to propose a framework to identify them. As part of the final thesis the IBM Rational Appscan software will be used and we will create scenarios about its usage.

This final thesis begins in the first chapter with a description of Web Information Systems and their architecture. This helps the reader to understand the separation of the vulnerabilities of such an information system, to different categories as the final thesis proceeds. Following that is an analysis of key vulnerabilities and how they are exploited by malicious users. In the second chapter there is the creation of the methodological framework by which we evaluate and categorize the different vulnerabilities based on some of their characteristics. Doing that we have a model that categorize every vulnerability of a Web Information System, examining various criteria. The third chapter the reader is introduced to the IBM Rational Appscan, a program that scans Web applications and presents the vulnerabilities it discovered. The forth chapter focuses on the usage scenarios of IBM Rational Appscan to demonstrate the findings and reports of the program. Finally the last chapter has the conclusions of this final thesis about the evaluation of Web Information System’s vulnerabilities.

ΕΥΧΑΡΙΣΤΙΕΣ

Ένα μεγάλο ευχαριστώ στον καθηγητή μου κ. Ηλιούδη Χρήστο που συνεργάστηκε μαζί μου για την πραγματοποίηση της παρούσας πτυχιακής και στους δικούς μου ανθρώπους που με στήριξαν κατά την διάρκεια των σπουδών μου.

Αφιερωμένη στον πατέρα μου...

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΥΡΕΤΗΡΙΟ ΠΙΝΑΚΩΝ – ΣΧΗΜΑΤΩΝ

ΚΕΦΑΛΑΙΟ 1: Ευπάθειες Web πληροφοριακών συστημάτων

Στο κεφάλαιο αυτό θα ασχοληθούμε με τις ευπάθειες που παρουσιάζουν τα Web πληροφοριακά συστήματα. Στην αρχή θα γίνει μια περιγραφή του τι ορίζουμε ως Web πληροφοριακό σύστημα και ποιες διαφορές το διακρίνουν από το πληροφοριακό σύστημα. Έπειτα έχοντας αναλύσει την σύνθεση του θα χρησιμοποιήσουμε την κατηγοριοποίηση των ευπαθειών που χρησιμοποίει το OWASP (Open Web Application Security Project) για να περιγράψουμε τις κυριότερες από αυτές.

1.1 Εισαγωγή στα Web πληροφοριακά συστήματα

Αρχικά πριν να μελετήσουμε τις ευπάθειες των Web πληροφοριακών συστημάτων θα παρουσιάσουμε τα βασικά χαρακτηριστικά ενός τέτοιου συστήματος. Με αυτόν τον τρόπο αργότερα θα εστιάσουμε καλύτερα στο ποιές είναι οι πιθανές ευαισθησίες του, ποιό μέρος του συστήματος επηρεάζουν και ποιά λύση μπορεί να τις διορθώσει και να μας παρέχει ασφάλεια.

Ένα πληροφοριακό σύστημα βοηθάει στον έλεγχο, στο συντονισμό, στην ανάλυση προβλημάτων, στη λήψη αποφάσεων και στην ανάπτυξη νέων προϊόντων. Είναι ένα οργανωμένο σύνολο το οποίο αποτελείται από τα εξής στοιχεία: (Τασόπουλος, Αθήνα 2005)

Ανθρώπινο Δυναμικό (Το σύνολο των ανθρώπων που εργάζονται με το πληροφοριακό σύστημα σε διάφορους ρόλους όπως χρήστες, διαχειριστές κ.τ.λ.)

Διαδικασίες (Το σύνολο των οδηγιών για την χρήση και τον συνδυασμό όλων των στοιχείων υποδομής ενός Πληροφοριακού Συστήματος)

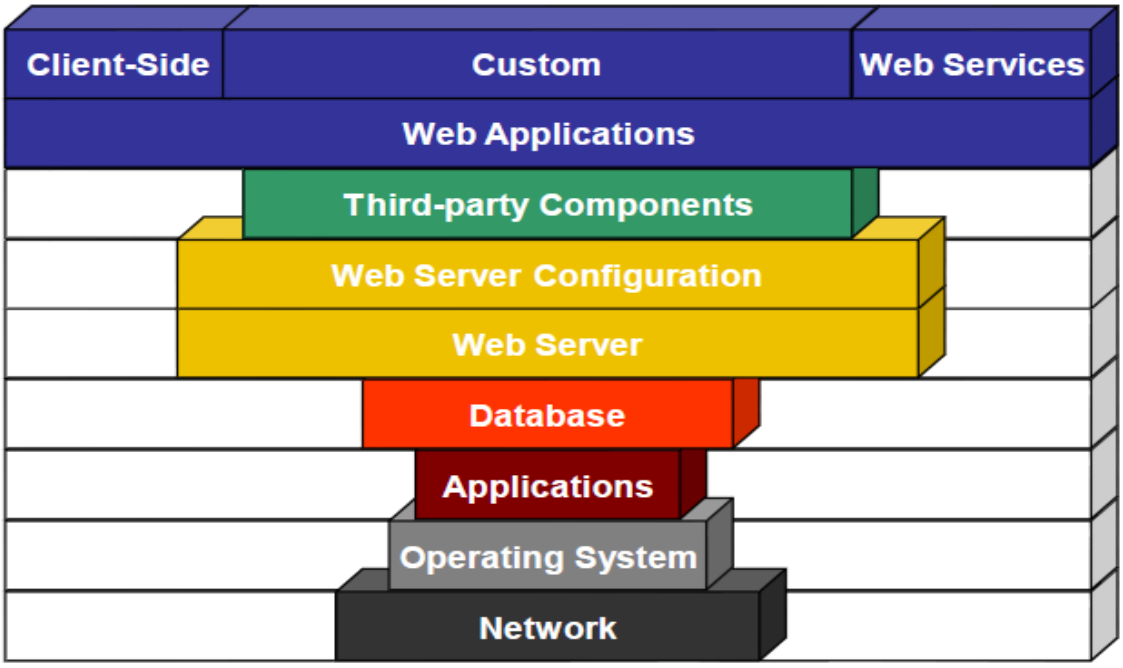
Βάσεις Δεδομένων (Για αποθήκευση των δεδομένων που θα χρειαστούν για την σωστή λειτουργία του πληροφοριακού συστήματος)

Λογισμικό (Το σύνολο του λογισμικού που απαιτείται για να εκπληρώνει το σύστημα τις λειτουργίες του)

Υλικός Εξοπλισμός (Περιλαμβάνει το σύνολο του υλικού εξοπλισμού όπως Η/Υ,εκτυπωτές,μηχανήματα κ.τ.λ.)

Δίκτυο (Η συνδεσμολογία και οι παράμετροι της για την επικοινωνία των επιμέρους τμημάτων του συστήματος)

Ένα Web πληροφοριακό σύστημα είναι ένα πληροφοριακό σύστημα το οποίο βασίζεται σε Web τεχνολογίες. Οι τεχνολογίες αυτές μπορούν να χρησιμοποιηθούν στο κομμάτι της αλληλεπίδρασης του χρήστη με το σύστημα (front-end) π.χ. μια εφαρμογή είναι διαθέσιμη στον Ιστό (ή σε κάποιο Intranet) μέσω ενός φυλλομετρητή. Ακόμα επιτρέπουν την εύκολη χρήση και συντήρηση εξατομικευμένης πρόσβασης για τον τελικό χρήστη (end-user access). Οι Web τεχνολογίες μπορούν να χρησιμοποιηθούν και στο κομμάτι της εσωτερικής λειτουργίας (back-end) του Web πληροφοριακού συστήματος. Οργανώνουν και συνδέουν τα δεδομένα εντός του συστήματος, χρησιμοποιώντας τον παγκόσμιο ιστό σαν παροχέα αυτών των δεδομένων (ή κάποιο Intranet).Τα τμήματα ενός Web πληροφοριακού συστήματος τα οποία θα χρησιμοποιήσουμε στο δεύτερο κεφάλαιο της πτυχιακής εργασίας φαίνονται στο σχήμα 1.1. Όπως γίνεται γρήγορα αντιληπτό κάποια χαρακτηριστικά μεταξύ πληροφοριακού συστήματος και Web πληροφοριακού συστήματος είναι κοινά, όμως ακόμα και σε αυτά ο τρόπος σχεδίασης και η λειτουργικότητα τους διαφέρει.



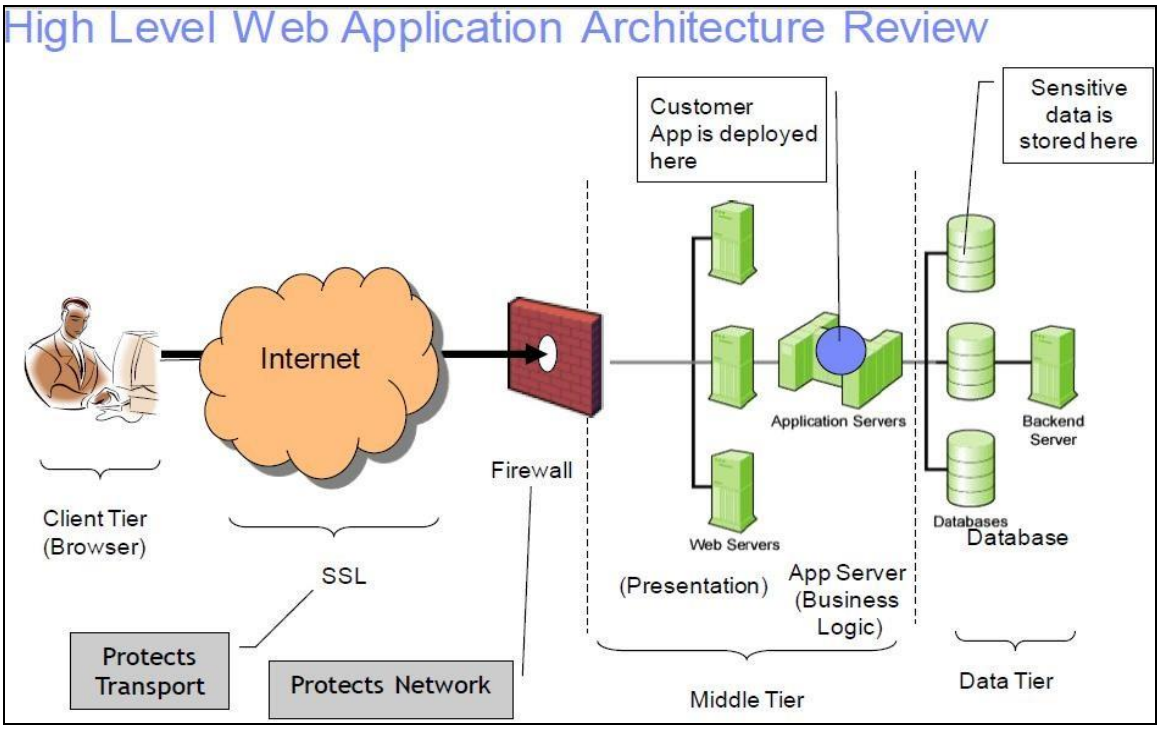
Σχήμα 1.1 Τμήματα ενός Web Πληροφοριακού Συστήματος

Ακολουθεί μια σύντομη περιγραφή των τμημάτων ενός Web πληροφοριακού συστήματος όπως αυτά παρουσιάζονται στο σχήμα 1.1.

- **Δίκτυο(Network):** Το δίκτυο περιλαμβάνει όλες τις λειτουργίες που σχετίζονται με την ορθή, ασφαλή και αμφίδρομη μεταφορά δεδομένων μεταξύ του πελάτη και του συστήματος ή μεταξύ τμημάτων του συστήματος.
- **Λειτουργικό Σύστημα(Operating System):** Αναφέρεται στο εγκατεστημένο λειτουργικό σύστημα του web πληροφοριακού συστήματος.
- **Εφαρμογές(Applications):** Χρησιμοποιούνται από το σύστημα για να εκτελούν συγκεκριμένες ενέργειες οι οποίες διευκολύνουν της εσωτερικές του λειτουργίες.
- **Βάσεις Δεδομένων(Database):** Περιλαμβάνουν τις βάσεις δεδομένων του συστήματος στις οποίες αποθηκεύονται δεδομένα όπως στοιχεία των πελατών, είσοδοι και έξοδοι των εφαρμογών κ.τ.λ.
- **Εξυπηρετητής ιστού(Web Server):** Το πρόγραμμα το οποίο είναι υπεύθυνο για την αποστολή περιεχομένου μέσω του HTTP (Hypertext Transfer Protocol) όπως δυναμικές ιστοσελίδες.
- **Παραμετροποίηση του εξυπηρετητή ιστού(Web Server Configuration):** Το τμήμα αυτό ασχολείται με όλες τις παραμέτρους που ορίζουμε στον εξυπηρετητή ιστού ώστε να λειτουργεί σύμφωνα με τις απαιτήσεις του πληροφοριακού συστήματος μας.
- **Τμήματα λογισμικού άλλων κατασκευαστών(Third-party Components):** Σε αυτά εντάσσονται όλα τα τμήματα λογισμικού που χρησιμοποιούνται για τις υπηρεσίες μας αλλά ανήκουν σε άλλους κατασκευαστές. Ένα παράδειγμα είναι το ActiveX που συχνά χρησιμοποιείται σαν τμήμα λογισμικού άλλου κατασκευαστή από εφαρμογές.
- **Διαδικτυακές εφαρμογές(Web Applications):** Πρόκειται για εφαρμογές οι όποιες χρησιμοποιούνται από τους χρήστες μέσω του παγκόσμιου ιστού. Τέτοιες είναι το ηλεκτρονικό ταχυδρομείο, οι σελίδες ηλεκτρονικών αγοραπωλησιών, οι ηλεκτρονικές δημοπρασίες κ.τ.λ.

- **Η πλευρά του πελάτη(Client-Side):** Το τμήμα αυτό αναφέρεται σε απαραίτητες λειτουργίες οι οποίες όμως εκτελούνται στην πλευρά του πελάτη και όχι στο σύστημα.
- **(Custom):** Περιλαμβάνει όλα τα στοιχεία που χρειάζεται να έχει ο πελάτης στην διάθεση του για την διαδραστική επικοινωνία με το Web πληροφοριακό σύστημα.
- **Υπηρεσίες ιστού(Web Services):** Πρόκειται για διεπαφές προγραμματισμού εφαρμογών (API) οι οποίες εκτελούνται μέσω του πρωτοκόλλου HTTP.

Όπως φαίνεται και παρακάτω στο σχήμα 1.2 η αρχιτεκτονική ενός Web πληροφοριακού συστήματος βασίζεται στον χωρισμό του σε επίπεδα. Ουσιαστικά τα συστήματα αυτά σχεδιάζονται με την μορφή n-tier architectures όπου n είναι ο αριθμός των επιπέδων. Το παράδειγμα μας χρησιμοποιεί την αρχιτεκτονική των πολλαπλών επιπέδων.



Σχήμα 1.2 Αρχιτεκτονική Web Πληροφοριακού Συστήματος

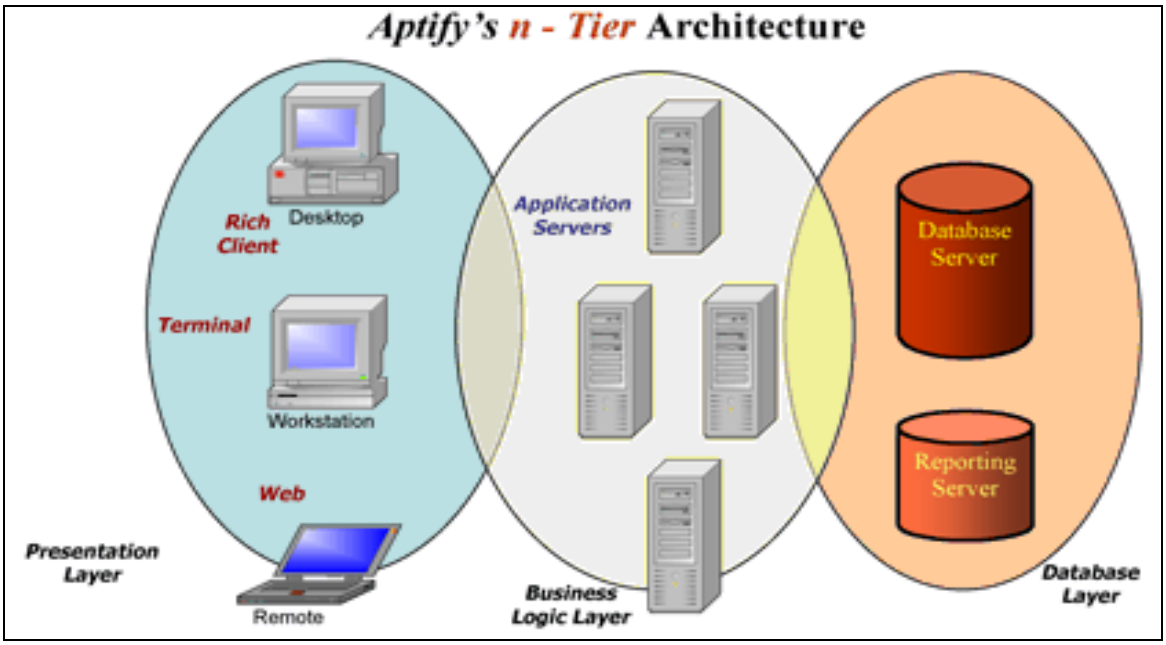
Οι λειτουργίες του κάθε επίπεδου είναι οι εξής:

Το πρώτο επίπεδο αποτελεί το μέρος της παρουσίασης του συστήματος μας στους χρήστες. Σε αυτό προβάλλονται πληροφορίες που έχουν να κάνουν με υπηρεσίες που θέλουμε να παρουσιάσουμε στους χρήστες όπως π.χ. οι λίστες με τα διάφορα προϊόντα , οι τρόποι πληρωμής και το καλάθι αγορών. Αυτό το επίπεδο επικοινωνεί με τα υπόλοιπα στέλνοντας τους τα αποτελέσματα της αλληλεπίδρασης του με τον χρήστη.

Το δεύτερο επίπεδο ονομάζετε μεσαίο, λογικό ή επίπεδο εφαρμογών. Σε αυτό γίνονται όλοι οι υπολογισμοί και οι επεξεργασίες πάνω στα αποτελέσματα που συγκεντρώθηκαν από το πρώτο επίπεδο.

Το τρίτο επίπεδο ή επίπεδο δεδομένων χρησιμοποιείται για την αποθήκευση και την ανάκτηση δεδομένων. Τα δεδομένα που φυλάσσονται σε αυτό παραμένουν ουδέτερα και ανεξάρτητα από το υπόλοιπο σύστημα για λόγους ασφάλειας και απόδοσης. (Eckerson, Wayne, January 1995)

Πιο αφαιρετικά η αρχιτεκτονική προδιαγράφεται παρακάτω.



Σχήμα 1.3 Επίπεδα Αρχιτεκτονικής

Τα τμήματα της εφαρμογής λογισμικού διατάσσονται με τρόπο που όλες οι εργασίες των τριών επιπέδων (παρουσίασης, διαχείρισης δεδομένων και επιχειρησιακής λογικής) εκτελούνται σε έναν ή περισσότερους για κάθε κατηγορία εξυπηρετητές. Ο εξυπηρετητής παρουσίασης δεν είναι παρά ένας εξυπηρετητής ιστού (web server). Ο πελάτης δεν απαιτείται να διαθέτει κανένα τμήμα της εφαρμογής παρά μόνο τη δυνατότητα επικοινωνίας με τον εξυπηρετητή ιστού, δηλαδή μια δικτυακή σύνδεση και ένα πρόγραμμα πλοήγησης στο διαδίκτυο το οποίο ονομάζεται φυλλομετρητής (browser). Για τον λόγο αυτό ο πελάτης αναφέρεται και ως “web client”. Τα προβλήματα ανάγκης συντήρησης των συστημάτων των πελατών εκμηδενίζονται, γεννώνται όμως άλλα, αυτά της ταχύτητας και της ασφάλειας των δικτυακών συνδέσεων.(Βεσκούκης 2000)

1.2 Ευπάθειες των Web πληροφοριακών συστημάτων

Η ενασχόληση με τις ευπάθειες των Web πληροφοριακών συστημάτων θα πρέπει να γίνει από μια παγκόσμια σκοπιά. Επειδή ως αντικείμενο υπάρχουν εδώ και πολλά χρόνια η εισαγωγή μας στις ευπάθειες θα ξεκινήσει με την ανάλυση αυτών που έχει επιλέξει ως συχνότερες το OWASP(Open Web Application Security Project). Το πρόγραμμα OWASP ιδρύθηκε στις Ηνωμένες Πολιτείες της Αμερικής και αποτελεί μια παγκόσμια κοινότητα η οποία ασχολείται με την έρευνα πάνω στις ευπάθειες των διαδικτυακών εφαρμογών. Σκοπός του είναι παρέχει τη γνώση και τα εργαλεία σε οργανισμούς δίνοντας τους την ικανότητα να δημιουργήσουν και να συντηρήσουν ασφαλή συστήματα και εφαρμογές.

Διακρίνουμε τις συχνότερα εμφανιζόμενες ευπάθειες των Web πληροφοριακών συστημάτων σύμφωνα με το OWASP σε δύο κατηγορίες. Αυτός ο διαχωρισμός κάνει ευκολότερο το έργο της δημιουργίας ενός πλαισίου προσδιορισμού των ευπαθειών. Από την μια πλευρά έχουμε τις ευπάθειες που αφορούν την δικτύωση του Web πληροφοριακού συστήματος. Τέτοιες ευπάθειες εμφανίζονται στην επικοινωνία μεταξύ των χρηστών και του συστήματος κυρίως σε επίπεδο TCP/IP. Παραδείγματα αυτής της κατηγορίας είναι η μη ασφαλής ανταλλαγή ευαίσθητων δεδομένων με κρυπτογράφηση της δικτυακής επικοινωνίας

και η λάθος ή μη ορθή χρήση μεθόδων κρυπτογράφησης για τα δεδομένα αυτά. Η δεύτερη κατηγορία αφορά τις ευπάθειες των λειτουργικών οντοτήτων. Δηλαδή ευπάθειες στους εξυπηρετητές Web, στις εφαρμογές και στις βάσεις δεδομένων. Σε αυτή την κατηγορία εντάσσονται ευπάθειες τύπου XSS(Cross Site Scripting), Injections και άλλες.

1.2.1 Ευπάθειες επικοινωνίας συστήματος-χρήστη

Η πρώτη κατηγορία ευπαθειών περιλαμβάνει αυτές που αφορούν τον τρόπο επικοινωνίας μεταξύ των χρηστών και του συστήματος. Επειδή τα δεδομένα που κινούνται μέσω διαδικτύου είναι ευκόλως προσβάσιμα στον οποιονδήποτε, από πολύ νωρίς προέκυψαν ευπάθειες που αφορούσαν την μεταφορά τους. Χρησιμοποιώντας μεθόδους κρυπτογράφησης οι μηχανικοί πληροφορικής προσπαθούν να παρέχουν λύσεις στις ευπάθειες αυτής της κατηγορίας. Οι ευπάθειες που θα αναλυθούν περαιτέρω είναι (α) η μη ασφαλής κρυπτογράφηση δεδομένων και (β) η μη ασφαλής επικοινωνία συστήματος-χρήστη.

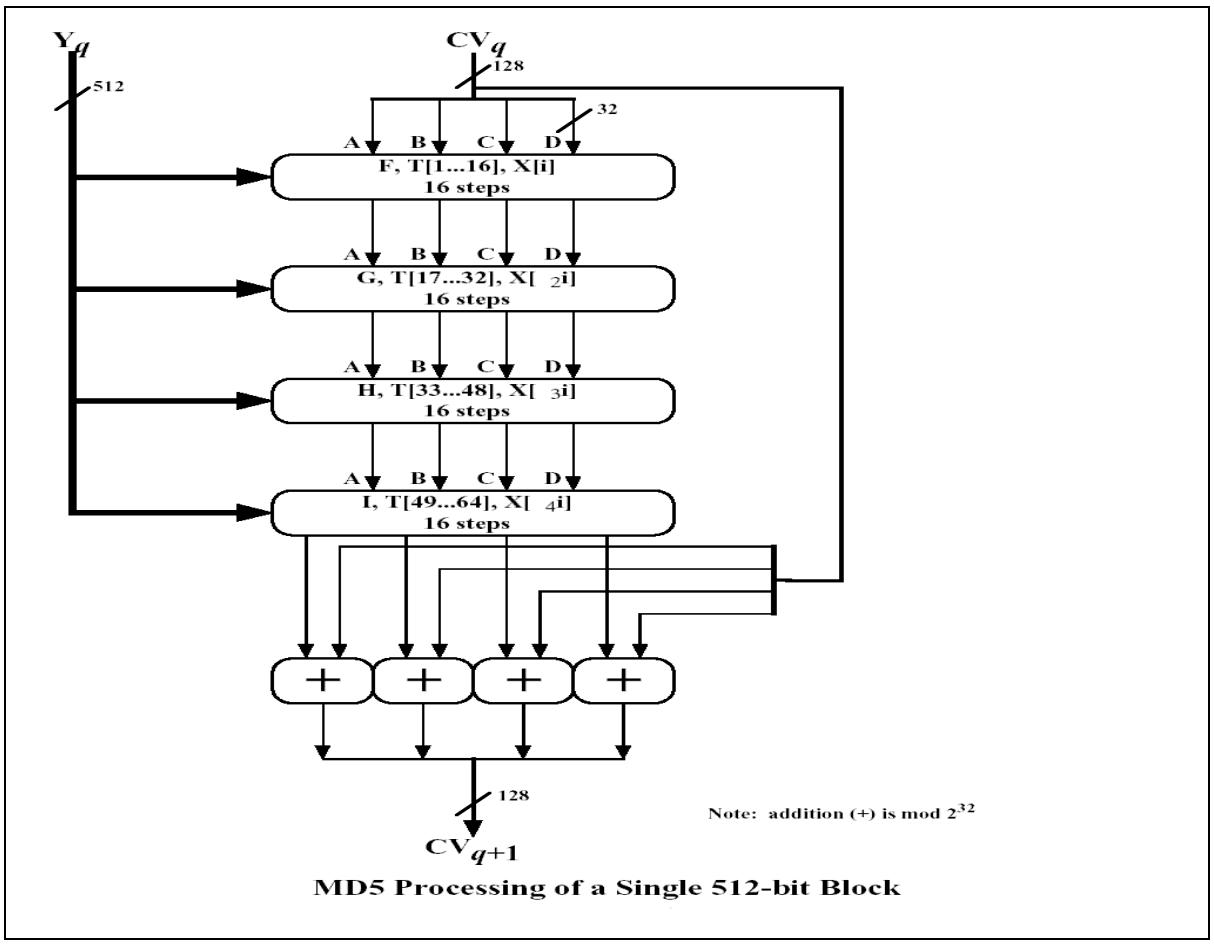
1.2.1.1 Μη ασφαλής κρυπτογράφηση δεδομένων

Με τον όρο μη ασφαλής κρυπτογράφηση δεδομένων εννοούμε την αδυναμία προστασίας ευαίσθητων προσωπικών δεδομένων με χρήση κρυπτογραφίας. Η κρυπτογραφία αποτελεί κύριο συστατικό της προστασίας των εφαρμογών ενός πληροφοριακού συστήματος σχεδιασμένου να λειτουργεί στο διαδίκτυο. Στις μέρες μας είναι πολύ συχνή η αδυναμία της κρυπτογράφησης των ευαίσθητων δεδομένων. Αυτό συμβαίνει για δύο κύριους λόγους. Είτε λόγω ελλιπούς κρυπτογράφησης που χρησιμοποιεί αδύναμους ή ξεπερασμένους αλγόριθμους όπως οι MD5, SHA-1, RC3, RC4 είτε λόγω λανθασμένης χρήσης ισχυρών αλγορίθμων. Αυτά τα λάθη μπορούν να οδηγήσουν σε κοινοποίηση των προσωπικών δεδομένων. (Michael Howard et al, 2010)

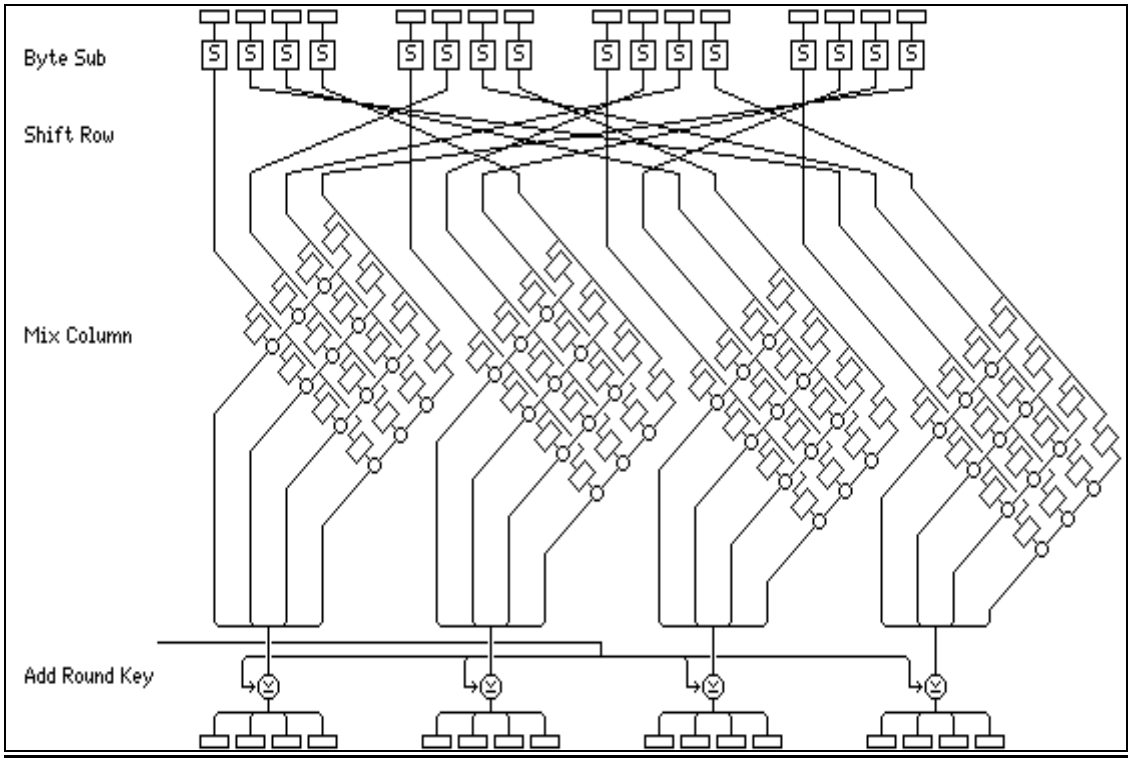
Οι κυριότερες αιτίες αυτής της ευπάθειας είναι οι εξής:

- Απουσία κρυπτογράφησης ευαίσθητων δεδομένων
- Χρήση προσωπικών αλγορίθμων που δεν έχουν πιστοποιηθεί
- Αλόγιστη χρήση δυνατών αλγορίθμων κρυπτογράφησης (AES)
- Χρήση αδύναμων αλγορίθμων των οποίον οι αδυναμίες έχουν επαληθευτεί
- Αποθήκευση κλειδιών σε μη προστατευμένες περιοχές

Παρακάτω στο σχήμα 1.4 βλέπουμε τα βήματα της κρυπτογράφησης ενός μπλοκ δεδομένων μήκους 512 kb με τον πεπαλαιωμένο αλγόριθμο MD5 και στο σχήμα 1.5 τον ισχυρό αλγόριθμο AES του οποίου η αλόγιστη χρήστη αποτελεί επίσης ευπάθεια.



Σχήμα 1.4 MD5



Σχήμα 1.5 AES (Advance Encryption Standard)

1.2.1.2 Μη ασφαλής επικοινωνία συστήματος-χρήστη

Η ευπάθεια της μη ασφαλούς επικοινωνίας μεταξύ συστήματος και χρήστη αναφέρεται στη συχνή αποτυχία κρυπτογράφησης της δικτυακής κίνησης όταν μεταφέρονται ευαίσθητα προσωπικά δεδομένα. Η κρυπτογράφηση (κυρίως με χρήση του πρωτοκόλλου SSL) πρέπει να πραγματοποιείται για κάθε αυθεντικοποιημένη σύνδεση, ειδικά σε ότι έχει να κάνει με το διαδίκτυο. Σε αντίθετη περίπτωση οι εφαρμογές του συστήματος μπορεί να εκθέσουν tokens αυθεντικοποίησης ή συνόδου. Επίσης η κρυπτογράφηση πρέπει να χρησιμοποιείται όποτε στην μεταφορά περιλαμβάνονται δεδομένα πιστωτικών καρτών ή πληροφοριών υγείας, που θεωρούνται αυστηρώς προσωπικά. Τα συστήματα που μένουν πίσω σε επίπεδο κρυπτογράφησης διατρέχουν υψηλό κίνδυνο από κακόβουλους χρήστες. Οι προδιαγραφές της PCI[1] απαιτούν την

[1] Payment Card Industry https://www.pcisecuritystandards.org/

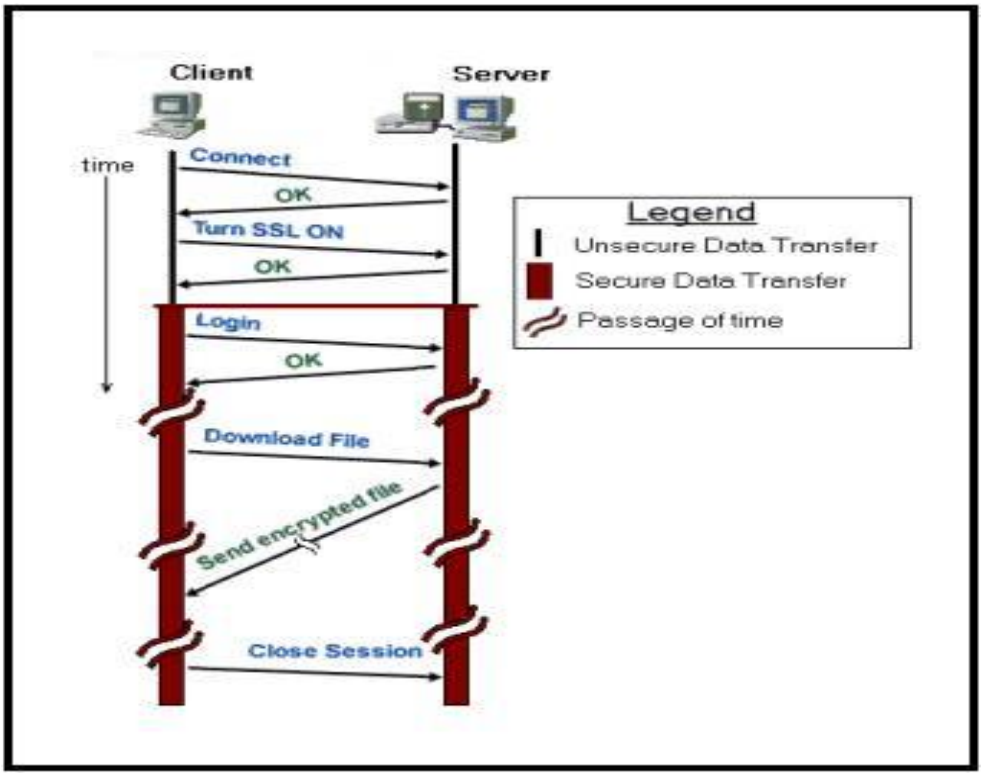
κρυπτογράφηση όλων των δεδομένων που περιέχουν πληροφορίες πιστωτικών καρτών τα οποία μεταφέρονται μέσω του διαδικτύου.

Σε αυτήν την ευπάθεια ένας κακόβουλος χρήστης που παρακολουθεί το κανάλι μετάδοσης, μπορεί να αντλήσει σημαντικές πληροφορίες, εφόσον τα δεδομένα δεν έχουν κρυπτογραφηθεί. Βεβαίως από κανάλι σε κανάλι παρατηρείται διαφορά στο πόσο εύκολα μπορεί κανείς να παρατηρεί τα δεδομένα που μεταφέρονται.

Η χρήση του πρωτοκόλλου SSL για επικοινωνία με τους τελικούς χρήστες είναι απαραίτητη ενέργεια. Αυτό συμβαίνει επειδή είναι σχεδόν σίγουρο πως οι χρήστες θα χρησιμοποιούν ένα μη ασφαλές κανάλι μετάδοσης. Επειδή το πρωτόκολλο HTTP σε κάθε αίτηση του περιέχει πιστοποιητικά αυθεντικοποίησης ή ένα token συνεδρίας, όλη η κυκλοφορία πρέπει να περνάει μέσω SSL και όχι απλώς μέσω μιας αίτησης πιστοποίησης.

Γενικότερα η άρνηση της χρήσης SSL αποτελεί μεγάλο ρίσκο στην μεταφορά δεδομένων που περιέχουν προσωπικά στοιχεία. (OWASP, 2007)

Στο σχήμα 1.6 βλέπουμε το πρωτόκολλο SSL ανάμεσα σε έναν χρήστη και σε ένα σύστημα.



Σχήμα 1.6 SSL (Secure Sockets Layer)

1.2.2 Ευπάθειες των λειτουργικών οντοτήτων

Η δεύτερη κατηγορία ευπαθειών που συναντάται ευρέως σε Web πληροφοριακά συστήματα είναι αυτή που αφορά της λειτουργικές τους οντότητες. Ως λειτουργικές οντότητες ονομάζουμε τις εφαρμογές του συστήματος, τις βάσεις δεδομένων, τους εξυπηρετητές ιστού και γενικότερα όλες τις οντότητες που χρησιμοποιούνται για την ορθή και ουσιαστική λειτουργία του συστήματος. Τέτοιες ευπάθειες χαρακτηρίζονται από το μέρος του συστήματος που αφήνουν εκτεθειμένο. Έτσι έχουμε τις ευπάθειες έκχυσης (injection flaws) που ένα μεγάλο μέρος τους σχετίζεται με τις βάσεις δεδομένων του συστήματος, τις ευπάθειες XSS οι οποίες δημιουργούνται όταν κακόβουλοι χρήστες εκμεταλλεύονται τα cookies και τα session tokens της αλληλεπίδρασης των χρηστών με το σύστημα μας και άλλες.

Οι ευπάθειες που θα αναλυθούν παρακάτω είναι : (α) η αποτυχία του περιορισμού της πρόσβασης μέσο URL, (β) η ευπάθεια Cross Site Scripting (XSS), (γ) η παράνομη πιστοποίηση και διαχείριση συνδέσεων, (δ) η διαρροή πληροφοριών και μη σωστός χειρισμός λαθών, (ε) η πλαστογραφία αίτησης Cross Site (CSRF), (στ) η μη ασφαλής άμεση αναφορά σε αντικείμενο, (ζ) η εκτέλεση επιβλαβούς αρχείου και (ι) τα Injection Flaws.

1.2.2.1 Αποτυχία του περιορισμού της πρόσβασης μέσο URL

Η αποτυχία του περιορισμού της πρόσβασης μέσο URL(Uniform Resource Locator), αναφέρεται στο URL της εκάστοτε εφαρμογής του πληροφοριακού μας συστήματος. Συχνά, η μόνη προστασία για ένα URL είναι ότι οι συνδέσεις με εκείνη την σελίδα δεν παρουσιάζονται στους μη εξουσιοδοτημένους χρήστες. Ωστόσο, ένας κακόβουλος χρήστης μπορεί να είναι σε θέση να βρει και να έχει πρόσβαση σε αυτές τις σελίδες, να χρησιμοποιήσει τις λειτουργίες, και να έχει πρόσβαση στα δεδομένα τους. Έλεγχοι πρόσβασης πρέπει να εκτελούνται προτού χορηγηθεί ένα αίτημα πρόσβασης σε μια ευαίσθητη λειτουργία,

εξασφαλίζοντας ότι ο χρήστης έχει εξουσιοδοτηθεί για να έχει πρόσβαση σε εκείνη την λειτουργία.

Η μέθοδος επίθεσης για αυτήν την ευπάθεια καλείται «forced browsing». Πρόκειται για ένα είδος επίθεσης με τεχνική ωμής βίας(brute force attack) στην οποία ο επιτιθέμενος δοκιμάζει διευθύνσεις που υποψιάζεται πως είναι απροστάτευτες. Το μεγάλο μέγεθος μιας εφαρμογής έχει ως συνέπεια ένα σύνθετο πρότυπο που είναι δύσκολο να κατανοηθεί από τους υπεύθυνους για την ανάπτυξη και τους ειδικούς ασφάλειας, ώστε να υλοποιήσουν τον κώδικα έλεγχου πρόσβασης για κάθε τμήμα της. Αυτή η πολυπλοκότητα καθιστά πιθανή την εμφάνιση λαθών και ως αποτέλεσμα έχει το να αφήνονται τμήματα της εφαρμογής εκτεθειμένα.

Μερικά κοινά παραδείγματα αυτής της ευπάθειας είναι:

- Τα «κρυμμένα» ή «ειδικά» URL, που δίνονται μόνο στους διαχειριστές ή σε εξουσιοδοτημένους χρήστες στο επίπεδο παρουσίασης, αλλά είναι προσβάσιμα από όλους τους χρήστες εάν γνωρίζουν ότι υπάρχουν.
- Οι εφαρμογές επιτρέπουν συχνά την πρόσβαση σε «κρυμμένα» αρχεία, όπως τα στατικά XML ή παραγόμενες αναφορές από το σύστημα.
- Κώδικας που επιβάλλει μια πολιτική ελέγχου πρόσβασης αλλά είναι ξεπερασμένος ή ανεπαρκής.
- Κώδικας που αξιολογεί τα προνόμια στον πελάτη αλλά όχι στον κεντρικό υπολογιστή. (OWASP, 2007)

1.2.2.2 Cross-Site Scripting (XSS)

Η τεχνική Cross-Site Scripting (XSS) γνώρισε μεγάλη διάδοση με την έκρηξη του web 2.0 και την κατασκευή σελίδων με δυναμικό περιεχόμενο τύπου JavaScript οι οποίες εκτελούν σειρά εντολών (scripts) στον υπολογιστή μας αλλά και στον εξυπηρετητή ιστού στον οποίο φιλοξενούνται οι ιστοσελίδες, με αποτέλεσμα η τρύπα ασφαλείας να δίνει την δυνατότητα εκτέλεσης κώδικα στο

απομακρυσμένο σύστημα, αλλά και στα συστήματα που θα συνδεθούν σε αυτό. Στην ουσία η τεχνική αυτή είναι ένα είδος HTML injection.Πρόκειται για την επικρατέστερη και πιο ολέθρια ευαισθησία των διαδικτυακών εφαρμογών. Μια επίθεση XSS έχει ως στόχο τα προσωπικά δεδομένα των χρηστών ενός συστήματος και μπορεί να οδηγήσει σε ολική παραβίαση της ασφάλειας, όταν πληροφορίες χρηστών έχουν κλαπεί. Οι περισσότερες επιθέσεις περιλαμβάνουν 2 μέρη: είτε τον επιτιθέμενο και το σύστημα είτε τον επιτιθέμενο και τον χρήστη. Αντιθέτως η επίθεση XSS περιλαμβάνει 3 μέρη: τον επιτιθέμενο, τον χρήστη και το σύστημα. Ο στόχος της επίθεσης XSS είναι να κλέψει τα cookies του χρήστη ή οποιωνδήποτε άλλων ευαίσθητων πληροφοριών που μπορούν να προσδιορίσουν τον χρήστη στο Web πληροφοριακό σύστημα. Με το token του νόμιμου χρήστη, ο εισβολέας μπορεί να προχωρήσει σε αλληλεπίδραση με το σύστημα μιμούμενος τον νόμιμο χρήστη. (Klein, March 2008)

Ένας επιτιθέμενος μπορεί να χρησιμοποιήσει το cross site scripting για να στείλει ένα κακόβουλο script σε κάποιον χρήστη. Ο φυλλομετρητής του χρήστη δεν έχει κανένα τρόπο να ξέρει ότι το script δεν προήλθε από μια εμπιστευτική πηγή, και θα το εκτελέσει. Επειδή θεωρεί ότι το script προήλθε από μια έμπιστη πηγή, το κακόβουλο script μπορεί να έχει πρόσβαση σε οποιαδήποτε cookies , στα tokens συνεδρίας, ή άλλες ευαίσθητες πληροφορίες που αποθηκεύονται από τον φυλλομετρητή και που χρησιμοποιούνται σε εκείνη τη σελίδα. Αυτά τα script μπορούν ακόμη και να ξαναγράψουν το περιεχόμενο της σελίδας HTML.

Οι επιθέσεις XSS μπορούν γενικά να ταξινομηθούν σε δύο κατηγορίες: Στις stored και στις reflected. Οι stored επιθέσεις είναι εκείνες όπου το κακόβουλο script αποθηκεύεται μόνιμα στους διακομιστές που αποτελούν στόχο, π.χ. σε μια βάση δεδομένων, σε ένα φόρουμ μηνυμάτων, ένα αρχείο καταγραφής(log file) επισκέψεων, ένα πεδίο για σχόλια, κλπ. Το θύμα ανακτά έπειτα το κακόβουλο script από τον διακομιστή όταν ζητά τις αποθηκευμένες πληροφορίες. Οι reflected επιθέσεις είναι εκείνες όπου το κακόβουλο script απεικονίζεται από τον εξυπηρετητή ιστού σε ένα μήνυμα λάθους, στο αποτέλεσμα μιας αναζήτησης, ή σε οποιαδήποτε άλλη απάντηση που περιλαμβάνει κάποιο μέρος ή ολόκληρη την είσοδο που στέλνεται στον διακομιστή ως τμήμα του αιτήματος. Οι reflected επιθέσεις γίνονται στα θύματα μέσω μιας άλλης διαδρομής, όπως μέσο ενός μηνύματος ηλεκτρονικού ταχυδρομείου, ή μέσο κάποιου άλλου εξυπηρετητή του

δικτύου. Όταν ένας χρήστης πέφτει θύμα απάτης κάνοντας κλικ σε κάποιον κακόβουλο σύνδεσμο (link) ή υποβάλλει στοιχεία σε κάποια ειδικά επεξεργασμένη φόρμα, το κακόβουλο script περνάει στον εξυπηρετητή ιστού που αποτελεί στόχο, ο οποίος στέλνει την επίθεση πίσω στον φυλλομετρητή του χρήστη. Ο φυλλομετρητής εκτελεί έπειτα το script επειδή πιστεύει ότι προήλθε από μια εμπιστευτική πηγή.

Η συνέπεια μιας επίθεσης XSS είναι η ίδια ανεξάρτητα από εάν πρόκειται για stored ή reflected επίθεση. Η διαφορά είναι στο πώς η ζημιά που προκαλείται φθάνει στον διακομιστή. Μην επαναπαύεστε με τη λογική ότι κάποια σελίδα είναι «read only» ή «brochure-ware» και συνεπώς δεν είναι τρωτή σε σοβαρές reflected επιθέσεις XSS. Το XSS μπορεί να προκαλέσει ποικίλα προβλήματα για τον χρήστη που κυμαίνονται από μια απλή ενόχληση έως το να χάσει τον ίδιο τον λογαριασμό του. Οι πιο μεγάλες επιθέσεις XSS μπορούν να προκαλέσουν τη δημοσίευση των cookies συνεδρίας του χρήστη, που η δημοσίευση αυτή επιτρέπει σε έναν επιτιθέμενο να καταλάβει τη σύνδεση του χρήστη και μαζί με αυτή και τον λογαριασμό του. Άλλες καταστρεπτικές επιθέσεις περιλαμβάνουν την δημοσίευση των αρχείων των χρηστών, την εγκατάσταση Trojan horses, που πηγαίνουν αυτόματα τον χρήστη σε κάποια άλλη σελίδα (redirect), ή που τροποποιούν το περιεχόμενο του δικτυακού τόπου. Μια διαρροή ασφαλείας, ευάλωτη σε XSS, που επιτρέπει σε έναν επιτιθέμενο να τροποποιήσει ένα δελτίο τύπου ή κάποια ιστοσελίδα με ειδήσεις, θα μπορούσε να έχει επιπτώσεις στην εικόνα μιας επιχείρησης και να μειώσει την εμπιστοσύνη των πελατών της σε αυτήν. Μια διαρροή ασφαλείας, ευάλωτη σε XSS, σε μια φαρμακευτική εφαρμογή θα μπορούσε να επιτρέψει σε έναν επιτιθέμενο να τροποποιήσει τις πληροφορίες της δόσης του φαρμάκου με συνέπεια να δοθεί υπερβολική δόση σε κάποιον ασθενή.

Οι επιτιθέμενοι χρησιμοποιούν συχνά διάφορες μεθόδους για να κωδικοποιήσουν το κακόβουλο τμήμα του script, όπως η χρήση Unicode, έτσι ώστε το αίτημα να φαίνεται λιγότερο ύποπτο στον χρήστη. Υπάρχουν εκατοντάδες παραλλαγές αυτών των επιθέσεων, συμπεριλαμβανομένων και εκδόσεων που δεν απαιτούν σύμβολα του τύπου < >. Για αυτόν τον λόγο, το να προσπαθήσουμε να φιλτράρουμε αυτά τα script δεν σημαίνει ότι σίγουρα θα πετύχει σαν μέθοδος άμυνας. Αντί αυτού προτείνεται να τσεκάρουμε την εγκυρότητα της εισόδου με κάποια πολύ αυστηρά κριτήρια, που σχετίζονται με το είδος της εισόδου που

αναμένουμε. Οι επιθέσεις με XSS παρουσιάζονται συνεχώς με τη μορφή εμφωλευμένου JavaScript κώδικα. Παρόλα αυτά, οποιοδήποτε ενσωματωμένο active content είναι μια πιθανή πηγή κινδύνου, που περιλαμβάνει: ActiveX (OLE), VBScript, Shockwave, Flash και περισσότερα.

Τα προβλήματα που προκύπτουν από XSS επιθέσεις μπορούν επίσης να δημιουργηθούν σε εξυπηρετητές ιστού και εφαρμογών. Οι περισσότεροι εξυπηρετητές ιστού και εφαρμογών παράγουν απλές ιστοσελίδες που βγάζουν κάποια μηνύματα στην περίπτωση λάθους, όπως 404 ‘page not found’ ή 500 ‘internal server error’. Εάν αυτές οι σελίδες περιέχουν οποιεσδήποτε πληροφορίες από το αίτημα του χρήστη, όπως το URL στο οποίο προσπαθούσαν να έχουν πρόσβαση, μπορούν να είναι ευάλωτες σε μια reflected επίθεση XSS.

Η πιθανότητα κάποια διαδικτυακή εφαρμογή να είναι ευάλωτη σε XSS είναι εξαιρετικά υψηλή. Υπάρχουν πολλοί τρόποι να προσβληθούν από τέτοιες επιθέσεις δικτυακές εφαρμογές, στέλνοντάς τους κακόβουλα script. Οι προγραμματιστές, προσπαθώντας να φιλτράρουν κακόβουλα τμήματα αυτών των αιτημάτων είναι πολύ πιθανό να αγνοήσουν άλλες πιθανές επιθέσεις ή ευπάθειες. Η εύρεση αυτών των διαρροών ασφαλείας δεν είναι παρά πολύ δύσκολη για τους επιτιθεμένους, καθώς το μόνο που χρειάζονται είναι ένας φυλλομετρητής και λίγος χρόνος. Υπάρχουν πολλά δωρεάν εργαλεία διαθέσιμα στο δίκτυο που βοηθούν τους επιτιθέμενους να βρουν αυτές τις διαρροές καθώς επίσης και τον τρόπο που θα μπορέσουν να κάνουν τις επιθέσεις XSS σε κάποια διαδικτυακή εφαρμογή στόχο. (Fogie et al , 2007)

1.2.2.3 Παράνομη πιστοποίηση και διαχείριση των συνδέσεων

Η πιστοποίηση και η διαχείριση των συνδέσεων περιλαμβάνει όλες τις μορφές χειρισμού πιστοποίησης του χρήστη και διαχείρισης ενεργών συνδέσεων. Η πιστοποίηση είναι μια κρίσιμη πλευρά αυτής της διαδικασίας, αλλά ακόμη και αξιόπιστοι μηχανισμοί πιστοποίησης είναι δυνατό να υπονομευθούν από λειτουργίες διαχείρισης πιστοποιητικών με διαρροές ασφάλειας, συμπεριλαμβανομένης της “αλλαγής κωδικού πρόσβασης”, “υπενθύμισης κωδικού πρόσβασης”, “αποθήκευσης κωδικού πρόσβασης”, “ενημέρωσης λογαριασμού”,

και άλλων σχετικών λειτουργιών. Όλες οι λειτουργίες διαχείρισης λογαριασμών θα πρέπει να ζητούν επαναπιστοποίηση, ακόμη και αν ο χρήστης έχει έγκυρο id σύνδεσης.

Η πιστοποίηση χρήστη στο διαδίκτυο εμπεριέχει συνήθως τη χρήση μιας ταυτότητας χρήστη(user id) και ενός συνθηματικού(password). Ισχυρότερες μέθοδοι πιστοποίησης είναι διαθέσιμες στο εμπόριο, όπως κρυπτογραφικές συσκευές ή βιομετρικοί έλεγχοι βασισμένοι σε λογισμικό και υλικό, αλλά τέτοιοι μηχανισμοί είναι απαγορευτικοί λόγω κόστους για τις περισσότερα Web πληροφοριακά συστήματα. Οι προγραμματιστές συχνά υποτιμούν την πολυπλοκότητα του σχεδιασμού ενός σχεδίου πιστοποίησης και διαχείρισης συνδέσεων, το οποίο να προστατεύει επαρκώς τα δεδομένα σε κάθε τμήμα του συστήματος. Οι δικτυακές εφαρμογές πρέπει να εγκαθιστούν συνδέσεις, ώστε να εντοπίζουν τη ροή των αιτημάτων από κάθε χρήστη. Το HTTP δεν παρέχει αυτή τη δυνατότητα, έτσι οι δικτυακές εφαρμογές πρέπει να τις δημιουργούν οι ίδιες. Συχνά, το περιβάλλον των δικτυακών εφαρμογών παρέχει δυνατότητα συνδέσεων, αλλά πολλοί προγραμματιστές προτιμούν να δημιουργούν τις δικές τους συνδέσεις από την αρχή. Σε κάθε περίπτωση, αν οι συνδέσεις δεν προστατεύονται επαρκώς, ένας επιτιθέμενος μπορεί να πάρει τον έλεγχο μιας ενεργής σύνδεσης και να αποσπάσει την ταυτότητα χρήστη. Η δημιουργία ενός σχεδίου ώστε να δημιουργηθούν ισχυρές συνδέσεις και να προστατεύονται κατά τη διάρκεια του κύκλου ζωής τους έχει απασχολήσει πολλούς προγραμματιστές.(M. Howard and D. LeBlanc, 2002)

1.2.2.4 Διαρροή πληροφοριών και μη σωστός χειρισμός λαθών

Διαρροή πληροφοριών και μη σωστός χειρισμός λαθών έχουμε όταν από μια εφαρμογή μπορεί ακούσια να διαρρεύσουν πληροφορίες σχετικά με την διαμόρφωσή της, τις εσωτερικές της λειτουργίες ή τον τρόπο παραβίασης του απορρήτου της λόγω προβλημάτων. Από την εφαρμογή μπορεί επίσης να διαρρεύσουν εσωτερικά της χαρακτηριστικά κατά τη διάρκεια του χρόνου που χρειάζεται για να επεξεργαστεί ορισμένες διαδικασίες, ή μέσω των αντιδράσεων σε διαφορετικές εισόδους , όπως η εμφάνιση του ίδιου μηνύματος λάθους σε λάθη με

διαφορετικό αριθμό σφάλματος. Από τις δικτυακές εφαρμογές διαρρέουν επίσης πληροφορίες για την εσωτερική τους κατάσταση κυρίως με τη μορφή λεπτομερών μηνυμάτων λάθους ή μηνυμάτων λάθους αποσφαλμάτωσης. Συχνά αυτές οι πληροφορίες μπορούν να οδηγήσουν σε ισχυρότερες επιθέσεις κατά των υπολοίπων ευαισθησιών. (Michael Howard et al, 2010)

1.2.2.5 Πλαστογραφία αίτησης Cross-Site (CSRF)

Πλαστογραφία αίτησης Cross-Site (CSRF) είναι μία από της νεότερες ευαισθησίες που όμως είναι πολύ απλή και με καταστροφικά αποτελέσματα. Σε μια επίθεση CSRF ο κακόβουλος χρήστης αναγκάζει τον φυλλομετρητή ενός επαληθευμένου από το σύστημα χρήστη να στείλει μια αίτηση σε μια ευάλωτη διαδικτυακή εφαρμογή, για να εκτελέσει μια συγκεκριμένη ενέργεια προσποιούμενος τον χρήστη. Ο κακόβουλος κώδικας δεν είναι συχνά στη σελίδα που δέχτηκε την επίθεση. Γι'αυτό και ονομάζεται "Cross-Site".

Αυτή η ευπάθεια είναι αρκετά διαδεδομένη κυρίως σε συστήματα που :

- Δεν έχουν ελέγχους έγκρισης για ευάλωτες ενέργειες
- Θα επεξεργαστούν μια ενέργεια εάν μια προεπιλεγμένη πιστοποίηση μπορεί να δοθεί στο αίτημα (π.χ. admin/admin ή guest/guest).
- Εξουσιοδοτούν τα αιτήματα βασιζόμενα μόνο στις πιστοποιήσεις που έχουν επιβεβαιωθεί αυτόματα όπως το cookie περιόδου λειτουργίας εάν ο χρήστης έχει ήδη πιστοποιηθεί στο σύστημα , ή η λειτουργία "Αποθήκευσης Ονόματος Χρήστη και Κωδικού Πρόσβασης" εάν δεν έχει ήδη πιστοποιηθεί.

Δυστυχώς σήμερα οι περισσότερες διαδικτυακές εφαρμογές βασίζονται αποκλειστικά σε αυτόματα υποβαλλόμενες πιστοποιήσεις όπως τα cookies συνόδου, βασικά πιστοποιητικά αναγνώρισης, source IP διευθύνσεις , SSL πιστοποιητικά και πιστοποιητικά τομέα των Windows.

Αυτή η ευπάθεια είναι γνωστή και με άλλα ονόματα όπως Session Riding, One Click Attack, Cross-Site Reference Forgery, Hostile Linking και Automation Attack. (Edward W. Felten and William Zeller, 2008)

1.2.2.6 Μη ασφαλής άμεση αναφορά σε αντικείμενο

Η άμεση αναφορά σε αντικείμενο εμφανίζεται όταν ένας προγραμματιστής εκθέτει μια αναφορά σε κάποια εσωτερική εφαρμογή του αντικειμένου, όπως ένα αρχείο, έναν κατάλογο, το αρχείο των βάσεων δεδομένων ή ένα URL. Ένας εισβολέας μπορεί να εκμεταλλευτεί τις άμεσες αναφορές σε ένα αντικείμενο για να αποκτήσει πρόσβαση σε άλλα αντικείμενα χωρίς άδεια, εκτός εάν υπάρχει κάποιος έλεγχος πρόσβασης.

Για παράδειγμα στα συστήματα Internet Banking είναι συχνό να χρησιμοποιείται ο αριθμός λογαριασμού ως πρωτεύον κλειδί. Ως εκ τούτου είναι δελεαστικό το να χρησιμοποιήσει κάποιος απευθείας τον αριθμό λογαριασμού στο web interface. Ακόμα και αν οι προγραμματιστές έχουν προνοήσει να χρησιμοποιήσουν παραμετροποιημένα ερωτήματα SQL για την πρόληψη εκχύσεων SQL (SQL injections), εάν δεν υπάρχει επιπλέον έλεγχος για το αν ο χρήστης είναι ο κάτοχος του λογαριασμού και έχει την άδεια να δει τον λογαριασμό, ένας επιτιθέμενος μπορεί να πειραματιστεί με την παράμετρο του αριθμού λογαριασμού και να δει ή να αλλάξει όλους τους λογαριασμούς.

Αυτός ο τύπος επίθεσης σημειώθηκε στη σελίδα της Αυστραλιανής εφορίας GST το 2000 όπου ένας νόμιμος αλλά κακόβουλος χρήστης απλός άλλαξε τον αριθμό φορολογικού μητρώου στο URL. Με αυτόν τον τρόπο συνέλεξε πληροφορίες για περίπου 17000 εταιρίες από το σύστημα. Αυτό το είδος επίθεσης είναι αρκετά σύνηθες, αλλά κατά ένα μεγάλο μέρος μη δοκιμασμένος στις τρέχουσες εφαρμογές. (OWASP, 2007)

1.2.2.7 Εκτέλεση επιβλαβούς αρχείου

Η εκτέλεση επιβλαβούς αρχείου είναι μια ευαισθησία που εντοπίζεται σε πολλές εφαρμογές. Οι προγραμματιστές συχνά χρησιμοποιούν ή συνδέουν απευθείας, δυνητικά κακόβουλες εισόδους με ρεύματα αρχείων και λειτουργιών ή με εσφαλμένως έμπιστα αρχεία εισόδου. Σε πολλές πλατφόρμες επιτρέπεται η χρήση αναφορών σε εξωτερικά αντικείμενα, όπως URL ή αναφορών στο σύστημα αρχείων. Όταν τα δεδομένα είναι ανεπαρκώς ελεγμένα, αυτό μπορεί να οδηγήσει το αυθαίρετο, απομακρυσμένο και κακόβουλο περιεχόμενο στο να υποβάλετε σε επεξεργασία από τον εξυπηρετητή ιστού.

Αυτό επιτρέπει στους επιτιθέμενους:

- Την απομακρυσμένη εκτέλεση κώδικα
- Την απομακρυσμένη εγκατάσταση root kit και να οδηγήσουν σε πλήρη συμβιβασμό το σύστημα

Η ευαισθησία αυτή είναι εξαιρετικά διαδεδομένη σε PHP και πρέπει να δοθεί ιδιαίτερη προσοχή σε συναρτήσεις ρευμάτων και συναρτήσεις αρχείων ώστε να βεβαιωθούμε ότι είσοδοι που παρέχονται από τον χρήστη δεν επηρεάζουν τα ονόματα των αρχείων.(OWASP, 2007)

1.2.2.8 Injection Flaws

Οι διαρροές ασφάλειας μέσω injection flaws επιτρέπουν στους επιτιθεμένους να στείλουν κακόβουλο κώδικα μέσω μιας διαδικτυακής εφαρμογής σε ένα άλλο σύστημα. Αυτές οι επιθέσεις περιλαμβάνουν κλήσεις στο λειτουργικό σύστημα μέσω των system calls, τη χρήση εξωτερικών προγραμμάτων μέσω των εντολών του shell, καθώς επίσης και κλήσεις σε βάσεις δεδομένων μέσω της SQL (δηλ., έκχυση SQL). Ολόκληρα τα scripts που γράφονται στην perl, python και άλλες γλώσσες μπορούν να γίνουν injected σε κακοσχεδιασμένες διαδικτυακές εφαρμογές και να εκτελεσθούν. Όταν μια διαδικτυακή εφαρμογή χρησιμοποιεί έναν διερμηνέα οποιουδήποτε τύπου υπάρχει ο κίνδυνος μιας επίθεσης με injection.

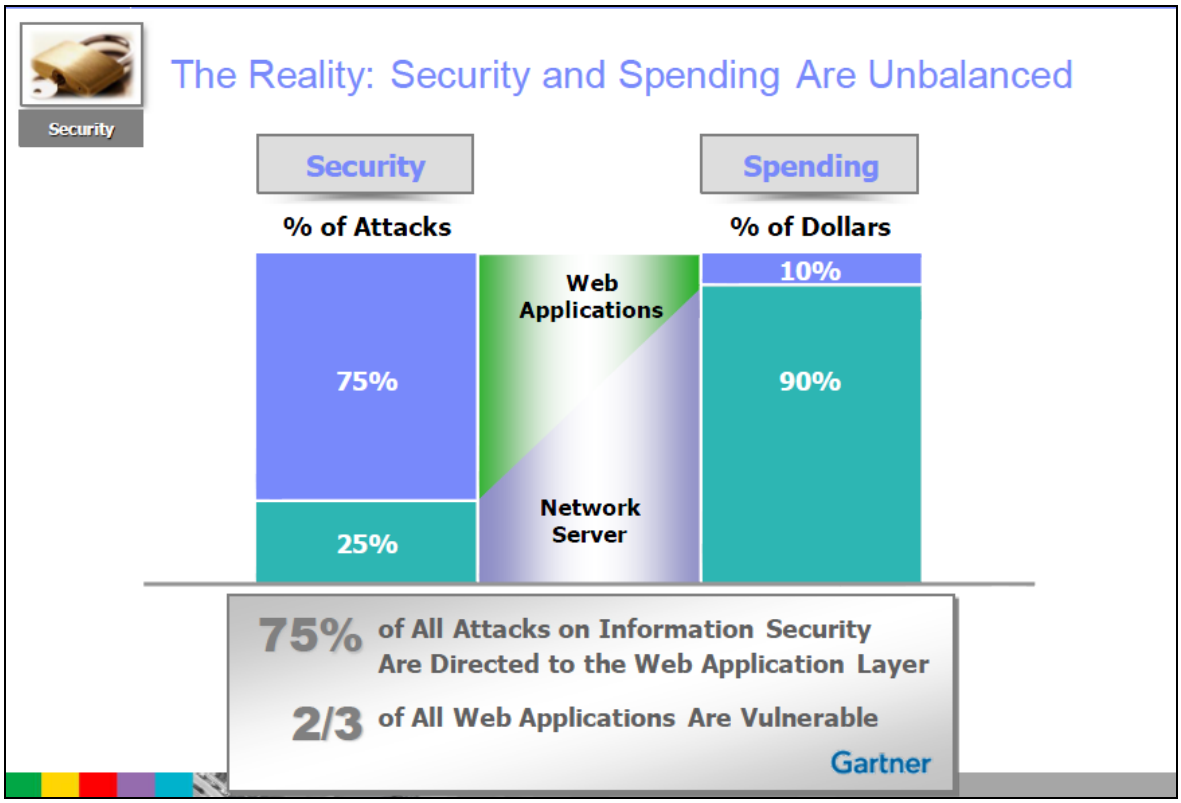
Πολλές διαδικτυακές εφαρμογές χρησιμοποιούν στοιχεία των λειτουργικών συστημάτων και εξωτερικών προγραμμάτων για να εκτελέσουν τις συναρτήσεις τους. Το Sendmail είναι πιθανώς το πιο συχνά κληθέν εξωτερικό πρόγραμμα, αλλά χρησιμοποιούνται επίσης πολλά ακόμη προγράμματα. Όταν μια διαδικτυακή εφαρμογή περνά τις πληροφορίες από ένα αίτημα HTTP κατευθείαν ως τμήμα ενός εξωτερικού αιτήματος, η μετάβαση πρέπει να διεκπεραιωθεί προσεκτικά. Διαφορετικά, ο επιτιθέμενος μπορεί να εμφωλεύσει τους ειδικούς χαρακτήρες (meta), τις κακόβουλες εντολές, ή μεθόδους αλλαγής – τροποποίησης εντολών στις πληροφορίες και η διαδικτυακή εφαρμογή θα τα περάσει τυφλά στο εξωτερικό σύστημα όπου και θα εκτελεστούν. (OWASP, 2007)

Η έγχυση SQL (SQL injection) είναι μια ιδιαίτερα διαδεδομένη και επικίνδυνη μορφή έκχυσης. Για να εκμεταλλευτεί μια διαρροή ασφαλείας μέσω εγχύσεων SQL, ο επιτιθέμενος πρέπει να βρει μια παράμετρο μέσω της οποίας η διαδικτυακή εφαρμογή περνά σε μια βάση δεδομένων. Ο επιτιθέμενος μπορεί να εξαπατήσει τη διαδικτυακή εφαρμογή στέλνοντας μια κακόβουλη ερώτηση στη βάση δεδομένων, ενσωματώνοντας προσεκτικά κακόβουλο κώδικα σε εντολές SQL, μέσα στο περιεχόμενο της παραμέτρου. Αυτές οι επιθέσεις δεν είναι δύσκολο να γίνουν και πολλά εργαλεία έχουν τη δυνατότητα να ανιχνεύσουν διαδικτυακούς τόπους για τέτοιου είδους προβλήματα. Οι συνέπειες είναι ιδιαίτερα καταστρεπτικές, δεδομένου ότι ένας επιτιθέμενος μπορεί να λάβει, να αλλοιώσει, ή να καταστρέψει το περιεχόμενο βάσεων δεδομένων. (Chris Anley, 2002)

Οι επιθέσεις εγχύσεων πολλές φορές είναι πολύ εύκολο να ανακαλυφθούν από κάποιον επιτιθέμενο ο οποίος μπορεί και να τις εκμεταλλευτεί, αλλά μπορούν επίσης να είναι εξαιρετικά δύσκολο να βρεθούν. Οι συνέπειες μπορεί επίσης να είναι εξαιρετικά δριμείες ή ασήμαντες. Σε κάθε περίπτωση, η χρήση των εξωτερικών κλήσεων είναι αρκετά διαδεδομένη, έτσι η πιθανότητα μιας διαδικτυακής εφαρμογής που έχει μια ευπάθεια εγχύσεων εντολών πρέπει να θεωρηθεί υψηλή.

1.3 Σύνοψη

Οι ευπάθειες των Web πληροφοριακών συστημάτων έχουν απασχολήσει τους μηχανικούς πληροφορικής για πολλά χρόνια και θα συνεχίσουν να τους απασχολούν, καθώς αποτελούν πολύ σοβαρή υπόθεση για την ασφαλή λειτουργία των συστημάτων. Στο σχήμα 1.7 παρακάτω βλέπουμε την αναλογία ασφάλειας-εξόδων μιας μέσης εταιρίας ανάπτυξης Web πληροφοριακών συστημάτων. Από το σχήμα φαίνεται πως παρά το μεγάλο ποσοστό ευπαθειών στις διαδικτυακές εφαρμογές, το μεγαλύτερο μέρος των εξόδων γίνεται προς όφελος των εξυπηρετητών δικτύου. Από αυτό συμπεραίνουμε ότι δεν δίνεται η πρέπουσα σημασία στην ασφάλεια των διαδικτυακών εφαρμογών.



Σχήμα 1.7 Αναλογία ασφάλειας-εξόδων της μέσης εταιρίας ανάπτυξης Web πληροφοριακών συστημάτων

Προγράμματα όπως το OWASP και διάφοροι διεθνής οργανισμοί κρούουν τον κώδωνα του κινδύνου στις εταιρίες ανάπτυξης Web πληροφοριακών συστημάτων για την σημαντικότητα της ασφάλειας. Συγκεκριμένα το OWASP κάθε τρία χρόνια ανανεώνει την λίστα με τις δέκα πιο συχνά εμφανιζόμενες ευπάθειες. Η περιγραφή των ευπαθειών που έγινε στο κεφάλαιο αυτό βοηθά τον αναγνώστη

να καταλάβει τους κινδύνους που καλείται να αντιμετωπίσει ένα μηχανικός πληροφορικής ώστε να παρέχει όσο το δυνατό μεγαλύτερο επίπεδο ασφάλειας στο Web πληροφοριακό σύστημα.

ΚΕΦΑΛΑΙΟ 2: Μεθοδολογικό πλαίσιο αξιολόγησης ευπαθειών

Προκειμένου να αξιολογηθούν οι ευπάθειες των Web πληροφοριακών συστημάτων θα πρέπει να προταθεί ένα πλαίσιο προσδιορισμού τους. Στο σχήμα 1.1 είχαμε αναλύσει τα επιμέρους τμήματα ενός τέτοιου συστήματος. Η βασική ιδέα του πλαισίου είναι να κατηγοριοποιεί τις ευπάθειες ανάλογα με το ποια σημεία του συστήματος επηρεάζουν. Είναι ιδιαίτερα σημαντικό στην κατασκευή του να δοθεί η απαραίτητη προσοχή, διότι η χρήση του θα πρέπει να διακρίνει κάθε ευπάθεια, καινούργια ή ήδη υπάρχουσα, με βάση κάποια δεδομένα κριτήρια. Με αυτόν τον τρόπο το πλαίσιο προσδιορίζει τις ευπάθειες με μια δομημένη μέθοδο και δεν χρησιμοποιεί τις ήδη υπάρχουσες κατηγοριοποιήσεις.

2.1 Εισαγωγή στο μεθοδολογικό πλαίσιο

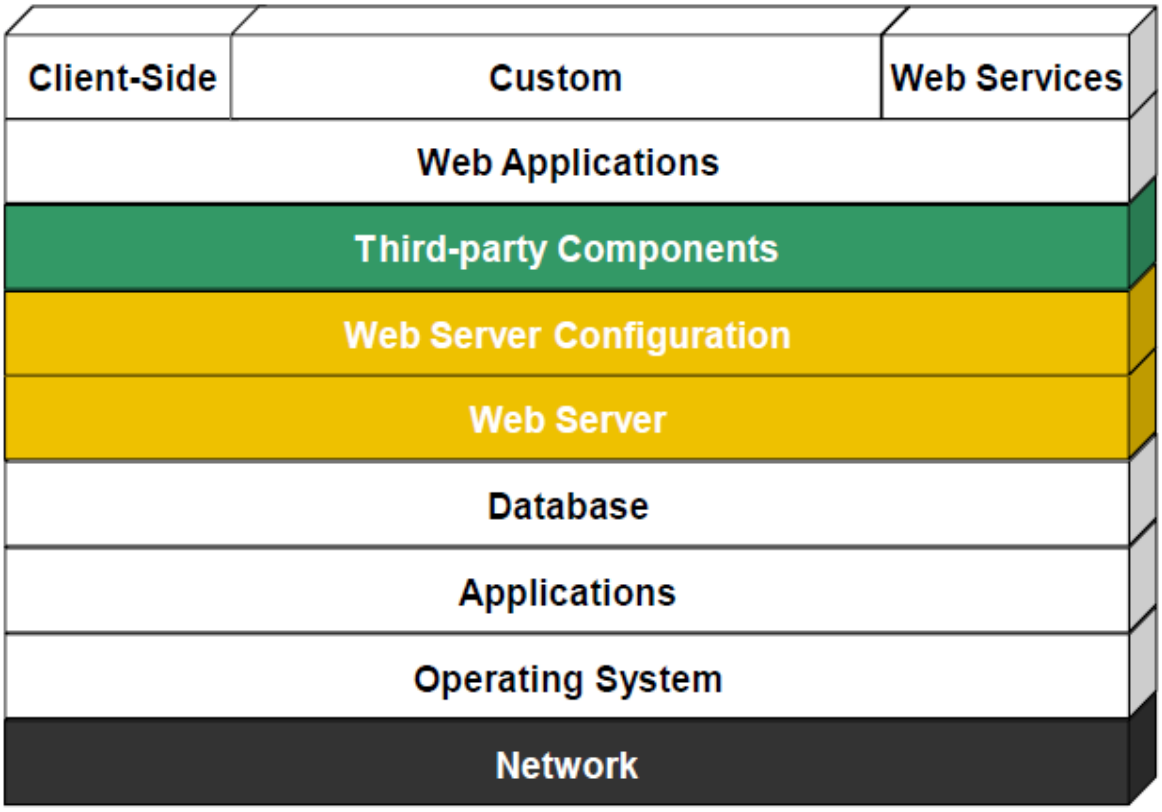
Αρχικά θα χωρίσουμε το πλαίσιο σε επίπεδα, καθένα από τα οποία θα περιλαμβάνει ένα ή περισσότερα τμήματα του Web πληροφοριακού συστήματος. Με αυτόν τον τρόπο κάθε ευπάθεια θα εντάσσεται στο επίπεδο που αντιστοιχούν τα τμήματα του συστήματος που επηρεάζει. Έπειτα θα αναφέρουμε όποιες ευπάθειες του πρώτου κεφαλαίου υπάγονται σε αυτά τα επίπεδα-κατηγορίες. Επίσης θα δοθούν παραδείγματα της επιρροής τους στο σύστημα.

Τα επίπεδα του πλαισίου είναι (α) το επίπεδο δικτύου, (β) το επίπεδο host, (γ) το επίπεδο βάσεων δεδομένων, (δ) το επίπεδο διαδικτυακών εφαρμογών και (ε) το επίπεδο κώδικα.

2.2.1 Το επίπεδο δικτύου

Σε αυτό το επίπεδο κατηγοριοποιούνται οι ευπάθειες που αφορούν το κομμάτι της επικοινωνίας του χρήστη με το πληροφοριακό μας σύστημα μέσω του διαδικτύου. Επίσης ιδιαίτερη σημασία πρέπει να δοθεί στην επικοινωνία των

κομματιών λογισμικού άλλου κατασκευαστή με το σύστημα. Πολλές φορές υπάρχει συνεργασία με άλλο λογισμικό προκειμένου να έχουμε το επιθυμητό αποτέλεσμα. Το επίπεδο του δικτύου περιλαμβάνει την διαδικασία μεταφοράς δεδομένων μεταξύ του χρήστη και του εξυπηρετητή ιστού του Web πληροφοριακού συστήματος, όπως επίσης και όλων των λογισμικών που καθιστούν εφικτή μια τέτοια επικοινωνία. Στο σχήμα 2.1 μπορείτε να διακρίνεται τα μέρη που επηρεάζουν οι ευπάθειες του επιπέδου δικτύου.



Σχήμα 2.1 Τα τμήματα επιπέδου δικτύου

Στο προηγούμενο κεφάλαιο σύμφωνα με την κατηγοριοποίηση των ευπαθειών που είχε γίνει από το OWASP διακρίναμε σε δύο κατηγορίες τις ευπάθειες. Η πρώτη που αφορούσε την επικοινωνία συστήματος-χρήστη ταυτίζεται με τις ευπάθειες του επιπέδου δικτύου στο παρόν μεθοδολογικό πλαίσιο. Ελλείψεις στην ασφάλεια των δεδομένων με μεθόδους κρυπτογραφίας και στην επικοινωνία του χρήστη με το σύστημα μέσω ενός ασφαλούς καναλιού, είναι τα πρώτα σημάδια ευπάθειας στο επίπεδο δικτύου του Web πληροφοριακού συστήματος. Εκτός από αυτές τις διακριτές ευπάθειες πρέπει να αναφερθεί πως τα

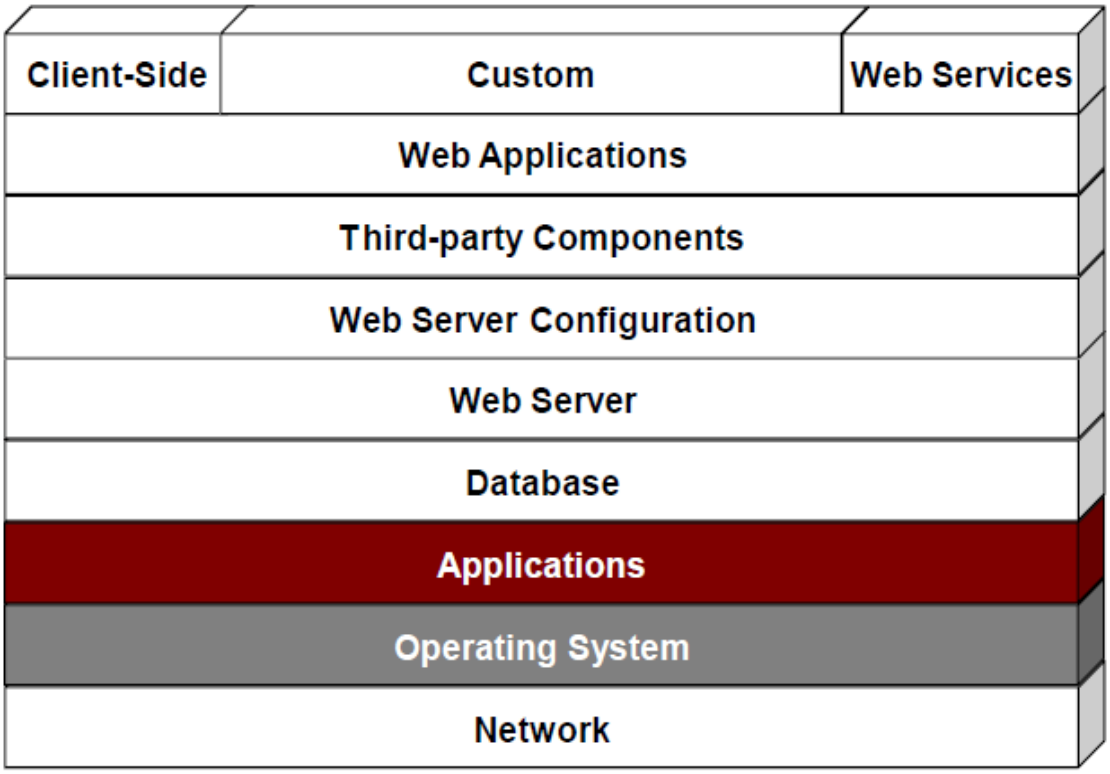
Web πληροφοριακά συστήματα είναι εξίσου, αν όχι περισσότερο, ευάλωτα στις επιθέσεις κακόβουλων χρηστών όπως κάθε υπολογιστικό σύστημα. Επειδή προσελκύουν τους κακόβουλους χρήστες λόγω του μεγάλου όγκου δεδομένων που διατηρούν και τις σημαντικότητας αυτών, χρήζουν ιδιαίτερης προστασίας και από αυτές τις απειλές. Πολλές φορές έχοντας δημιουργήσει ένα φαινομενικά ασφαλές σύστημα, μη δίνοντας την δέουσα προσοχή στην δικτυακή επικοινωνία καταλήγουμε στην ψευδαίσθηση πως τα δεδομένα που απορρέουν από αυτό είναι προστατευμένα.

Υπάρχει ένας πολύ μεγάλος αριθμός από ευπάθειες στο κομμάτι των δικτύων ενός Web πληροφοριακού συστήματος τις οποίες εκμεταλλεύονται οι κακόβουλοι χρήστες εξαπολύοντας επιθέσεις όπως Denial of Service(DoS), IP Spoofing, Eavesdropping και Packet Sniffing. Η πρώτη επίθεση στοχεύει στην άρνηση της παροχής των προβλεπόμενων υπηρεσιών του Web πληροφοριακού συστήματος στους χρήστες του, μέσω δικτύου. Στις δύο τελευταίες επιθέσεις ο επιτιθέμενος αποκτά πρόσβαση σε δεδομένα που εκπέμπονται και αφορούν πληροφορίες πιστοποίησης και περιεχόμενα βάσεων δεδομένων. Οι ευπάθειες του επιπέδου δικτύου κληρονομούνται και σε ανώτερα επίπεδα αρχιτεκτονικής του Web πληροφοριακού συστήματος όπως είναι το επίπεδο παρουσίασης που περιγράψαμε στο πρώτο κεφάλαιο. Για παράδειγμα με την τεχνική IP Spoofing ο επιτιθέμενος μπορεί να πιστοποιηθεί ως νόμιμος χρήστης σε ένα σύστημα το οποίο πιστοποιεί τους χρήστες τους με βάση το IP, και με αυτό τον τρόπο να αποκτήσει πλήρη πρόσβαση στο λογαριασμό του χρήστη. (William Stalling, 2005)

2.2.2 Το επίπεδο host

Όταν μια ευπάθεια σχετίζεται με το λειτουργικό σύστημα που είναι εγκατεστημένο στους εξυπηρετητές ή στις εφαρμογές που βοηθούν τις εσωτερικές λειτουργίες του συστήματος, τότε την εντάσσουμε στο επίπεδο host του πλαισίου προσδιορισμού. Κάθε λειτουργικό σύστημα έχει τις δικές του ιδιαιτερότητες και είναι αρκετά τα κενά στην ασφάλεια που περιέχονται στο καθένα. Σχεδιάζοντας ένα Web πληροφοριακό σύστημα θεωρείται πρέπων να γνωρίζουμε αυτές τις ιδιαιτερότητες. Επίσης κάθε εφαρμογή που χρησιμοποιείται για εσωτερικές

λειτουργίες πρέπει να ελέγχεται, σχετικά με το αν αποτελεί δυνητική ευπάθεια του συστήματος. Άξιο επισήμανσης αποτελεί το γεγονός πως στις δέκα πιο διαδεδομένες και συχνές ευπάθειες σύμφωνα με το OWASP δεν εμφανίζεται ούτε μια που να σχετίζεται καθολικά και άμεσα με το λειτουργικό σύστημα του Web πληροφοριακού συστήματος. Στο σχήμα 2.2 απεικονίζονται τα τμήματα του συστήματος που επηρεάζουν οι ευπάθειες που συνθέτουν το επίπεδο host.



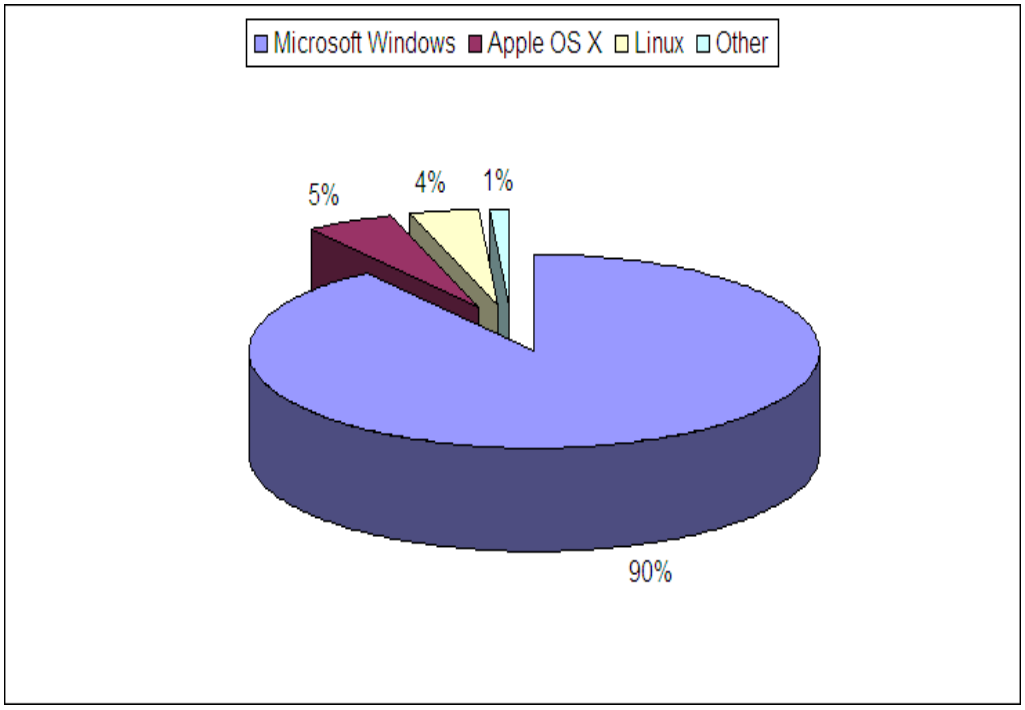
Σχήμα 2.2 Τα τμήματα του επιπέδου host

Για την προστασία των λειτουργικών τους συστημάτων (είτε εμπορικών είτε ελεύθερων) από απειλές, οι παροχείς είναι αναγκασμένοι να τα αναβαθμίζουν συνεχώς. Το 2005 το US CERT[2] στην ετήσια του καταγραφή των ευπαθειών στα λειτουργικά συστήματα καταμέτρησε συνολικά 5.198 ευπάθειες, εκ των οποίων 812 στα Microsoft Windows, 2.328 στα Linux/Unix και 2.058 ευπάθειες σε συστήματα με πολλαπλά λειτουργικά συστήματα. Στο ερώτημα του ποιό λειτουργικό σύστημα είναι το πιο ασφαλές, σε μια έρευνα που διεξήχθη το 2006-

[2] United States Computer Emergency Readiness Team http://www.us-cert.gov/

2007 από την Symantec η απάντηση που δόθηκε ήταν τα Microsoft Windows. Η αντικειμενική απάντηση είναι πως κάτι τέτοιο εξαρτάται από τις παραμετροποιήσεις του host. Οι περισσότερες ευπάθειες δεν αφορούν τα λειτουργικά συστήματα άμεσα, αλλά μπορούμε να διακρίνουμε τις ευπάθειες των συστατικών του πυρήνα τους, αντιμετωπίζοντας τα ως οντότητες.

Το κριτήριο πάνω στο οποίο εξετάζονται τα λειτουργικά συστήματα είναι οι ευπάθειες που μπορούν οι κακόβουλοι χρήστες να εκμεταλλευτούν μέσω της απομακρυσμένης πρόσβασης. Κρίσιμος παράγοντας είναι η διαθεσιμότητα του κάθε λειτουργικού συστήματος, καθώς οι κακόβουλοι χρήστες παρακινούνται για επιθέσεις σε λειτουργικά με μεγάλη συχνότητα εμφάνισης στα Web πληροφοριακά συστήματα. Στο παρακάτω σχήμα 2.2.1 φαίνεται τα ποσοστά των λειτουργικών συστημάτων σε υπολογιστές που λειτουργούν ως network host σε workstations είτε ως servers.



Σχήμα 2.2.1 Μερίδιο λειτουργικών συστημάτων σε υπολογιστές Host

Παρατηρώντας το παραπάνω σχήμα συμπεραίνουμε πως οι κακόβουλοι χρήστες είναι λογικό να κατευθύνονται περισσότερο στο να εκμεταλλεύονται ευπάθειες που αφορούν τα Microsoft Windows καθώς αποτελούν μεγαλύτερο

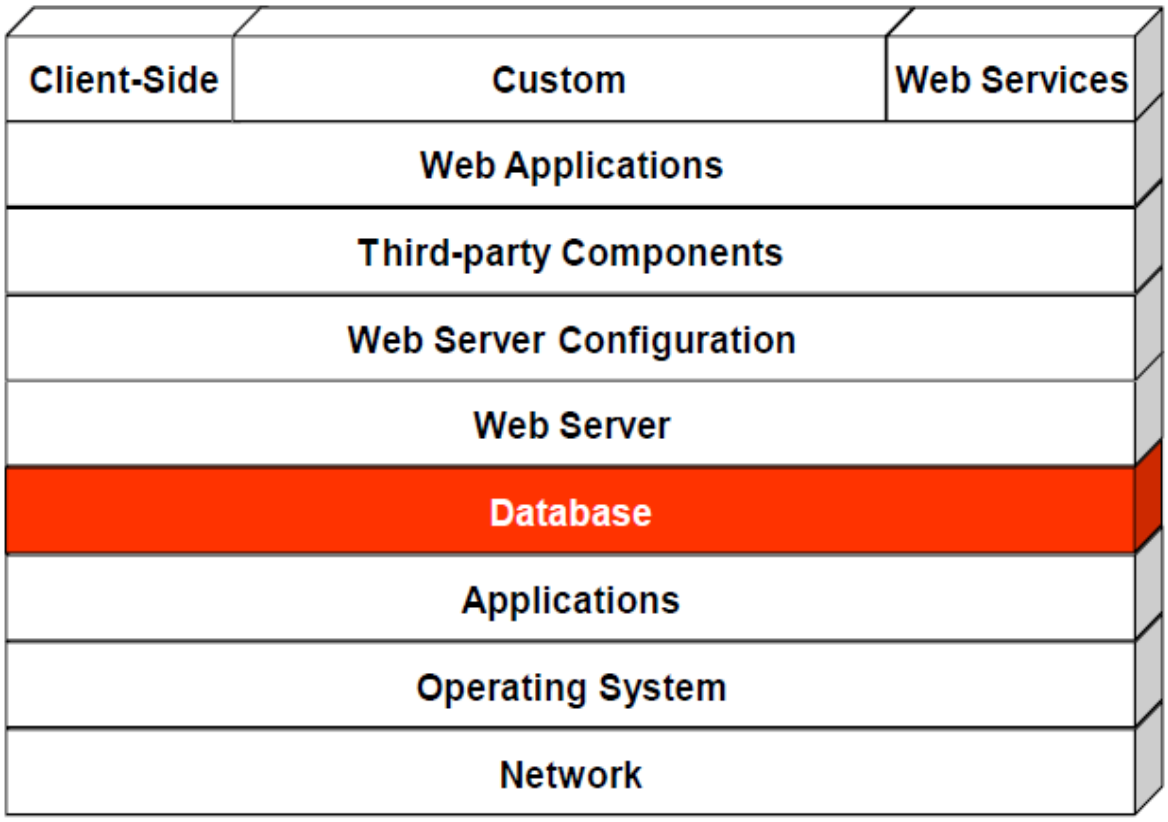
μερίδιο στο σύνολο των υπολογιστών που λειτουργούν ως Host. Ειδικότερα επειδή οι ευπάθειες στα λειτουργικά συστήματα αναμιγνύονται με τις ευπάθειες των εφαρμογών που τρέχουν στα συστήματα αυτά καταλαβαίνουμε πως είναι λογικά τα ποσοστά του σχήματα 2.2.1. Οι γνώμες όμως διχάζονται σχετικά με το εάν πρέπει να υπολογίζονται οι ευπάθειες των εφαρμογών του κάθε λειτουργικού συστήματος ως ευπάθειες του ίδιου του λειτουργικού συστήματος. Αρκετοί ειδικοί υποστηρίζουν πως μια τέτοια ευπάθεια εφαρμογής σαφώς και αφορά το λειτουργικό σύστημα το οποίο χρησιμοποιεί για να λειτουργήσει, ενώ άλλοι υποστηρίζουν πως με αυτήν την μέθοδο τα λειτουργικά συστήματα Linux/Unix λόγω του μεγάλου πλήθους των ελεύθερου λογισμικού εφαρμογών τους δείχνουν εσφαλμένος πιο ευάλωτα σε επιθέσεις κακόβουλων χρηστών.

2.2.3 Το επίπεδο βάσεων δεδομένων

Στο επίπεδο βάσεων δεδομένων περιλαμβάνονται οι ευπάθειες που αφορούν στην λειτουργία των βάσεων δεδομένων του Web πληροφοριακού συστήματος. Επειδή στις βάσεις δεδομένων αποθηκεύονται σημαντικές πληροφορίες και δεδομένα τόσο του πελάτη όσο και του συστήματος, οι ευπάθειες του επιπέδου βάσεων δεδομένων θεωρούνται μεγάλης βαρύτητας και η αγνόηση τους αποτελεί σοβαρό κίνδυνο

Στο επίπεδο βάσεων δεδομένων υπάγεται οποιοδήποτε μέρος ενός Web πληροφοριακού συστήματος χρησιμοποιείται για την αποθήκευση δεδομένων. Αυτό μπορεί να είναι από ένα απλό αρχείο κειμένου στο οποίο αποθηκεύονται και από το οποίο ανακτώνται δεδομένα, ένας κατάλογος στα Web πληροφοριακά συστήματα που χρησιμοποιούν το πρωτόκολλο LDAP[3] (Lightweight Directory Access Protocol), μέχρι συστήματα διαχείρισης βάσεων δεδομένων με χρήση ειδικών γλωσσών XML και SQL. Κυρίως λόγω της επικράτειας του σχεσιακού μοντέλου βάσεων δεδομένων η χρήση SQL είναι συχνότερη στα Web πληροφοριακά συστήματα ή η μετατροπή XML αρχείων σε SQL ερωτήματα. . Στο παρακάτω σχήμα παρατηρούμε τα τμήματα του συστήματος που επηρεάζουν οι ευπάθειες που αποτελούν το επίπεδο αυτό.

[3] http://en.wikipedia.org/wiki/LDAP



Σχήμα 2.3 Τα τμήματα του επιπέδου βάσεων δεδομένων

Η σημαντικότητα που έχουν οι βάσεις δεδομένων σε ένα Web πληροφοριακό σύστημα καθιστά απαραίτητη τη δημιουργία ενός τμήματος του μεθοδολογικού πλαισίου προσδιορισμού ευπαθειών, αποκλειστικά και μόνο για τις δικές του ευπάθειες. Η κυριότερη από αυτές που περιληπτικά περιγράφτηκε στο προηγούμενο κεφάλαιο είναι η έκχυση SQL (SQL injection). Αποτελεί την πιο άμεση ευπάθεια την οποία εκμεταλλεύονται οι κακόβουλοι χρήστες. Βάσεις δεδομένων που πάσχουν από αυτή την ευπάθεια είναι πολύ εύκολο να τις προσπελάσουν οι μη εξουσιοδοτημένοι χρήστες, καθώς ο τρόπος αυτής της προσπέλασης απαιτεί μονάχα ένα φυλλομετρητή ιστού και την απαραίτητη γνώση για εκμετάλλευση της ευπάθειας. Οπουδήποτε στο σύστημα υπάρχει πεδίο εισαγωγής δεδομένων σε βάση δεδομένων, όπως φόρμα ταυτοποίησης χρηστών ή πεδίο αναζήτησης, αυτό αποτελεί και πιθανή ευπάθεια εάν δεν έχουν ληφθεί τα απαραίτητα μέτρα. Τα μέτρα αυτά είναι το φιλτράρισμα των δεδομένων που εισέρχονται στις βάσεις δεδομένων από τους χρήστες και ο τρόπος ισχυρής γραφής (strongly typed) που ουσιαστικά αποτελεί γραφή με περιορισμούς

ασφαλείας. Δύο παραδείγματα για να γίνει πιο κατανοητός ο τρόπος ισχυρής γραφής είναι τα εξής: Σε ένα πεδίο μεταβλητών τύπου integer δεν επιτρέπεται να εισαχθεί μεταβλητή τύπου string.Επίσης μια διαδικασία που πραγματοποιείται σε ένα συνδυασμό από δεδομένα θα πρέπει να απαγορεύεται να πραγματοποιηθεί μόνο για αριθμούς.

Για να γίνει κατανοητή η ευπάθεια της έκχυσης SQL στις βάσεις δεδομένων ενός Web πληροφοριακού συστήματος θα δοθεί ένα παράδειγμα επίθεσης SQL το οποίο αντιμετωπίζεται με τους τρόπους που αναφέρθηκαν παραπάνω.

Ας υποθέσουμε πως έχουμε ένα σύστημα στο οποίο ο χρήστης μετά την πιστοποίηση έχει την δυνατότητα να αναζητήσει περιεχόμενο. Το περιεχόμενο αυτό μπορεί να είναι προϊόντα, πληροφορίες ή οποιασδήποτε μορφής δεδομένα που μπορούν να εισαχθούν και να αναζητηθούν σε μια βάση δεδομένων. Από πλευράς κώδικα η σύνδεση με μια βάση δεδομένων έχει την εξής μορφή:

```
V_cat = request(“dedomena”)
Sqlstr=”SELECT * FROM vasi WHERE Category=” & V_cat & “”
Set rs=conn.execute(sqlstr)
```

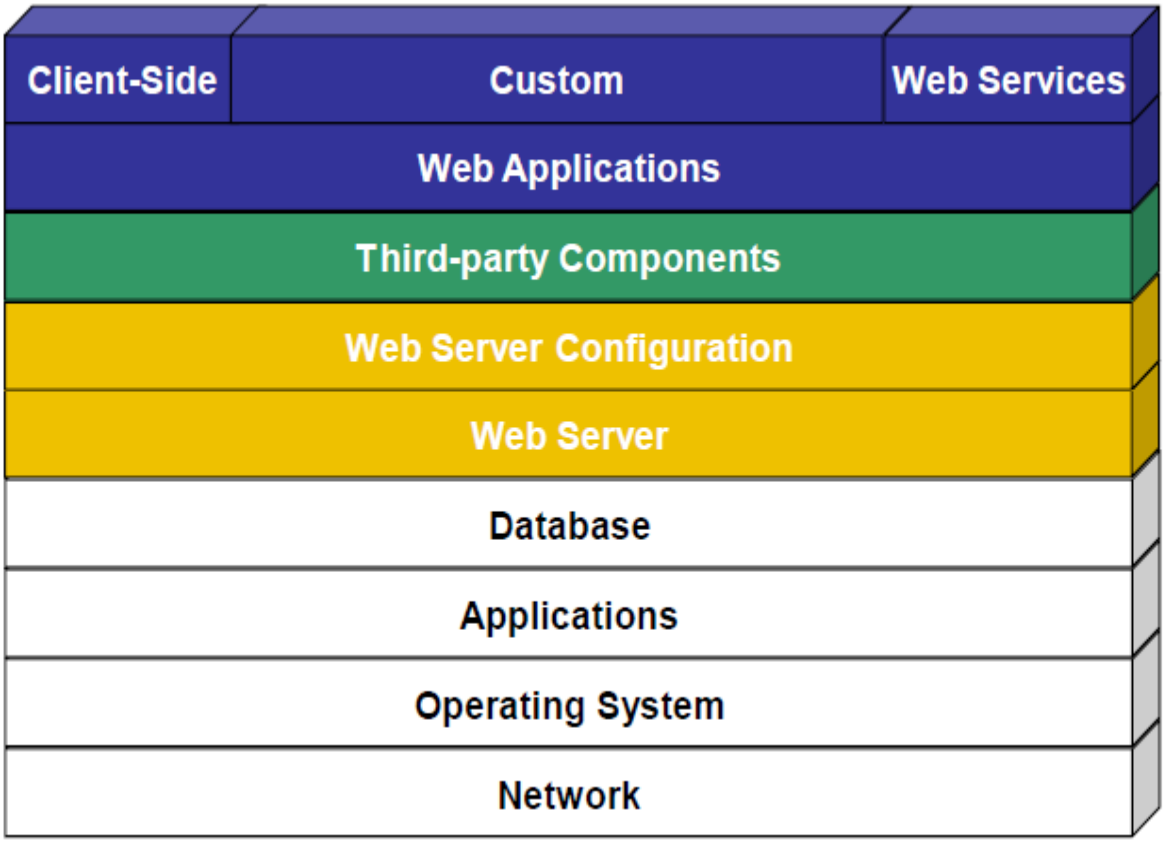
Με τις παραπάνω εντολές υπάρχει μια βάση δεδομένων όπου ο χρήστης δίνει μια τιμή και αναζητά όλες τις πληροφορίες για την κατηγορία της τιμής που έχει πληκτρολογήσει. Η εισαγωγή της τιμής από τον χρήστη μπορεί να γίνει είτε σε κάποιο πεδίο μιας φόρμας αναζήτησης είτε απευθείας στο URL της σελίδας (π.χ. www.test.gr/index.asp?category=hardware). Σε περίπτωση που το σύστημα μας αντιμετωπίζει ευπάθεια έκχυσης SQL ο κακόβουλος χρήστης μπορεί να πληκτρολογήσει την τιμή **’ or 1=1--** το οποίο όταν μεταφραστεί σε ερώτημα SQL θα έχει την εξής μορφή SELECT * FROM vasi WHERE Category = ‘hardware’ or 1=1--’ . Επειδή η διπλή παύλα είναι η εντολή στον MS SQL server να αγνοήσει το υπόλοιπο ερώτημα, με την εισαγωγή αυτή το ερώτημα θα μας εμφανίσει όλες τις τιμές του πίνακα vasi είτε είναι ίσες με το hardware είτε όχι. Όπως είναι φυσικό κάτι τέτοιο θα είχε καταστροφικά αποτελέσματα σε ό,τι έχει να κάνει με τις πληροφορίες του συστήματος. Αν και το παράδειγμα αυτό είναι αρκετά απλό με τον

συγκεκριμένο τρόπο επίθεσης σε αυτήν την ευπάθεια ένας κακόβουλος χρήστης μπορεί να λάβει τον πλήρη έλεγχο ενός συστήματος.

Ευτυχώς εκτός των άλλων τρόπων αντιμετώπισης τελευταία χρησιμοποιούνται αποθηκευμένες διαδικασίες (Stored Procedures) οι οποίες εκτελούν ελέγχους πληκτρολόγησης στα εισαγόμενα δεδομένα και βοηθάνε ριζικά στην αντιμετώπιση των ευπαθειών των βάσεων δεδομένων.

2.2.4 Το επίπεδο διαδικτυακών εφαρμογών

Ίσως το πιο σημαντικό επίπεδο του μεθοδολογικού πλαισίου προσδιορισμού ευπαθειών μιας και οι διαδικτυακές εφαρμογές αποτελούν πηγή αρκετών από τις μεγαλύτερες ευπάθειες που συναντάμε στα Web πληροφοριακά συστήματα. Σε αυτό το επίπεδο περιλαμβάνονται τα τμήματα του συστήματος που χρησιμοποιούν οι διαδικτυακές εφαρμογές για λειτουργήσουν. Μια διαδικτυακή εφαρμογή είναι ένα λογισμικό το οποίο είναι προσβάσιμο μέσω του παγκόσμιου ιστού. Έτσι τα τμήματα αυτού του επιπέδου εκτός από τις διαδικτυακές εφαρμογές είναι ο εξυπηρετητής ιστού με τις παραμετροποιήσεις του, τα τμήματα λογισμικού άλλων κατασκευαστών που χρειάζονται οι διαδικτυακές μας εφαρμογές για να εκτελέσουν ορισμένες από τις λειτουργίες τους, οι υπηρεσίες ιστού, οι απαραίτητες λειτουργίες της μεριάς του πελάτη και τα διάφορα στοιχεία για την διαδραστική επικοινωνία του πελάτη με το Web πληροφοριακό σύστημα. Παρακάτω βλέπουμε τα τμήματα του συστήματος τα οποία αποτελούν το επίπεδο διαδικτυακών εφαρμογών.



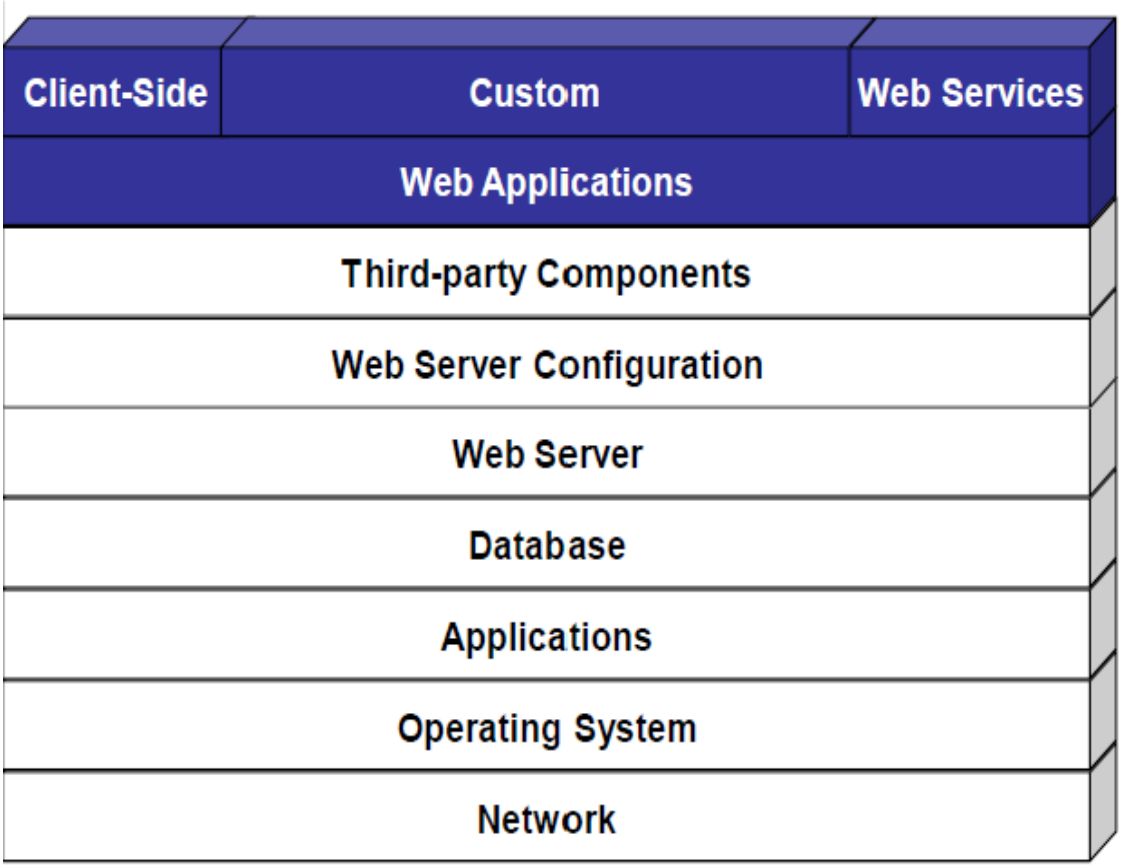
Σχήμα 2.4 Τα τμήματα του επιπέδου διαδικτυακών εφαρμογών

Στο επίπεδο διαδικτυακών εφαρμογών ανήκει μια πληθώρα ευπαθειών, όπως περιγράφθηκαν στο προηγούμενο κεφάλαιο, λόγω του μεγάλου εύρους που καλύπτουν τα τμήματα που περιλαμβάνει. Κυρίως επειδή οι διαδικτυακές εφαρμογές είναι το κομμάτι που αλληλεπιδρά με τον χρήστη σε ένα Web πληροφοριακό σύστημα και επίσης διαχωρίζουν το Web πληροφοριακό σύστημα από τα απλά πληροφοριακά συστήματα. Αναφορικά οι ευπάθειες του προηγούμενου κεφαλαίου που περιλαμβάνονται σε αυτό το επίπεδο του πλαισίου προσδιορισμού είναι οι εξής: Cross Site Scripting(XSS), παράνομη πιστοποίηση και διαχείριση συνδέσεων, διαρροή πληροφοριών και μη σωστός χειρισμός λαθών και η πλαστογραφία αίτησης Cross Site. Τα παραπάνω ενώ μπορεί να επηρεάζουν και άλλα τμήματα του Web πληροφοριακού συστήματος, εντάσσονται στο παρών επίπεδο κυρίως λόγω του ότι το μεγαλύτερο τους τμήμα αφορά τις διαδικτυακές εφαρμογές και τον τρόπο λειτουργίας τους. Η βελτίωση της ασφάλειας των διαδικτυακών εφαρμογών αποτελεί την καλύτερη λύση. Με μεθόδους όπως κωδικοποίηση των εισόδων που δίνει ο χρήστης στην εφαρμογή, για αποφυγή ευπάθειας Cross Site Scripting, και δημιουργία πινάκων πρόσβασης,

για ορισμό των όρων πρόσβασης ώστε να προστατευτούμε από παράνομες πιστοποιήσεις, μπορούμε να θωρακίσουμε το Web πληροφοριακό σύστημα από τέτοιες ευπάθειες των διαδικτυακών του εφαρμογών.

2.2.5 Το επίπεδο κώδικα

Το τελευταίο επίπεδο του πλαισίου προσδιορισμού των ευπαθειών είναι το επίπεδο κώδικα. Σε αυτό υπάγονται οι ευπάθειες που παρουσιάζονται στον κώδικα των διαδικτυακών εφαρμογών του Web πληροφοριακού συστήματος. Οποιοδήποτε κενό ασφάλειας δημιουργείται στο σύστημα λόγω της εσφαλμένης κωδικοποίησης ενός μέρους μιας εφαρμογής μπορεί να οδηγήσει σε καταστροφικές συνέπειες. Στο σχήμα 2.5 παρουσιάζονται τα τμήματα του επιπέδου κώδικα.



Σχήμα 2.5 Τα τμήματα του επιπέδου κώδικα

Μια πιο εξειδικευμένη μορφή ευπαθειών των Web πληροφοριακών συστημάτων αλλά και των διαδικτυακών τους εφαρμογών είναι αυτές που ανήκουν στο επίπεδο κώδικα και αφορούν λάθη ή παραλείψεις των προγραμματιστών στην συγγραφή του κώδικα. Βασικές ευπάθειες που έχουν αναφερθεί από το OWASP και ανήκουν σε αυτό το επίπεδο του πλαισίου προσδιορισμού ευπαθειών είναι η αποτυχία περιορισμού της πρόσβασης μέσω URL, η μη ασφαλής άμεση αναφορά σε αντικείμενο και η εκτέλεση επιβλαβούς αρχείου. Παρατηρούμε πως και οι τρείς ευπάθειες που χρησιμοποιούνται ως παράδειγμα για την κατανόηση του επιπέδου κώδικα έχουν να αντιμετωπίσουν ελλιπή έλεγχο πρόσβασης ή έλεγχο δεδομένων εισόδου. Για παράδειγμα ως μέτρο αντιμετώπισης της ευπάθειας της μη ασφαλούς άμεσης αναφοράς σε αντικείμενο αντί να χρησιμοποιήσουμε στον κώδικα μας το

<select name="language">

<option value="English">English</option>

είναι προτιμότερο να δημιουργήσουμε έναν πίνακα με κωδικοποιημένες τιμές για να πετύχουμε ασφαλέστερη αναφορά. Έτσι τροποποιούμε τον κώδικα μας και γράφουμε για το ίδιο κομμάτι

<select name="language">

<option value="78463a384a5aa4fad5fa73e2f506ecfc">English</option>

και με αυτόν τον τρόπο προστατεύουμε τις άμεσες αναφορές σε αντικείμενα από επιθέσεις τύπου ωμής βίας (brute forcing).Τέτοιες μικρές αλλά σημαντικές αλλαγές στον κώδικα των εφαρμογών του συστήματος είναι απαραίτητες για την αντιμετώπιση των ευπαθειών

2.3 Σύνοψη

Με την δημιουργία του μεθοδολογικού πλαισίου αξιολόγησης των ευπαθειών αμέσων ξεχωρίζουμε τις ευπάθειες σε συγκεκριμένες κατηγορίες. Οργανώνοντας τις ευπάθειες σύμφωνα με το πεδίο επιρροής τους πάνω στα τμήματα του Web πληροφοριακού συστήματος διευκολύνουμε το έργο του μηχανικού πληροφορικής. Αντί να προσπαθεί να αντιμετωπίσει μαζικά τις

ευπάθειες, κατανοώντας την επιρροή τους, θα τις αντιμετωπίσει τμηματικά. Επίσης αναλόγως με το τι θα κρίνει ως κυριότερο πρόβλημα ασφάλειας στο Web πληροφοριακό σύστημα, κατευθυνόμενος από το πλαίσιο αξιολόγησης, θα αντιμετωπίσει τις ευπάθειες ενός ή περισσοτέρων επιπέδων. Στην περίπτωση για παράδειγμα που το πρόβλημα αφορά την διαδικτυακή επικοινωνία μεταξύ συστήματος-χρήστη βάση του πλαισίου αντιμετωπίζουμε μια ευπάθεια που υπάγεται στο επίπεδο δικτύου και με αυτό τον τρόπο δεν χρειάζεται ο μηχανικός πληροφορικής να ασχοληθεί με τον κώδικα των εφαρμογών του συστήματος ή με τις βάσεις δεδομένων. Έτσι κερδίζει πολύτιμο χρόνο καθώς ευκολότερα και γρηγορότερα καθοδηγείται στη σωστή παραμετροποίηση των τμημάτων του Web πληροφοριακού συστήματος που σχετίζονται με την ευπάθεια που καλείται να αντιμετωπίσει.

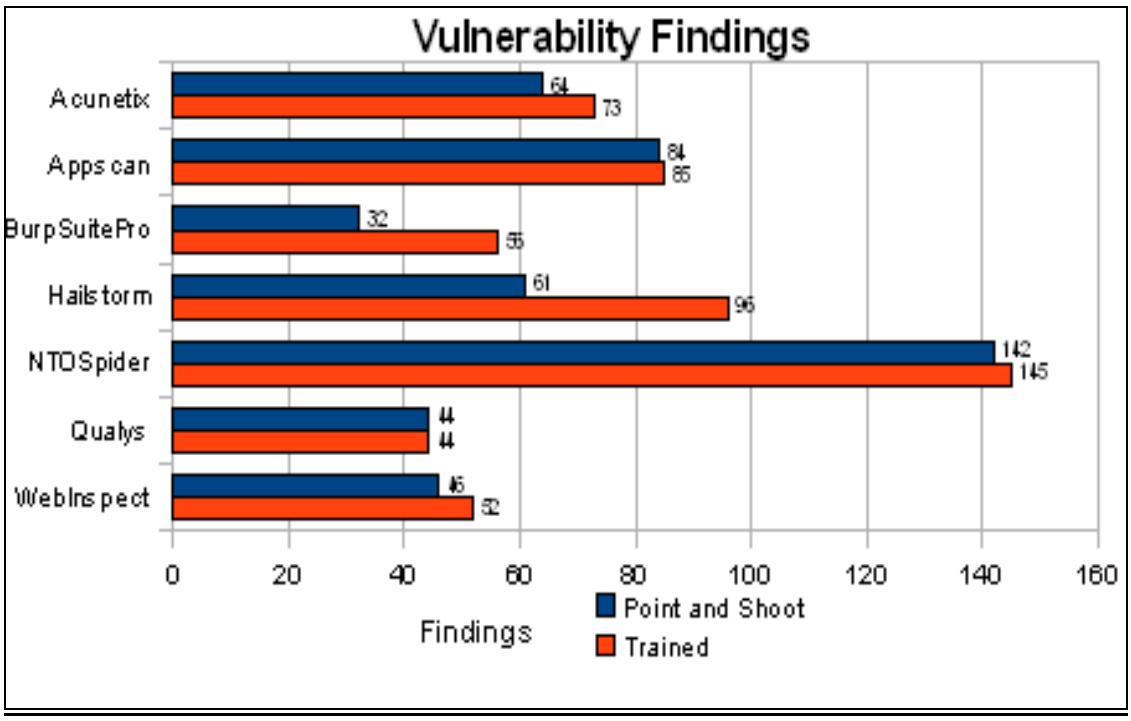
ΚΕΦΑΛΑΙΟ 3 : Το πρόγραμμα Rational Appscan

Από τα πέντε επίπεδα του μεθοδολογικού πλαισίου του προηγούμενου κεφαλαίου οι περισσότερες ευπάθειες εμφανίζονται στο επίπεδο διαδικτυακών εφαρμογών. Η συχνή τους εμφάνιση σε αυτό το επίπεδο συμβαίνει εξαιτίας της άμεσης σύνδεσης του με το επίπεδο παρουσίασης, της αρχιτεκτονικής ενός Web πληροφοριακού συστήματος που αναφέραμε στο πρώτο κεφάλαιο.

Για τις ανάγκες της πτυχιακής θα γίνει χρήση του προγράμματος Rational Appscan με το οποίο θα αναπτυχθούν σενάρια χρήσης του πάνω σε διαδικτυακές εφαρμογές. Προτού φτάσουμε στο σημείο της υλοποίησης αυτών των σεναρίων χρήσης θα πρέπει αρχικά να αναφέρουμε τις δυνατότητες που έχει το πρόγραμμα και στην συνέχεια πιο λεπτομερειακά να γίνει περιγραφή του χειρισμού του από τους χρήστες.

Η επιλογή του συγκεκριμένου προγράμματος έγινε από μια μεγάλη κατηγορία προγραμμάτων διαγνωστικών ελέγχων πάνω σε διαδικτυακές εφαρμογές. Σύμφωνα με μια έρευνα του Larry Suto το 2010, το Rational Appscan βρέθηκε στην δεύτερη θέση των διαγνωστικών προγραμμάτων ανίχνευσης ευπαθειών. Στην έρευνα αυτή τα προγράμματα εξετάστηκαν σε διάφορους τομείς όπως το πλήθος των ευπαθειών που ανιχνεύουν, το σύνολο του χρόνου που χρειάζονται για να πραγματοποιήσουν μια εξέταση, την ευκολία χρήσης τους και άλλα χαρακτηριστικά τους. Στο σχήμα 3.1 βλέπουμε την κατάταξη των προγραμμάτων Acunetix, Appscan, BurpSuitePro, Hailstorm, NTOSpider, Qualys και WebInspect.

Ο λόγος που επιλέχθηκε το Rational Appscan είναι η ευκολία χρήσης του σε σχέση με τα άλλα διαγνωστικά εργαλεία που υπάρχουν σε συνδυασμό με την υψηλή επίδοση του.



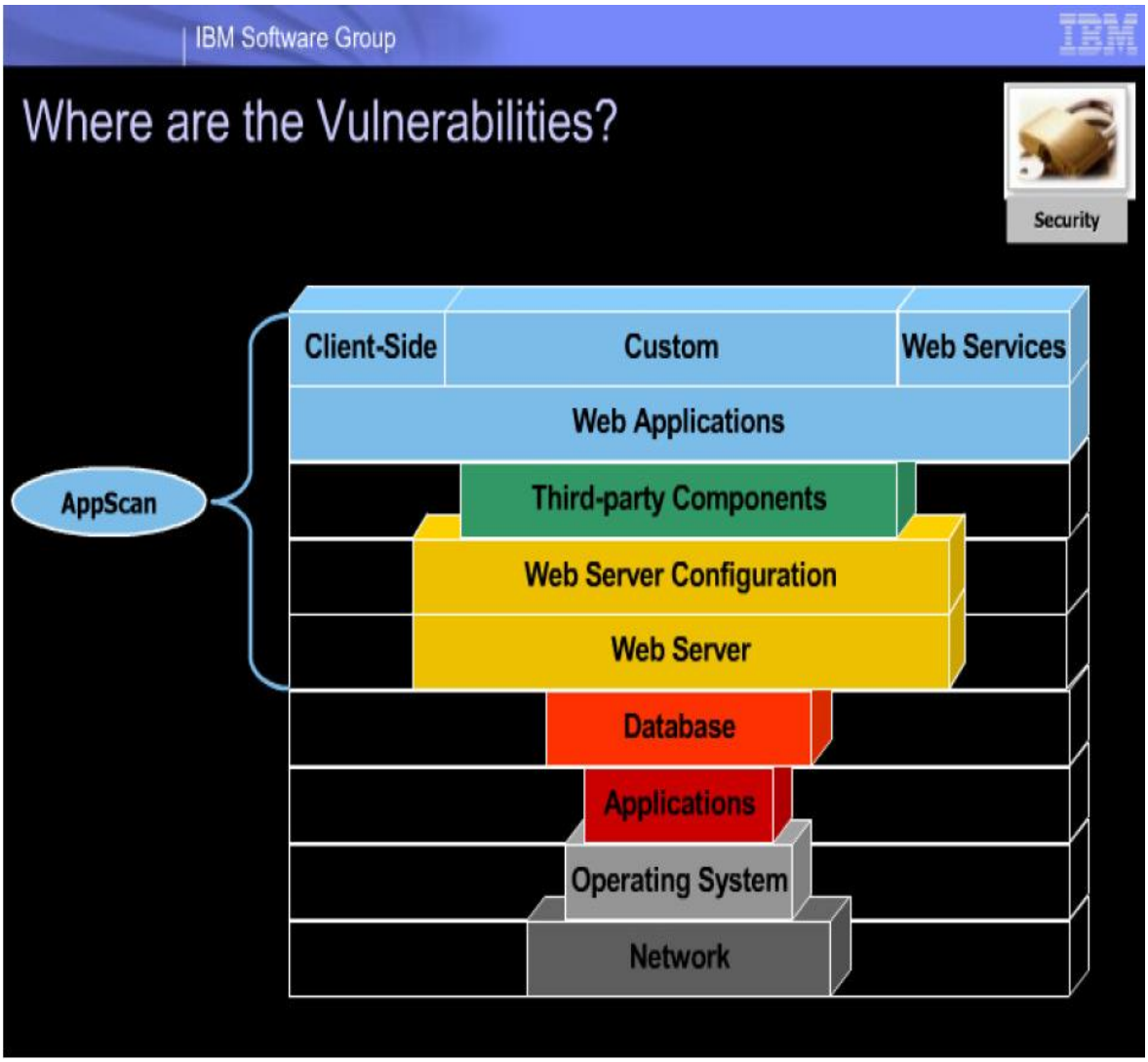
Σχήμα 3.1 Σύγκριση διαγνωστικών εργαλείων διαδικτυακών εφαρμογών

3.1 Περιγραφή του προγράμματος Rational Appscan

Το πρόγραμμα Rational Appscan της IBM είναι ένα από τα κορυφαία προγράμματα στην αγορά στον τομέα της εξέτασης του εσωτερικού των διαδικτυακών εφαρμογών. Μέσα από την εξέταση αυτή εντοπίζονται και δοκιμάζονται οι ευπάθειες των διαδικτυακών εφαρμογών. Μερικές από τις πιο συχνές που αποκαλύπτονται στις εφαρμογές μετά την εξέταση είναι η έκχυση SQL (SQL injection), το Cross Site Scripting (XSS) και η υπερχείλιση του Buffer (Buffer Overflow). Το Rational Appscan θεωρείται από τα πιο εύχρηστα προγράμματα του είδους του. Χρησιμοποιείται για να επιμορφώσει το προσωπικό των εταιριών δημιουργίας διαδικτυακών εφαρμογών, σχετικά με το πώς να αποτρέψουν επιθέσεις στις ευπάθειες που ανιχνεύει, χρησιμοποιώντας σωστή τροποποίηση των παραμέτρων των διαδικτυακών εφαρμογών και εξάσκηση στη συγγραφή ασφαλούς κώδικα. Σαν εργαλείο είναι επεκτάσιμο και συνεχώς αναβαθμίσιμο στις απαιτήσεις που έχει ένας τομέας σαν αυτόν της ασφάλειας των διαδικτυακών εφαρμογών. Καινούργιες ευπάθειες ή διαφορετικοί τρόποι επίθεσης στις ήδη υπάρχουσες, ανάγουν την επεκτασιμότητα των προγραμμάτων αυτών πολύ ψηλά

στις προτεραιότητες των χρηστών. Τέλος ως αποτέλεσμα της εξέτασης των διαδικτυακών εφαρμογών το Rational Appscan παράγει εκτενείς αναφορές οι οποίες είναι εξαιρετικά χρήσιμες στην κατανόηση της προόδου που έχει κάνει η ασφάλεια των διαδικτυακών εφαρμογών που εξετάζει. Οι παραμετροποιήσεις αυτών των αναφορών είναι αρκετές έτσι ώστε να ταιριάζουν σε κάθε περίσταση της εταιρίας.

Στο σχήμα 3.2 βλέπουμε τα τμήματα του Web πληροφοριακού συστήματος όπου εξετάζονται από το Rational Appscan σύμφωνα με την κατηγοριοποίηση που πραγματοποιήθηκε στο πρώτο κεφάλαιο της πτυχιακής εργασίας.



Σχήμα 3.2 Τμήματα εξέτασης του Rational Appscan

3.2 Χειρισμός του προγράμματος Rational Appscan

Αρχικά εκτελούμε το πρόγραμμα και επιλέγουμε να ξεκινήσουμε μια νέα εξέταση ενός Web πληροφοριακού συστήματος. Οι επιλογές που δίνονται είναι είτε εξέταση των διαδικτυακών εφαρμογών είτε των υπηρεσιών ιστού με την κατάλληλη επέκταση εγκατεστημένη στο πρόγραμμα. Έπειτα προχωράμε στην δεύτερη φόρμα επιλογών του προγράμματος που είναι η επιλογή του ενιαίου εντοπιστής πόρων (URL) του Web πληροφοριακού συστήματος που θέλουμε να εξετάσουμε. Επόμενη επιλογή είναι το συνθηματικό που δίνει πρόσβαση σε όλες τις περιοχές που θέλουμε να εξετάσει το σύστημα. Ο κωδικός και το όνομα χρήστη του διαχειριστή του συστήματος, στα περισσότερα συστήματα παρέχει πρόσβαση σε όποιες επιπλέον λειτουργίες εκτελεί η εφαρμογή και αν είναι γνωστός πρέπει να προτιμάται για την βέλτιστη εξέταση. Επόμενο βήμα είναι η επιλογή της σωστής πολιτικής (Policy) όσον αφορά την εξέταση. Οι πολιτικές που δίνονται από το πρόγραμμα προς επιλογή του χρήστη έχουν να κάνουν με ομάδες από διαγνωστικούς ελέγχους, που γίνονται προσανατολισμένοι σε κάποιο συγκεκριμένο τομέα του συστήματος. Αυτό το επιλέγει ο χρήστης όταν γνωρίζει το τμήμα στο οποίο θέλει να δώσει την απαραίτητη βαρύτητα. Για παράδειγμα η επιλογή μιας πολιτικής Infrastructure-Only προσανατολίζεται στον έλεγχο ευπαθειών που βασίζονται στα τμήματα λογισμικού άλλων κατασκευαστών και στην υποδομή του συστήματος. Από την άλλη η επιλογή Application-Only προσανατολίζεται στην λειτουργικότητα των εφαρμογών του συστήματος. Με την επιλογή Full Scan Configuration που δίνεται κατά την διάρκεια της διαδικασίας παραμετροποίησης της εξέτασης, ο χρήστης μπορεί να ορίσει όλους τους ελέγχους που υπάρχουν στην βιβλιοθήκη ελέγχων του Rational Appscan. Υπάρχει η δυνατότητα να εξεταστεί το σύστημα για μία μόνο ευπάθεια, για παράδειγμα έκχυση SQL, εάν είναι επιθυμητό από τον χρήστη ώστε να εξοικονομήσει χρόνο.

Μετά το πέρας όλων των παραμέτρων που χρειάζεται να δοθούν και των επιλογών που μπορεί να κάνει ο χρήστης, το πρόγραμμα πραγματοποιεί μια πλήρης εξέταση πάνω στο σύστημα. Έλεγχοι πραγματοποιούνται σε όλα τα μέρη των διαδικτυακών εφαρμογών εφόσον κάτι τέτοιο επιλέχθηκε στην πρώτη φόρμα επιλογής. Στην διάρκεια του ελέγχου χτίζεται από το πρόγραμμα ένας κατάλογος

με όλους τους συνδέσμους (links) ,τα cookies και τις παραμέτρους των εφαρμογών που συναντά έτσι ώστε στη συνέχεια να αποφασιστεί για ποιες ευπάθειες θα εξετασθεί το σύστημα. Κατά την διάρκεια αυτής της φάσης, ταυτόχρονα με τους ελέγχους, εκτελείται και ένα βοηθητικό πρόγραμμα ελέγχου το οποίο δίνει πληροφορίες για βελτιστοποίηση της εξέτασης σε περίπτωση που ο χρήστης δεν γνωρίζει άριστα την δομή του συστήματος ή δεν είναι εξοικειωμένος με το Rational Appscan. Μετά το τέλος της πλήρους εξέτασης το βοηθητικό πρόγραμμα εμφανίζει μια λίστα με επιλογές που θεωρεί πως ο χρήστης πρέπει να ενεργοποιήσει, να απενεργοποιήσει ή να τροποποιήσει. Συνήθως οι επιλογές που εμφανίζονται από το βοηθητικό πρόγραμμα έχουν μεγάλη σημασία για την βέλτιστη εξέταση του Web πληροφοριακού συστήματος.

Το πρώτο βήμα που κάνει το Rational Appscan στην εξέταση ενός Web πληροφοριακού συστήματος είναι η εξερεύνηση όλων των πτυχών του συστήματος με έμφαση στους τομείς που έχουμε ορίσει να εξεταστούν. Εφόσον ολοκληρωθεί αυτό το βήμα το πρόγραμμα περνάει στο δεύτερο που είναι ο έλεγχος του συστήματος. Σε αυτό το βήμα μιμείται επιθέσεις από κακόβουλους χρήστες με χρήση attack signatures (αρχεία που περιέχουν μια ακολουθία δεδομένων και χρησιμοποιούνται για να ανιχνεύσουν μια επίθεση). Οι επιθέσεις που μιμείται είναι παρόμοιες με αυτές που το σύστημα θα μπορούσε να αντιμετωπίζει καθημερινά. Κατά την διάρκεια του ελέγχου εμφανίζει τις ευπάθειες που ανακαλύπτει στο σύστημα και τις ξεχωρίζει σε τέσσερις κατηγορίες. Ευπάθειες υψηλού κινδύνου, ευπάθειες μεσαίου κινδύνου, ευπάθειες χαμηλού κινδύνου και παρατηρήσεις.

΄Υστερα από το τέλος του ελέγχου το πρόγραμμα προχωράει στο τρίτο βήμα. Σε αυτό εμφανίζει μια λίστα με όλες τις ευπάθειες που ανακάλυψε μετά το πέρας των διαδοχικών ελέγχων που έκρινε πως έπρεπε να πραγματοποιηθούν. Αφού τις κατηγοριοποιήσει στις τέσσερις κατηγορίες που αναφέρθηκαν παραπάνω, δίνει στον χρήστη τρεις καρτέλες προς επιλογή στην κάθε ευπάθεια. Η πρώτη είναι η αναφορά της ευπάθειας. Σε αυτήν δίνεται μια περιγραφή της ευπάθειας με σκοπό την επιμόρφωση του χρήστη. Περιλαμβάνει πληροφορίες για το τι είναι αυτή η ευπάθεια, με ποιον τρόπο επηρεάζει το σύστημα, με ποιόν τρόπο επωφελούνται οι κακόβουλοι χρήστες από αυτήν και άλλες σημαντικές πληροφορίες. Η δεύτερη καρτέλα ονομάζεται προτεινόμενη λύση. Σε αυτήν το

πρόγραμμα αναλύει με μεγάλη λεπτομέρεια τον τρόπο που χρειάζεται να ακολουθήσει ο χρήστης για να επιλύσει την ευπάθεια. Παρέχεται πληθώρα δομημένων τρόπων αντιμετώπισης των ευπαθειών, είτε γενικού τύπου είτε προσανατολισμένοι στην πλατφόρμα που υλοποιήθηκε το σύστημα (Asp.net, PHP, J2EE). Με αυτόν τον τρόπο το Rational Appscan συμβουλεύει τον χρήστη πώς να γράφει πιο ασφαλή κώδικα για την επίλυση της ευπάθειας. Η Τρίτη καρτέλα αιτήματος-απάντησης (request-respond), επιτρέπει στον χρήστη να παρατηρήσει τον έλεγχο που πραγματοποιήθηκε από το Rational Appscan. Με αυτόν τον τρόπο μπορεί ο χρήστης να βεβαιωθεί πως κατάλαβε για ποιόν λόγο θεωρείται ευπάθεια κάποιο μέρος του συστήματος από το πρόγραμμα.

Όλα όσα περιγράφτηκαν παραπάνω βρίσκονται στην επιλογή προβολής των θεμάτων ασφαλείας του συστήματος. Μια εναλλακτική προβολή που δίνεται από το Rational Appscan και αφορά τον σχεδιασμό του συστήματος είναι αυτή των ενεργειών αποκατάστασης. Αυτή η προβολή δεν προσπαθεί να επιλύσει σταδιακά τις ευπάθειες του συστήματος μια προς μια, αλλά εμφανίζει κάποιες προτάσεις που βοηθούν το σύστημα να προστατευτεί από ορισμένες ευπάθειες. Επίσης άξιο αναφοράς είναι πως με την προβολή αυτή και γνωρίζοντας ο χρήστης το μέγεθος των πόρων και του χρόνου που έχει στην διάθεση του για την ανάπτυξη του Web πληροφοριακού συστήματος, μπορεί να αποφασίσει καλύτερα που να δώσει βάρος κατά την ανάπτυξη του συστήματος ώστε να απαλείψει όσο γίνεται περισσότερες ευπάθειες.

Η τελευταία προβολή που ο χρήστης έχει την δυνατότητα να επιλέξει είναι αυτή των δεδομένων των εφαρμογών του συστήματος. Η προβολή αυτή εμφανίζει παραμέτρους των scripts, διαδραστικά στοιχεία του συστήματος (όπως URL), στοιχεία του συστήματος που επισκέφτηκε κατά την διάρκεια του ελέγχου, ανενεργοί σύνδεσμοι (broken links), φιλτραρισμένα στοιχεία, σχόλια, JavaScript και cookies από τον φυλλομετρητή στο επίπεδο παρουσίασης.

Τέλος μια από τις πιο χρήσιμες λειτουργίες του Rational Appscan αποτελεί η παραγωγή αναφορών μετά από κάθε εξέταση. Οι αναφορές που παράγονται από το πρόγραμμα μπορούν να χρησιμοποιηθούν σε διαφορετικούς τομείς μιας εταιρίας κατασκευής Web πληροφοριακών συστημάτων. Η λειτουργία παραγωγής αναφορών σχεδιάστηκε ώστε να δίνει την δυνατότητα στον χρήστη να επιλέξει πως θα είναι η τελική μορφή της αναφοράς. Εκτός των προτύπων αναφορών που

παρέχονται από το πρόγραμμα, δίνεται η δυνατότητα στον χρήστη να επιλέξει ο ίδιος πως θα παραμετροποιήσει την αναφορά.

Μια αναφορά περιλαμβάνει μια παράγραφο με τα αποτελέσματα της εξέτασης σε συνοπτική μορφή, αναλύει λεπτομερώς τα θέματα ασφάλειας που διαπίστωσε πως υπάρχουν στο Web πληροφοριακό σύστημα, προτείνει λύσεις ή τροποποιήσεις του συστήματος για τις ευπάθειες που διέγνωσε και τέλος αναφέρει όλες τις πληροφορίες που συνέλεξε για τα δεδομένα των εφαρμογών του συστήματος. Υπάρχει επίσης η δυνατότητα στην αναφορά να γίνεται σύγκριση των αποτελεσμάτων της εξέτασης με τις προδιαγραφές διαφόρων οργανισμών ή προγραμμάτων (OWASP, SANS, PCI), όπως αυτοί τις έχουν ορίσει για εταιρίες κατασκευής Web πληροφοριακών συστημάτων. Μια επιπλέον δυνατότητα είναι η προβολή αναφοράς στην οποία παρουσιάζεται το επίπεδο συμμόρφωσης του συστήματος σε αυστηρούς κανόνες που πρέπει να ακολουθούνται από εταιρίες σε διάφορα κράτη (FIPPA[4], COPPA[5], EFTA[6]). Επιπροσθέτως με την αναφορά Ανάλυσης Δέλτα (Delta Analysis) το Rational Appscan συγκρίνει αναφορές παλαιότερης εξέτασης με την τρέχουσα εξέταση για να διακρίνει ο χρήστης την πρόοδο που έχει πραγματοποιήσει το σύστημα στον τομέα των ευπαθειών μεταξύ των δύο αναφορών. Τέλος υπάρχουν τα πρότυπα αναφοράς που δημιουργούν αναφορές για ομάδες ατόμων μέσα σε μια εταιρία ώστε με αυτόν τον τρόπο να κερδίζεται χρόνος με μικρότερες και πιο ουσιαστικές αναφορές.

3.3 Σύνοψη

Σε κάθε περίπτωση Web πληροφοριακού συστήματος, είτε οι διαδικτυακές του εφαρμογές είναι χτισμένες σε HTML είτε σε AJAX, το Rational Appscan θα εξετάσει το σύστημα και με μεγάλη ακρίβεια θα αναφέρει τις ευπάθειες από τις οποίες κινδυνεύει και τις λύσεις στις οποίες ο χρήστης μπορεί να προβεί.

[4] Freedom of Information and Protection of Privacy Act (Καναδάς)

[5] Children Online Privacy Protection Act (Ηνωμένες Πολιτείες Αμερικής)

[6] Electronic Funds and Transfer Act (Ηνωμένες Πολιτείες Αμερικής)

ΚΕΦΑΛΑΙΟ 4: Ανάλυση-Αξιολόγηση πειράματος

Στα πλαίσια της παρούσας πτυχιακής εργασίας θα δημιουργηθεί σενάριο χρήσης του προγράμματος της IBM, Rational Appscan. Με την ακαδημαϊκή έκδοση του προγράμματος Rational Appscan υπάρχει η δυνατότητα εξέτασης μονάχα της διαδικτυακής εφαρμογής τραπεζικών συναλλαγών της ιστοσελίδας Altoro Mutual, η οποία έχει σχεδιαστεί αποκλειστικά για επιμορφωτικούς σκοπούς. Σκοπός του σεναρίου χρήσης είναι η καλύτερη κατανόηση των ευπαθειών που αναλύθηκαν στα προηγούμενα κεφάλαια, με παραδείγματα από ολοκληρωμένα Web πληροφοριακά συστήματα. Με αυτόν τον τρόπο θα κατανοηθούν πλήρως τα προβλήματα κοινών ευπαθειών που καλείται να αντιμετωπίσει κάποιος μηχανικός πληροφορικής υπεύθυνος για την ασφάλεια ενός Web πληροφοριακού συστήματος.

4.1 Σενάριο χρήσης του προγράμματος Rational Appscan για την ιστοσελίδα Altoro Mutual

Η ιστοσελίδα της Watchfire Inc. με όνομα Altoro Mutual είναι μια δυναμική ιστοσελίδα τραπεζικών συναλλαγών μέσω διαδικτύου (Web Banking) η οποία έχει δημιουργηθεί ως παράδειγμα επίδειξης της χρήσης διαγνωστικών προγραμμάτων. Η ιστοσελίδα περιλαμβάνει όλες τις λειτουργίες που θα περιείχε μια πραγματική ιστοσελίδα τραπεζικών λογαριασμών όπως πιστοποίηση χρηστών, μηχανή αναζήτησης περιεχομένων, διαχείριση λογαριασμών και άλλες λειτουργίες που συναντούνται σε πολλές ιστοσελίδες τέτοιου τύπου. Στα πλαίσια την πτυχιακής αυτής εργασίας θα γίνουν διαγνωστικοί έλεγχοι στην ιστοσελίδα Altoro Mutual με την χρήση του προγράμματος Rational Appscan και θα μελετηθούν τα αποτελέσματα τους. Ο λόγος που χρησιμοποιείται στο σενάριο χρήσης του προγράμματος Rational Appscan η συγκεκριμένη ιστοσελίδα, είναι διότι κατασκευάστηκε ώστε να περιέχει αντιπροσωπευτικές ευπάθειες που εμφανίζονται σε αυτού του είδους τις διαδικτυακές εφαρμογές. Με την χρήση του Rational

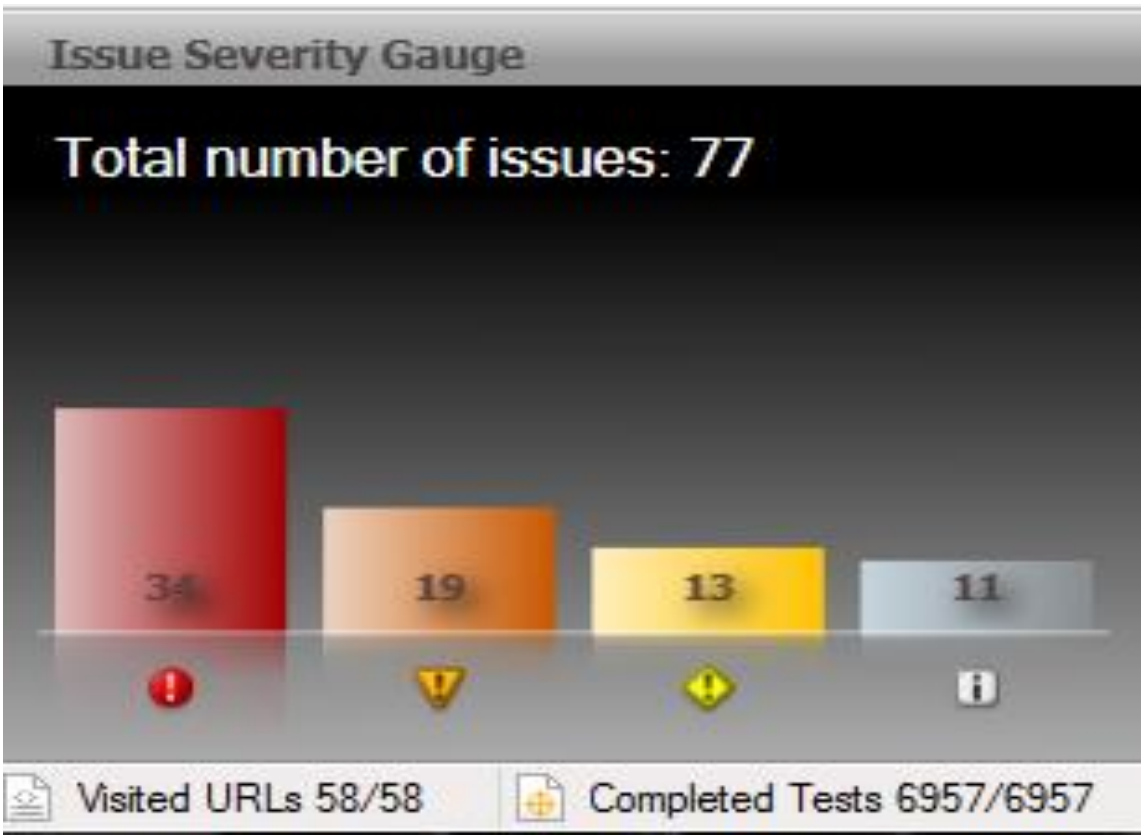
Appscan στην διαδικτυακή αυτή εφαρμογή θα περιγραφούν οι ευπάθειες που θα εντοπιστούν ώστε να γίνει κατανοητό το πεδίο επιρροής τους αλλά και ο τρόπος αντιμετώπισης αυτών.

Εικόνα 4.1 Η ιστοσελίδα Altoro Mutual

4.1.1 Εξέταση της ιστοσελίδας Altoro Mutual

Ξεκινώντας το πρόγραμμα Rational Appscan επιλέγουμε να πραγματοποιήσουμε μια εξέταση εφαρμογών (application only) στο URL *http://demo.testfire.net/* που αποτελεί την διεύθυνση που βρίσκεται η ιστοσελίδα προς εξέταση. Όπως περιγράψαμε στο προηγούμενο κεφάλαιο, σε αυτή την φάση το πρόγραμμα εξετάζει συνολικά την ιστοσελίδα για ευπάθειες διαδικτυακών

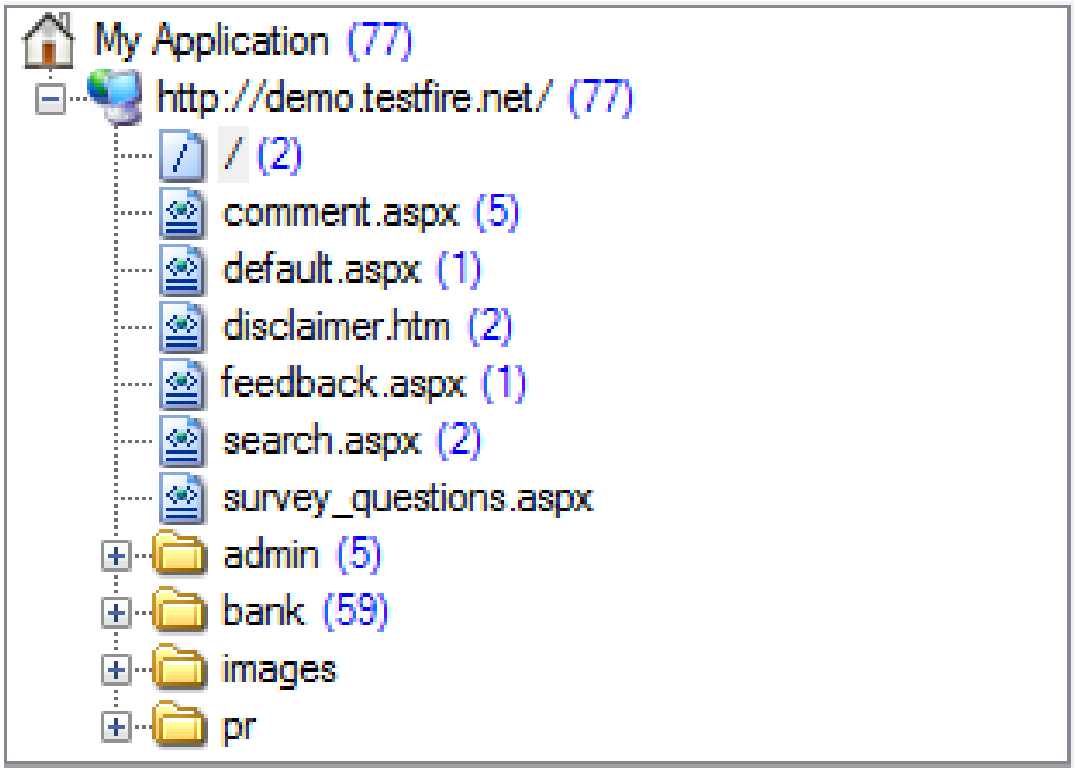
εφαρμογών. Αφού ολοκληρωθεί ο έλεγχος το πρόγραμμα μας εμφανίζει τον μετρητή αποτελεσμάτων που φαίνεται στην εικόνα 4.2. Χωρίζει τις ευπάθειες που αναγνώρισε σε τέσσερις κατηγορίες. Συνολικά βρέθηκαν εβδομήντα εφτά ευπάθειες εκ των οποίων οι τριάντα τέσσερις χαρακτηρίζονται ως υψηλού κινδύνου, οι δεκαεννέα ως μεσαίου κινδύνου, οι δεκατρείς ως χαμηλού κινδύνου και έντεκα ως παρατηρήσεις. Επιπροσθέτως φαίνεται ο αριθμός των URLs που επισκέφτηκε το πρόγραμμα κατά την εξέταση όπως επίσης και ο αριθμός των ελέγχων που πραγματοποίησε. Στην συγκεκριμένη εξέταση το Rational Appscan επισκέφτηκε πενήντα οχτώ URLs και ο συνολικός αριθμός των ελέγχων που πραγματοποίησε έφτασε τους έξι χιλιάδες εννιακόσους πενήντα εφτά. Παρατηρούμε ότι ο αριθμός των ελέγχων είναι αρκετά μεγάλος πράγμα που δηλώνει την εγκυρότητα των αποτελεσμάτων της εξέτασης.



Εικόνα 4.2 Μετρητής αποτελεσμάτων εξέτασης

Εκτός από την στατιστική σκοπιά της εξέτασης, κατά την διάρκεια της το Rational Appscan σχεδίασε και τον χάρτη πλοήγησης που σχηματίζουν τα URLs

της ιστοσελίδας. Με αυτόν τον τρόπο παρατηρούμε ποία μέρη της ιστοσελίδας πάσχουν από κάποια ευπάθεια και που συγκεντρώνεται το μεγαλύτερο μέρος των ευπαθειών. Κάτι τέτοιο βοηθάει τον υπεύθυνο για την ασφάλεια της ιστοσελίδας στο να κατανοήσει σε ποια μέρη της είναι αναγκαίο να δώσει την μεγαλύτερη σημασία, καθώς είναι πιο ευάλωτα. Στην εικόνα 4.3 παρατηρούμε τα αποτελέσματα της εξέτασης για την ιστοσελίδα της Altoro Mutual.

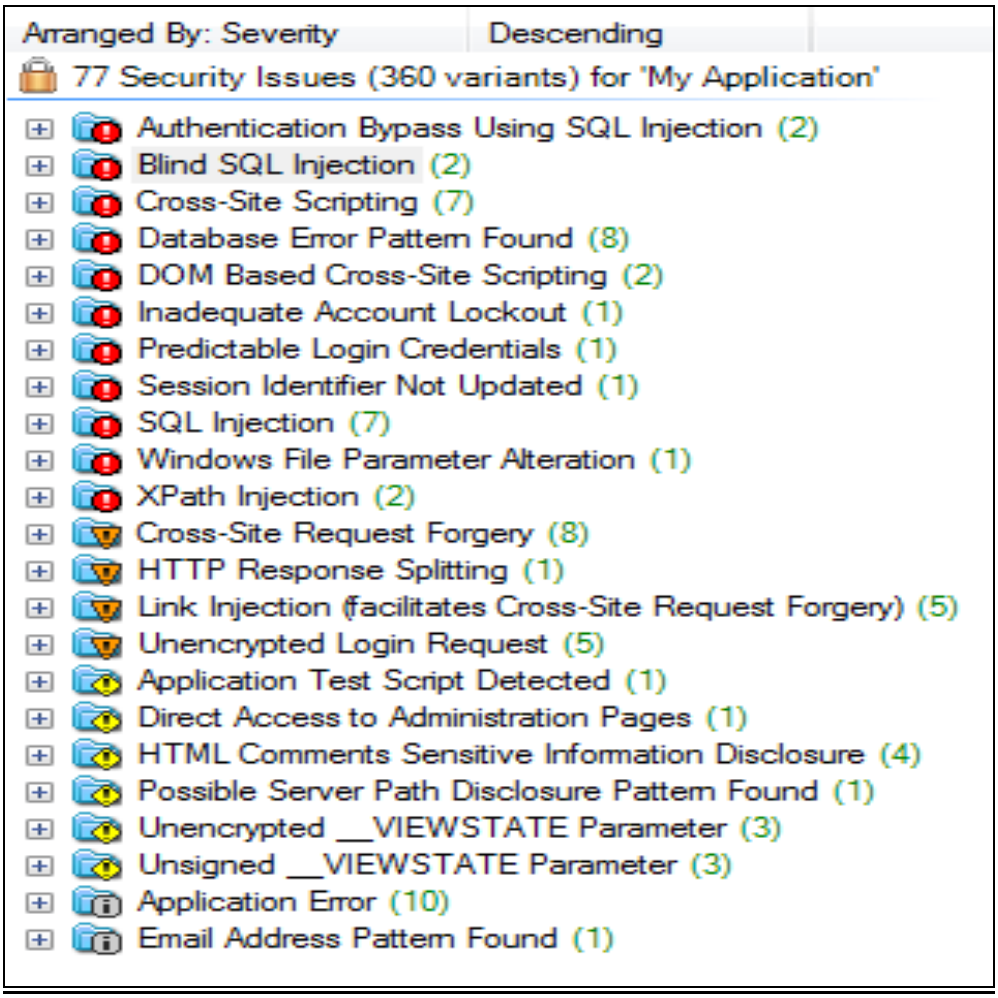


Εικόνα 4.3 Χάρτης πλοήγησης και ευπάθειες της ιστοσελίδας Altoro Mutual

Όπως βλέπουμε η πλειοψηφία των ευπαθειών σε αυτήν την ιστοσελίδα εντοπίζεται στο κομμάτι που αφορά τις λειτουργίες των τραπεζικών συναλλαγών (bank). Εφόσον το μεγαλύτερο κομμάτι των δεδομένων αποθηκεύονται και διαχειρίζονται από την διαδικτυακή εφαρμογή των τραπεζικών συναλλαγών κάτι τέτοιο είναι αναμενόμενο. Σε αυτό το κομμάτι της διαδικτυακής εφαρμογής βρίσκεται η διαδικασία πιστοποίησης των χρηστών, η διαχείριση των τραπεζικών λογαριασμών, η κατάθεση και ανάληψη χρημάτων, η μεταφορά χρημάτων και η εμφάνιση των πρόσφατων συναλλαγών. Όλες οι παραπάνω λειτουργίες αφορούν αυστηρώς προσωπικά δεδομένα και έχουν να κάνουν με τον πυρήνα των ενεργειών που ο χρήστης περιμένει από την συγκεκριμένη ιστοσελίδα. Από τις

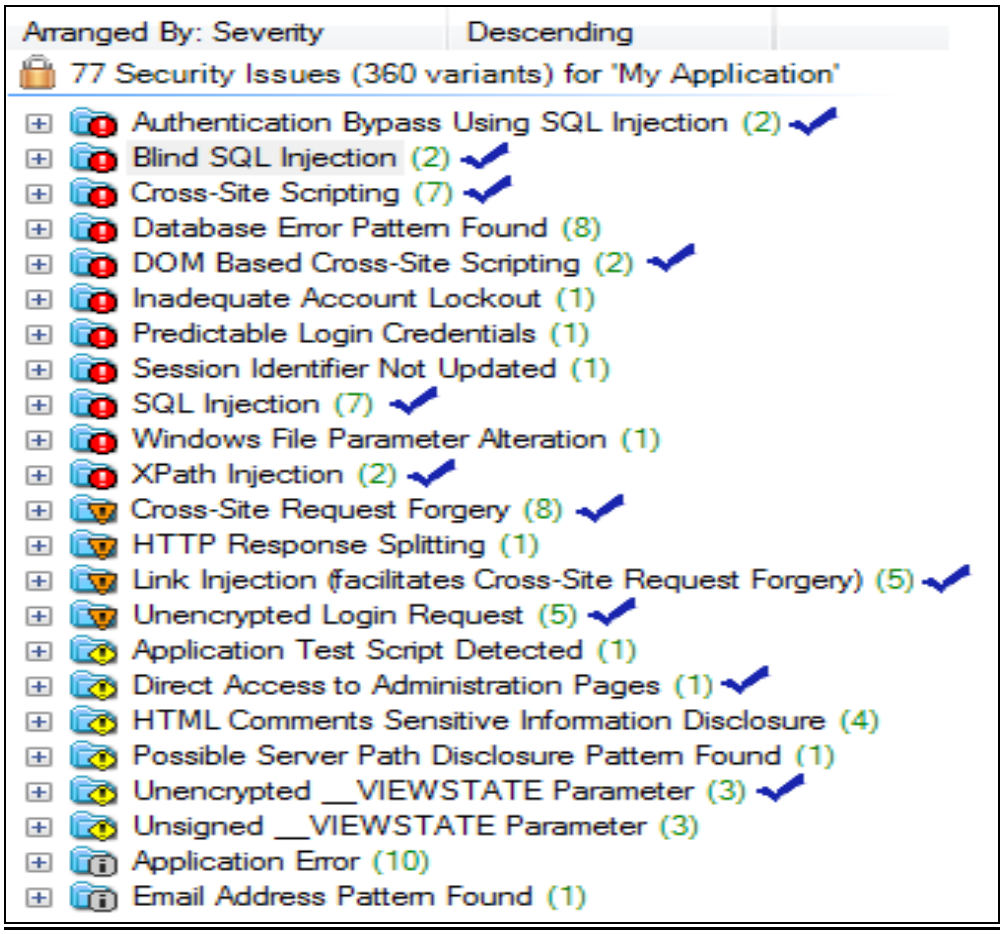
υπόλοιπες ευπάθειες ένας μικρός αριθμός αφορούν τις διευθύνσεις που έχουν πρόσβαση μόνο οι διαχειριστές της ιστοσελίδας. Είναι πολύ σημαντικό για μια ιστοσελίδα να διαχωρίζονται τα δικαιώματα πρόσβασης των χρηστών από αυτά των διαχειριστών. Μέτρα για αυτές τις ευπάθειες δικαιωμάτων πρόσβασης πρέπει να λαμβάνονται άμεσα από τους υπεύθυνους ασφάλειας, διότι εάν ένας κακόβουλος χρήστης καταφέρει να αποκτήσει πρόσβαση με τα δικαιώματα ενός διαχειριστή, η ζημιά στο Web πληροφοριακό σύστημα ή σε μια διαδικτυακή εφαρμογή θα είναι ολοκληρωτική. Τέλος μια μικρή ομάδα ευπαθειών εμφανίζεται στο αρχικό μέρος της ιστοσελίδας, προτού ο χρήστης πιστοποιηθεί και αρχίσει τις διαδικτυακές συναλλαγές. Σε αυτή την ομάδα ανήκουν ευπάθειες που βρίσκονται στην μηχανή αναζήτησης περιεχομένου, στην φόρμα επικοινωνίας των χρηστών με τους διαχειριστές της ιστοσελίδας και στους συνδέσμους της ιστοσελίδας που οδηγούν σε ιστοσελίδες τρίτων.

Έπειτα από την δημιουργία του χάρτη πλοήγησης το πρόγραμμα Rational Appscan μας εμφανίζει στο τέλος της εξέτασης μια λίστα με όλες τις ευπάθειες που εντόπισε, καθώς και τη συχνότητα εμφάνισης τους, ταξινομημένες με βάση την επικινδυνότητα τους. Με αυτόν τον τρόπο ο χρήστης του προγράμματος μετά την εξέταση αποκτά μια πλήρη εικόνα των ευπαθειών που καλείται να αντιμετωπίσει. Ακριβώς επειδή οι ευπάθειες είναι ταξινομημένες και ομαδοποιημένες από το πρόγραμμα με τον τρόπο αυτό, είναι ευκολότερο να ξεκινήσει κάποιος να αντιμετωπίζει την ίδια ευπάθεια σε όλα τα διαφορετικά σημεία της ιστοσελίδας που αυτή εμφανίζεται. Κάθε ευπάθεια που εμφανίζεται σε περισσότερα από ένα σημεία της ιστοσελίδας, ακόμα και αν έχει διαφορετική μορφή, αντιμετωπίζεται με παρόμοιο τρόπο. Στην εικόνα 4.4 παρουσιάζονται οι ευπάθειες της ιστοσελίδας Altoro Mutual όπως της κατέγραψε και τις ταξινόμησε το πρόγραμμα Rational Appscan.



Εικόνα 4.4 Πίνακας ευπαθειών της ιστοσελίδας Altoro Mutual

Παρατηρούμε από την εικόνα 4.4 ότι πολλές ευπάθειες που εντόπισε το πρόγραμμα Rational Appscan έχουν αναλυθεί στα προηγούμενα κεφάλαια της πτυχιακής εργασίας. Στην εικόνα 4.5 παρουσιάζονται οι ευπάθειες της ιστοσελίδας Altoro Mutual που έχουν ήδη αναλυθεί στο πρώτο κεφάλαιο ως οι δέκα πιο συχνά εμφανιζόμενες ευπάθειες σύμφωνα με το πρόγραμμα OWASP. Παρά το μεγάλο πλήθος των ευπαθειών που έχουν τα Web πληροφοριακά συστήματα, οι συχνότερα εμφανιζόμενες ευπάθειες, όπως αυτές που δημοσιεύτηκαν από το OWASP, θα αποτελούν πάντα κομμάτι του συνόλου των ευπαθειών των Web πληροφοριακών συστημάτων. Από αυτό συμπεραίνουμε τη σημαντικότητα που έχει για έναν μηχανικό πληροφορικής, που ασχολείται με θέματα ασφάλειας τέτοιων Web πληροφοριακών συστημάτων, η γνώση του τρόπου αντιμετώπισης αυτών των συχνά εμφανιζόμενων ευπαθειών.



Εικόνα 4.5 Κοινές ευπάθειες OWASP Top10 - Altoro Mutual

Οι κοινές ευπάθειες της ιστοσελίδας Altoro Mutual με τις ευπάθειες του OWASP είναι το Cross-Site Scripting, οι ευπάθειες τύπου injection, η πλαστογραφία αίτησης Cross-Site, η μη ασφαλής κρυπτογράφηση δεδομένων και η μη ασφαλής άμεση αναφορά σε αντικείμενο. Από τις δέκα ευπάθειες του OWASP οι πέντε εμφανίζονται και στην ιστοσελίδα.

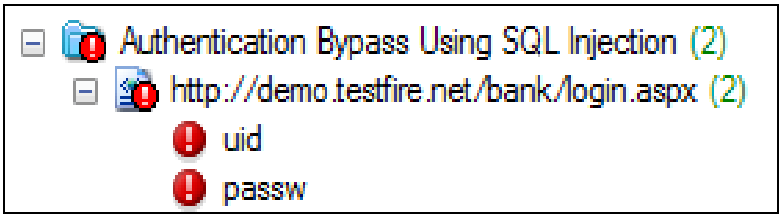
Με το τέλος της εξέτασης της ιστοσελίδας Altoro Mutual από το πρόγραμμα Rational Appscan, ο χρήστης έχει πλήρη εικόνα του πλήθους και της επικινδυνότητας των ευπαθειών που εντοπίστηκαν. Φτάνοντας σε αυτό το σημείο καλείται να επιλέξει τον τρόπο που θα αντιμετωπίσει τις ευπάθειες αυτές. Οι επιλογές του είναι να προσπαθήσει να εξαλείψει τις ευπάθειες αντιμετωπίζοντας τες μία προς μία σύμφωνα με τον πίνακα ευπαθειών ή βάση του χάρτη πλοήγησης που έχει δημιουργήσει το πρόγραμμα, να αντιμετωπίζει τις ευπάθειες σε κάθε τμήμα της ιστοσελίδας.

4.1.2 Ανάλυση των ευπαθειών της ιστοσελίδας Altoro Mutual

Με το πέρας της εξέτασης της ιστοσελίδας Altoro Mutual θα προχωρήσουμε στην ανάλυση των ευπαθειών που εντοπίστηκαν. Ξεκινώντας από τις ευπάθειες που χαρακτηρίζονται από το πρόγραμμα ως υψηλού κινδύνου, θα αναφέρεται το τμήμα της ιστοσελίδας στο οποίο ανιχνεύτηκαν, θα δίνεται μια περιγραφή της ευπάθειας και θα προτείνονται τρόποι με τους οποίους μπορούμε να τις αντιμετωπίσουμε.

4.1.2.1 Παράκαμψη πιστοποίησης με χρήση έκχυσης SQL (Authentication bypass using SQL Injection)

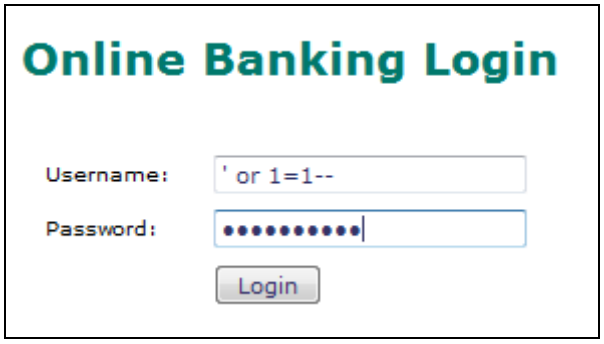
Η πρώτη ευπάθεια βάση κινδύνου με αλφαβητική ταξινόμηση που εντοπίστηκε από το πρόγραμμα, είναι η παράκαμψη της πιστοποίησης του συστήματος με την χρήση της έκχυσης SQL. Όπως αναφέραμε και στα προηγούμενα κεφάλαια η έκχυση SQL αποτελεί μια από τις συχνότερες ευπάθειες που εμφανίζονται στα Web πληροφοριακά συστήματα. Οι τρόποι που εκμεταλλεύεται αυτή η ευπάθεια από τους κακόβουλους χρήστες είναι πολυάριθμοι.



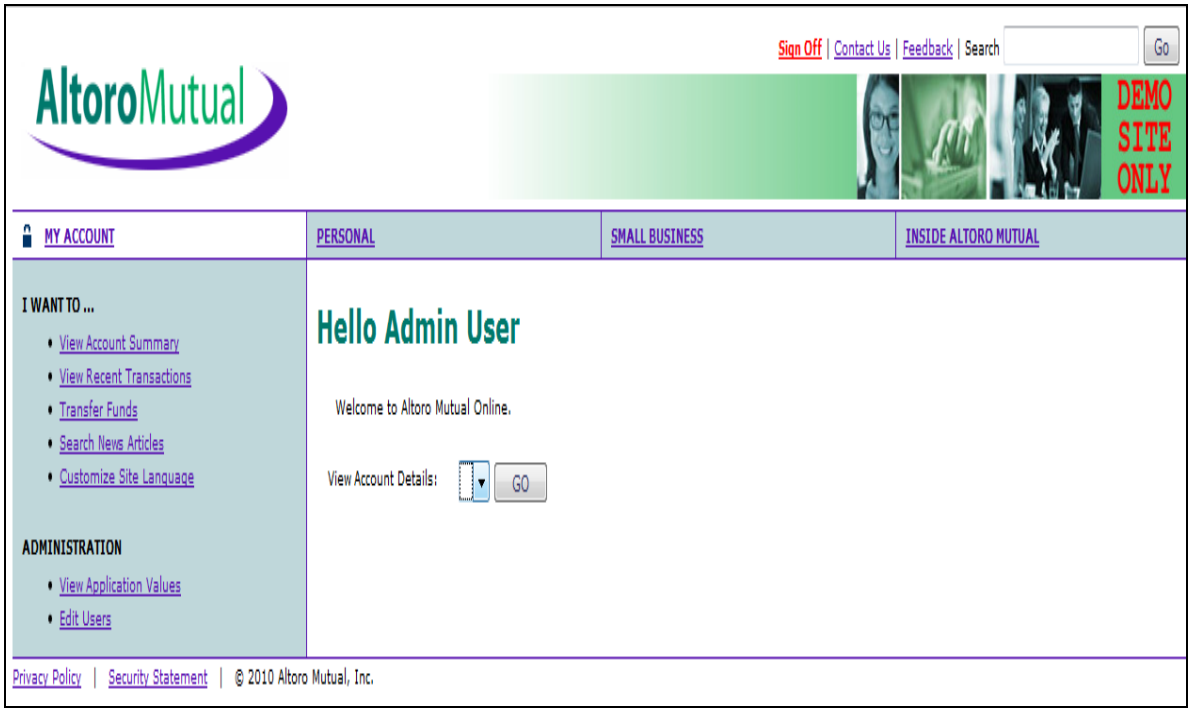
Εικόνα 4.6 Τμήματα ευπάθειας παράκαμψης πιστοποίησης με έκχυση SQL

Στην παραπάνω εικόνα 4.6 παρατηρούμε ότι η ευπάθεια εμφανίζεται στο κομμάτι της πιστοποίησης των χρηστών στην ιστοσελίδα και αφορά τα πεδία του ονόματος χρήστη (uid - user id) και του κωδικού πρόσβασης (passw - password). Η ευπάθεια εντοπίστηκε από το πρόγραμμα μετά από μια σειρά δοκιμών. Στις εικόνες 4.7 και 4.8 παρατηρούμε τα αποτελέσματα που έχει η πληκτρολόγηση ***‘ or***

1=1-- ως όνομα χρήστη και κωδικό πρόσβασης στα πεδία πιστοποίησης χρηστών της ιστοσελίδας. Το συγκεκριμένο αποτελεί χαρακτηριστικό παράδειγμα το οποίο περιγράφηκε στο κεφάλαιο 2.2.3 ως μορφή έκχυσης SQL.



Εικόνα 4.7 Επίθεση έκχυσης SQL στη φόρμα πιστοποίησης



Εικόνα 4.8 Είσοδος ως νόμιμος χρήσης με χρήση έκχυσης SQL

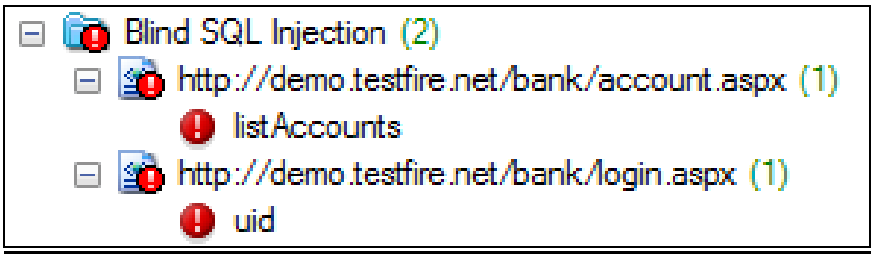
Με αυτόν τον τρόπο έκχυσης SQL πιστοποιούμαστε στο σύστημα ως ο πρώτος νόμιμος χρήστης στην βάση δεδομένων, στη συγκεκριμένη περίπτωση ως ο διαχειριστής, εφόσον τόσο το όνομα χρήστη όσο και ο κωδικός πρόσβασης είναι

αληθή. Όπως αντιλαμβανόμαστε κάτι τέτοιο έχει ολέθρια αποτελέσματα για την ασφάλεια της ιστοσελίδας.

Ο τρόπος αντιμετώπισης που προτείνει το Rational Appscan είναι ο έλεγχος των χαρακτήρων που εισάγονται στην φόρμα πιστοποίησης, ώστε να απαγορεύονται συγκεκριμένοι χαρακτήρες όπως: | (κάθετη παύλα), & (και), $ (σύμβολο δολαρίου), % (σύμβολο επί τις εκατό), ; (ερωτηματικό), ‘ (μονή απόστροφος), “ (quotation mark) , + (το σύμβολο της πρόσθεσης) και άλλοι χαρακτήρες που δυνητικά είναι επικίνδυνοι.

4.1.2.2 Τυφλή έκχυση SQL (Blind SQL Injection)

Επόμενη ευπάθεια της ιστοσελίδας είναι η τυφλή έκχυση SQL. Η ευπάθεια αυτή δημιουργείται από τα διαφορετικά μηνύματα σφάλματος (error messages) που εμφανίζει η ιστοσελίδα.

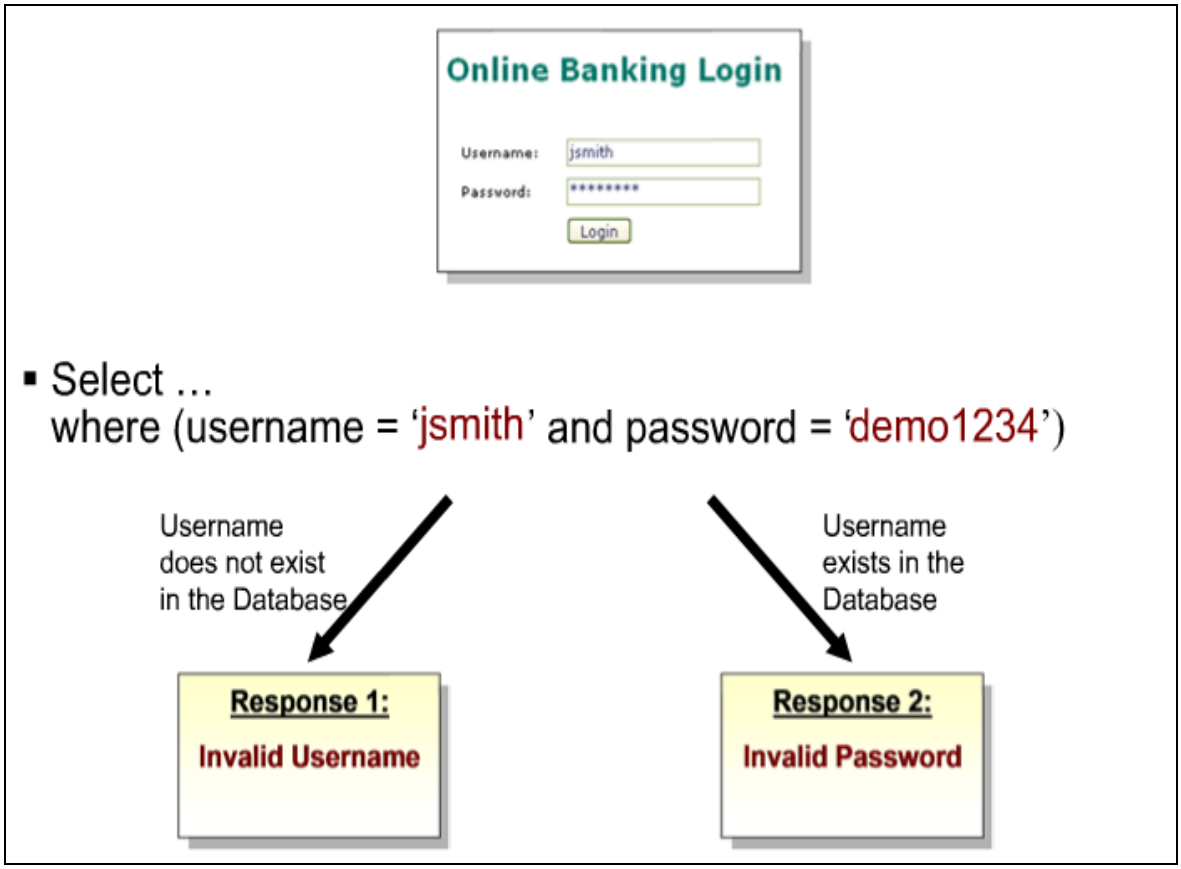


Εικόνα 4.9 Τμήματα ευπάθειας τυφλής έκχυσης SQL

Στην εικόνα 4.9 παρατηρούμε πως η ευπάθεια εμφανίζεται στο σύστημα πιστοποίησης και στο κομμάτι του λογαριασμού ενός πιστοποιημένου χρήστη. Ο λόγος που εντοπίζεται η ευπάθεια από το πρόγραμμα, είναι η διαφορετική αντίδραση, υπό μορφή μηνυμάτων σφάλματος, της ιστοσελίδας. Για παράδειγμα όταν ένας κακόβουλος χρήστης πληκτρολογήσει εσφαλμένο όνομα χρήστη και κωδικό πρόσβασης τότε η απάντηση της ιστοσελίδας θα είναι “Λανθασμένο όνομα χρήστη”. Στην περίπτωση όμως που τυχαία ο κακόβουλος χρήστης εισάγει όνομα χρήστη που υπάρχει καταχωρημένο, τότε το μήνυμα σφάλματος αλλάζει σε “Λανθασμένος κωδικός πρόσβασης”. Ένας κακόβουλος χρήστης όταν εντοπίσει

την συγκεκριμένη ευπάθεια μπορεί να εισάγει ερωτήματα SQL στο πεδίο του ονόματος χρήστη, τα οποία αφορούν πληροφορίες άλλων πινάκων της βάσης δεδομένων. Με αυτό τον τρόπο εάν λάβει το μήνυμα λάθους "Λανθασμένος κωδικός πρόσβασης" γνωρίζει πως τα στοιχεία που εισήγαγε στο πεδίο του ονόματος χρήστη είναι αληθή. Στο σχήμα 4.10 φαίνονται οι διαφορετικές αντιδράσεις της ιστοσελίδας.

Ο τρόπος αντιμετώπισης αυτής της ευπάθειας είναι η τροποποίηση των μηνυμάτων σφάλματος που επιστρέφει η ιστοσελίδα έτσι ώστε ο κακόβουλος χρήστης να μην μπορεί να τα χρησιμοποιήσει για να εξορύξει επιπλέον πληροφορίες για το σύστημα.



Εικόνα 4.10 Διαφορετικά μηνύματα σφάλματος

4.1.2.3 Cross-Site Scripting (XSS)

Η ευπάθεια Cross-Site Scripting η οποία θεωρείται ως μια από της συχνότερες στα Web πληροφοριακά συστήματα παγκοσμίως εντοπίζεται και στην ιστοσελίδα Altoro Mutual από το πρόγραμμα. Το πρόγραμμα βρίσκει επτά σημεία τις ιστοσελίδας στα οποία υπάρχει η δυνατότητα να χρησιμοποιηθεί η ευπάθεια αυτή από κακόβουλους χρήστες. Αυτά φαίνονται στην εικόνα 4.11 και είναι η λειτουργία της επιλογής γλώσσας για την διαδικτυακή εφαρμογή, το πεδίο συνθηματικού χρήστη στη φόρμα πιστοποίησης, η λειτουργία μεταφοράς χρημάτων, η φόρμα επικοινωνίας και η μηχανή αναζήτησης περιεχομένου.

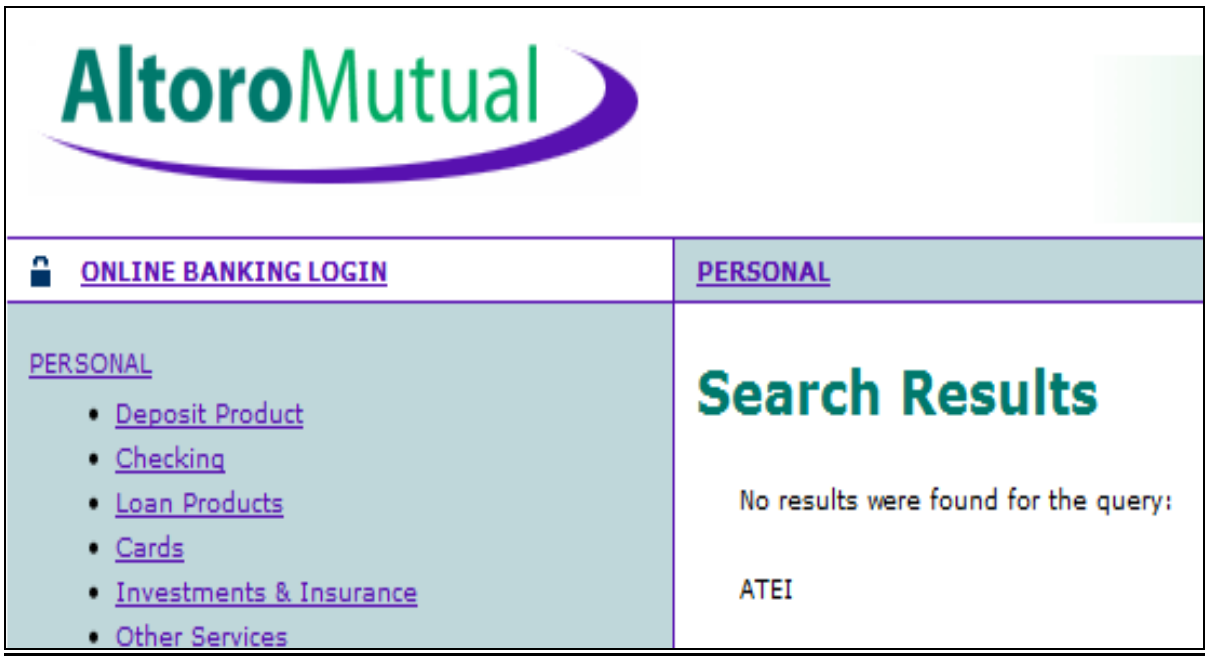
Εικόνα 4.11 Τμήματα ευπάθειας Cross-Site Scripting

Στα παραπάνω σημεία η ευπάθεια εντοπίστηκε με διαδοχικές “επιθέσεις” του προγράμματος στην ιστοσελίδα, μιμούμενο κάποιον κακόβουλο χρήστη. Ο τρόπος εκμετάλλευσης της ευπάθειας είναι η χρήση εμφωλευμένου JavaScript ή VBScript κώδικα. Ο εμφωλευμένος κώδικας εισάγεται από τον κακόβουλο χρήστη είτε σε ένα πεδίο που επιτρέπει εισαγωγή δεδομένων, είτε απευθείας στο URL της ιστοσελίδας. Με αυτόν τον τρόπο σε περίπτωση που ο εμφωλευμένος κώδικας

εισάγεται ως έχει, στον πηγαίο κώδικα της ιστοσελίδας, αυτό σημαίνει πως δεν έχουν ληφθεί τα απαραίτητα μέτρα φιλτραρίσματος για την εισαγωγή δεδομένων. Εάν επιστραφεί κάποιο μήνυμα, συνηθίζεται η χρήση alert() στον εμφωλευμένο κώδικα, τότε συμπεραίνουμε ότι δεν έχουν ληφθεί τα απαραίτητα μέτρα φιλτραρίσματος και για την επιστροφή αποτελεσμάτων. Ο τρόπος με τον οποίο εκμεταλλεύεται ο κακόβουλος χρήστης την ευπάθεια, αφού την εντοπίσει, είναι η αποστολή του συνδέσμου με τον εμφωλευμένο κώδικα σε διευθύνσεις ηλεκτρονικού ταχυδρομείου (e-mail phishing) προσποιούμενος γνήσια διεύθυνση ηλεκτρονικού ταχυδρομείου ή ανάρτηση του συνδέσμου σε ιστοσελίδες και φόρουμ ώστε να παγιδέψει τα υποψήφια θύματα. Με την εκτέλεση αυτού του εμφωλευμένου κώδικα, υπάρχει η δυνατότητα υποκλοπής των session tokens των νόμιμων χρηστών, των cookies και άλλων προσωπικών πληροφοριών. Έπειτα μπορούν να χρησιμοποιηθούν από τον κακόβουλο χρήστη, για να μιμηθεί τον νόμιμο χρήστη και να ταυτοποιηθεί στη θέση του στο Web πληροφοριακό σύστημα. Στις εικόνες 4.12 ως 4.16 βλέπουμε ένα παράδειγμα εισαγωγής εμφωλευμένου κώδικα στην ιστοσελίδα Altoro Mutual.

Το παράδειγμα αρχίζει με την αναζήτηση στην μηχανή αναζήτησης περιεχομένων της ιστοσελίδας για την φράση ATEI. Έπειτα παρατηρούμε πως το κείμενο προς αναζήτηση προστέθηκε μέσα στον κώδικα της ιστοσελίδας. Εάν αντί για ATEI πληκτρολογήσουμε εμφωλευμένο κώδικα, για παράδειγμα **<script>alert(55555)</script>** θα δούμε ότι εκτελείτε και επιστρέφει ως μήνυμα το 55555. Αυτό δείχνει πως η σελίδα πάσχει από την ευπάθεια Cross-Site Scripting. Όλα τα τμήματα της ιστοσελίδας στα οποία το πρόγραμμα εντόπισε την ευπάθεια είναι ευάλωτα με τον ίδιο τρόπο στους κακόβουλους χρήστες.

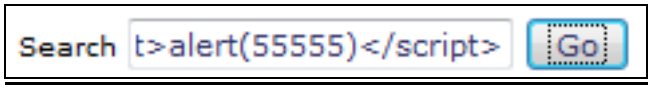
Εικόνα 4.12 Πραγματοποίηση αναζήτησης



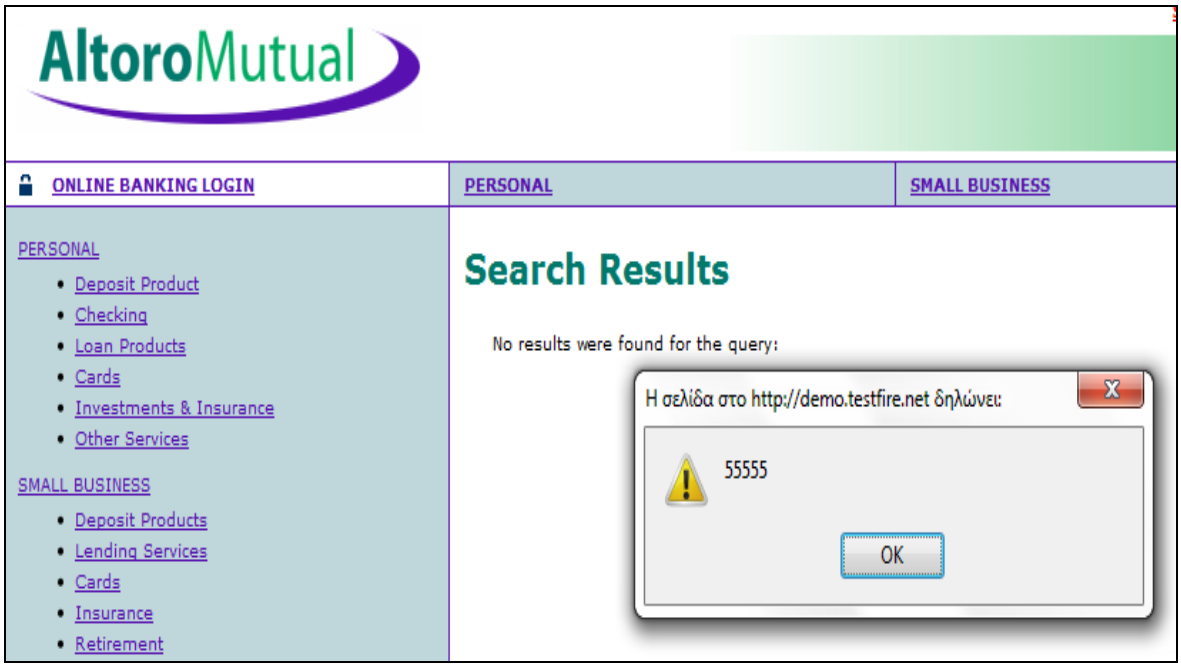
Εικόνα 4.13 Αποτέλεσμα αναζήτησης

```
<div class="fl" style="width: 99%;">

<h1>Search Results</h1>

<p>No results were found for the query:<br /><br />
<span id="_ctl0__ctl0_Content_Main_lblSearch">ATEI</span></p>

</div>
```

Εικόνα 4.14 Πηγαίος κώδικας σελίδας αποτελεσμάτων αναζήτησης



Εικόνα 4.15 Εισαγωγή εμφωλευμένου κώδικα στην μηχανή αναζήτησης



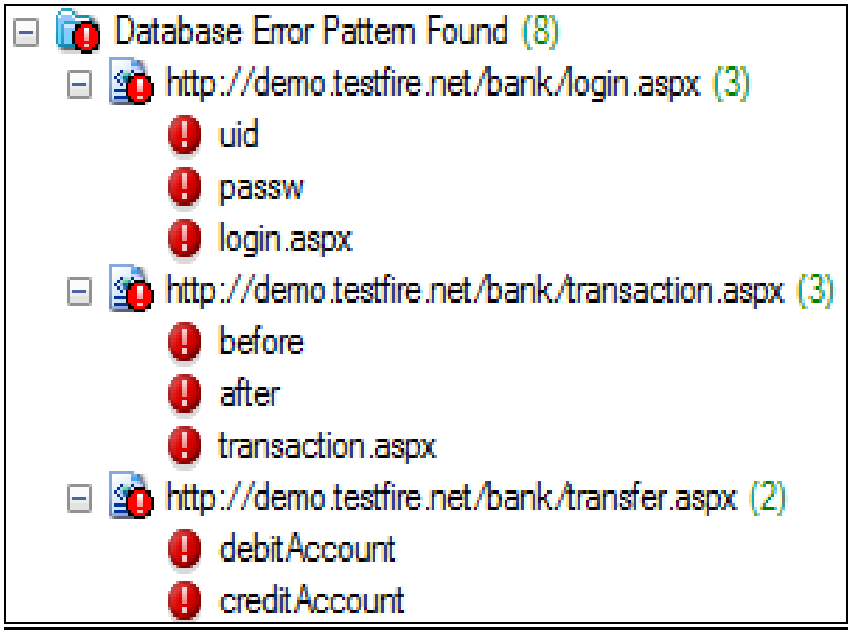
Εικόνα 4.16 Αποτέλεσμα εκτέλεσης εμφωλευμένου κώδικα

Σε πρόσφατες έρευνες οι δυνατότητες τέτοιων ευπαθειών Cross-Site Scripting έχουν ως αποτέλεσμα οι κακόβουλοι χρήστες να παίρνουν τον ολοκληρωτικό έλεγχο του φυλλομετρητή του θύματος μαζί με πληροφορίες όπως ιστοσελίδες που επισκέφτηκε, καταχωρίσεις σε φόρουμ και το ιστορικό του φυλλομετρητή. Επίσης μπορούν να αναγκάσουν τον χρήστη στο να συμμετέχει σε επιθέσεις άρνησης εξυπηρέτησης (denial of service) και σάρωσης θυρών (port scanning).

4.1.2.4 Αποκάλυψη μοτίβου μηνυμάτων σφάλματος της βάσης δεδομένων (Database Error Pattern Found)

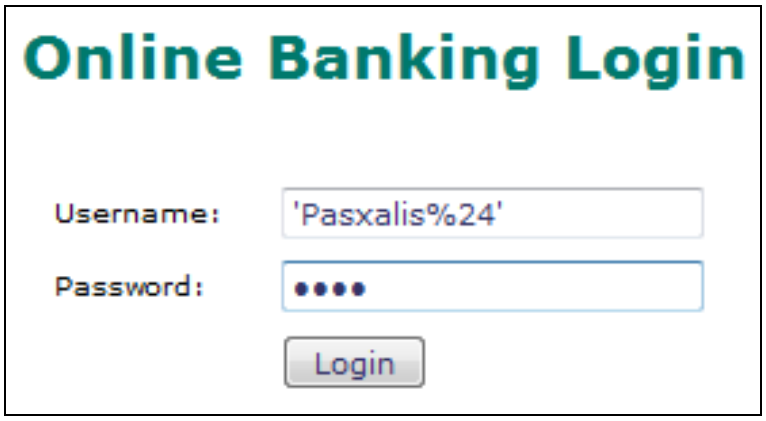
Ακόμα μια ευπάθεια που επηρεάζει την ιστοσελίδα Altoro Mutual σε περισσότερα από ένα σημεία, είναι η αποκάλυψη του μοτίβου των μηνυμάτων σφάλματος της βάσης δεδομένων. Κύριο χαρακτηριστικό αυτής της ευπάθειας είναι η επιστροφή μηνυμάτων σφάλματος σε εισαγωγή δεδομένων λανθασμένης σύνταξης, απευθείας από την βάση δεδομένων. Από αυτό συμπεραίνουμε ότι η ιστοσελίδα σε εκείνα τα τμήματα της είναι ευάλωτη σε επιθέσεις έκχυσης SQL, λόγω του ότι τα εισαγόμενα δεδομένα έφτασαν μέχρι τον έλεγχο της βάσης

δεδομένων. Στην εικόνα 4.17 φαίνονται τα τμήματα της ιστοσελίδας που εντοπίζεται η ευπάθεια.

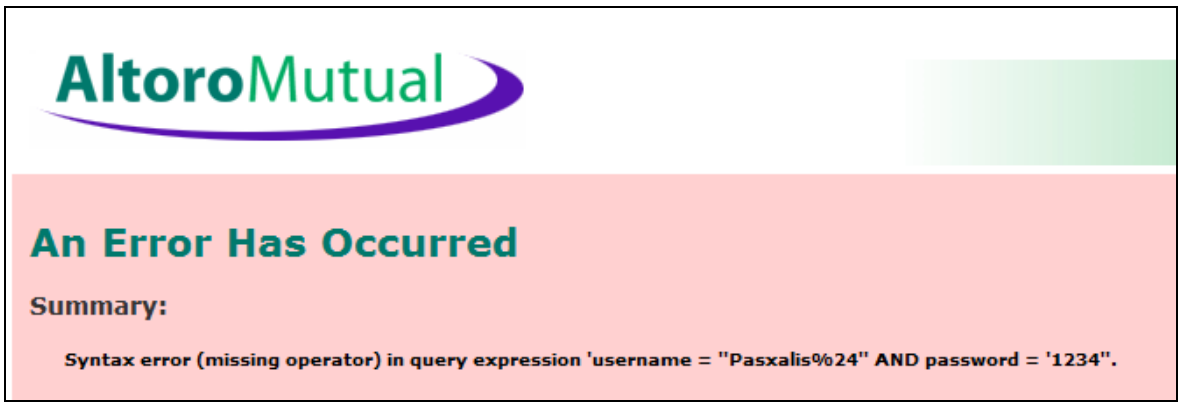


Εικόνα 4.17 Τμήματα ευπάθειας Database Error Pattern Found

Ένα παράδειγμα για την κατανόηση του τρόπου εντοπισμού αυτής της ευπάθειας από έναν κακόβουλο χρήστη, είναι να δώσει στην φόρμα πιστοποίησης στο πεδίο του ονόματος χρήστη μια σειρά από χαρακτήρες που περιέχουν και ειδικούς χαρακτήρες όπως το **%** (επί τις εκατό) ή **‘** (μονή απόστροφος) π.χ. **‘Pasxalis%24’** και ένα κωδικό πρόσβασης με προβλεπόμενους χαρακτήρες όπως π.χ. **1234**. Όταν η σελίδα εμφανίζει το μήνυμα σφάλματος, στον πηγαίο κώδικα βλέπουμε τις πληροφορίες που δηλώνουν πως το μήνυμα σφάλματος προέρχεται από την βάση δεδομένων και δεν είναι μέρος της εξορισμού απάντησης της ιστοσελίδας σε εισαγωγή λανθασμένων δεδομένων.



Εικόνα 4.18 Εισαγωγή δεδομένων με ειδικούς χαρακτήρες στην φόρμα πιστοποίησης



Εικόνα 4.19 Εμφάνιση μηνύματος λάθους στην ιστοσελίδα

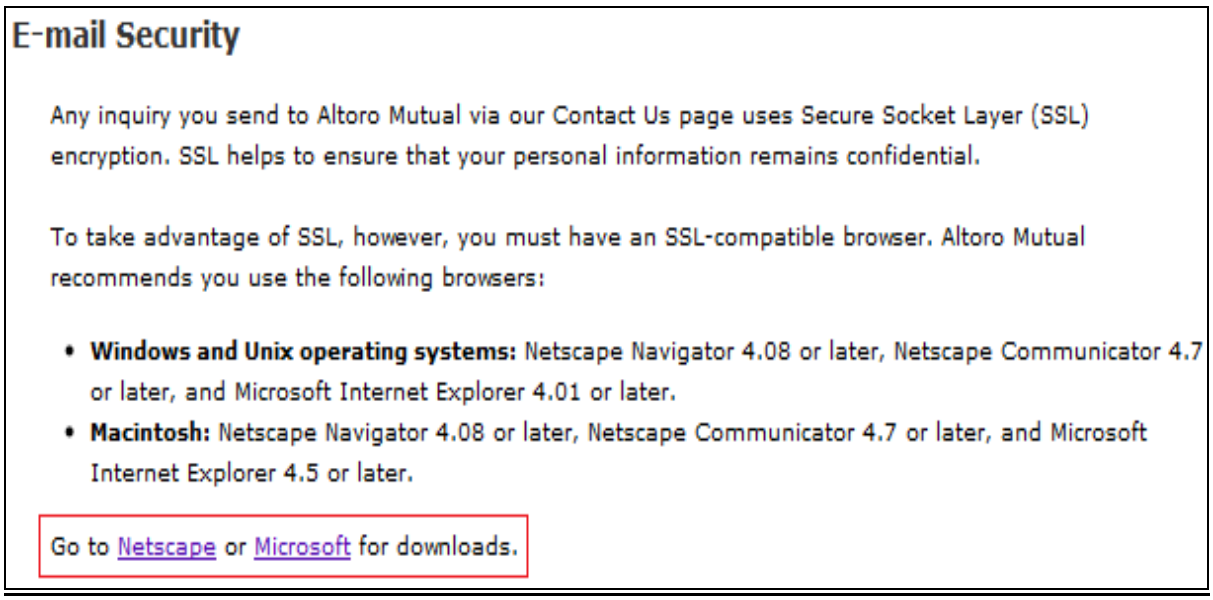
```
<h1>An Error Has Occurred</h1>

<h2>Summary:</h2>

<p><b><span id="_ctl0_Content_lblSummary">Syntax error (missing operator) in query expression 'username = ''Pasxalis%24'' AND
</span></b></p>

<h2>Error Message:</h2>

<p><span id="_ctl0_Content_lblDetails">System.Data.OleDb.OleDbException: Syntax error (missing operator) in query expression
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextForSingleResult(tagDBPARAMS dbParams, Object&amp; executeResult)
```

Εικόνα 4.20 Μήνυμα σφάλματος βάσης δεδομένων (OleDbException)

Ο τρόπος αντιμετώπισης αυτής της ευπάθειας είναι η απαγόρευση της εισαγωγής ειδικών χαρακτήρων που μπορούν να χρησιμοποιηθούν για ερωτήματα SQL στα πεδία εισαγωγής δεδομένων.

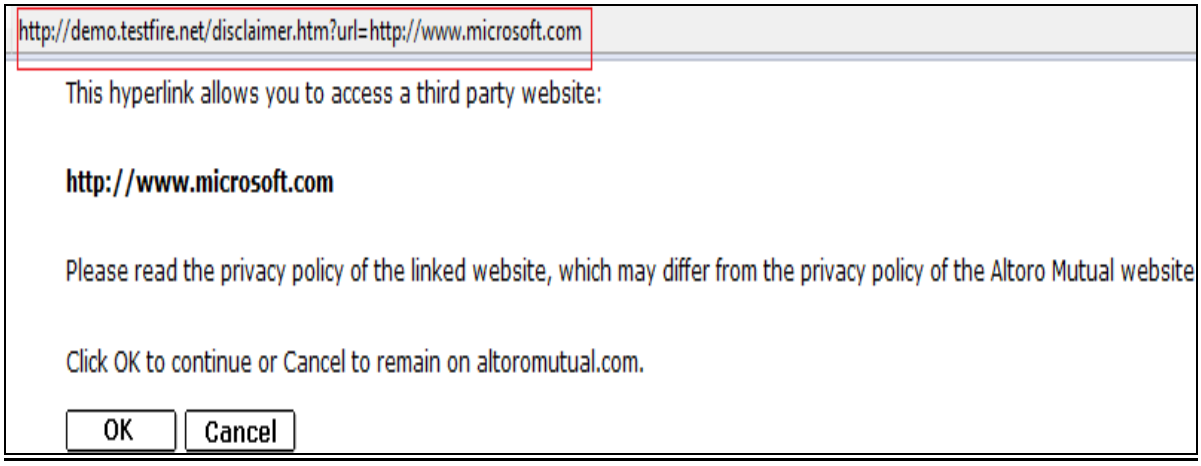
4.1.2.5 Cross-Site Scripting με βάση το DOM (DOM Based Cross-Site Scripting)

Μια παραλλαγή της ευπάθειας Cross-Site Scripting είναι αυτή που βασίζεται στο DOM[7] (Document Object Model). Η ευπάθεια εντοπίζεται στην ιστοσελίδα όταν εισάγονται δεδομένα από ένα αντικείμενο DOM, για να την αλλάξουν δυναμικά. Εάν αυτά τα δεδομένα δεν ελέγχονται τότε η ιστοσελίδα είναι ευάλωτη. Στο σενάριο χρήσης της ιστοσελίδας Altoro Mutual αυτή η ευπάθεια εντοπίστηκε στους συνδέσμους τρίτων ιστοσελίδων που περιείχε στην φόρμα επικοινωνίας. Μέσω του document.URL που είναι αντικείμενο του DOM ο κακόβουλος χρήστης μπορεί να εισάγει ένα κομμάτι κώδικα στο URL και να επιτύχει μια επίθεση Cross-Site Scripting με τα αποτελέσματα που περιγράψαμε στο κεφάλαιο 4.1.2.3. Στις παρακάτω εικόνες φαίνεται το παράδειγμα εκμετάλλευσης αυτής της ευπάθειας.



Εικόνα 4.21 Σύνδεσμοι τρίτων ιστοσελίδων

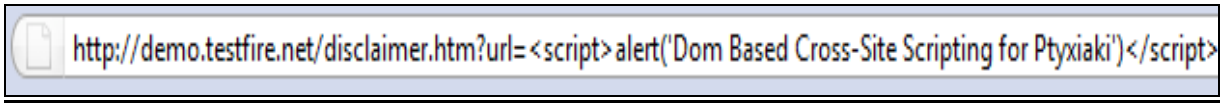
[7] http://en.wikipedia.org/wiki/Document_Object_Model



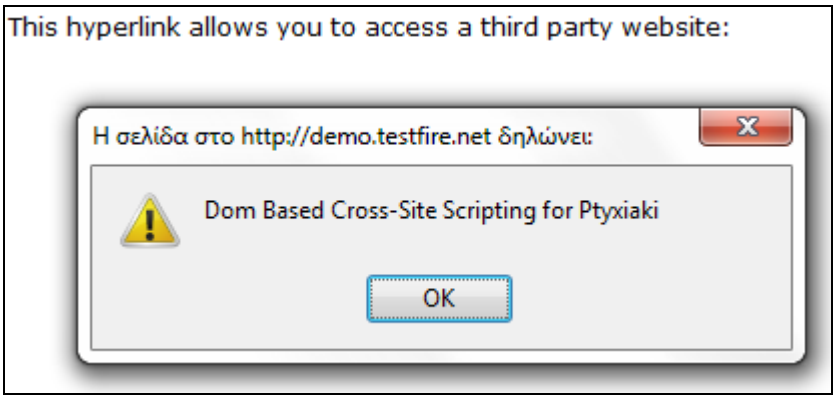
Εικόνα 4.22 Νέο παράθυρο για έλεγχο απομάκρυνσης από την ιστοσελίδα

```
    var iPos = document.URL.indexOf("url=")+4;
    var sDst = document.URL.substring(iPos,document.URL.length);
  </script>
</head>

<body bgcolor=#FFFFFF link="#5811B0" vlink="#5811B0" leftmargin="0"

  <center>
  <table width=90% border=0>
    <tr>
      <td>
        <p>This hyperlink allows you to access a third party websit
        <br /><br />
        <b><script>document.write(unescape(sDst));</script></b>
```

Εικόνα 4.23 Ο κώδικας του νέου παραθύρου κάνει χρήση του document.URL με παράμετρο το url



Εικόνα 4.24 Εισαγωγή JavaScript στην παράμετρο url του παραθύρου ελέγχου

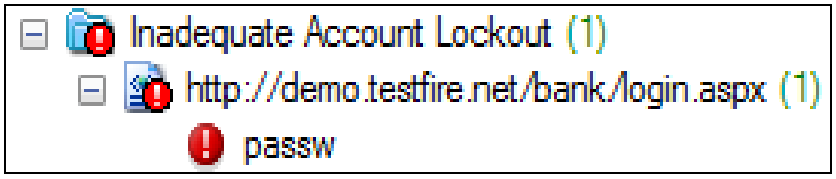


Εικόνα 4.25 Αποτέλεσμα εκτέλεσης JavaScript

Ο τρόπος αντιμετώπισης της ευπάθειας είναι ο ίδιος με αυτόν της ευπάθειας Cross-Site Scripting, δηλαδή φιλτράρισμα και έλεγχος των δεδομένων εισόδου.

4.1.2.6 Ανεπαρκής έλεγχος προσπαθειών εισόδου (Inadequate Account Lockout)

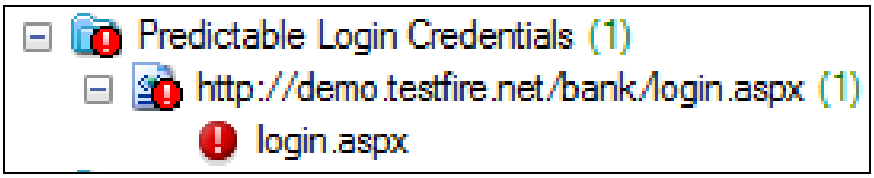
Με τον όρο ανεπαρκής έλεγχος προσπαθειών εισόδου ονομάζουμε μια συγκεκριμένη ευπάθεια, η οποία εντοπίστηκε στην ιστοσελίδα, που αφορά τον έλεγχο που γίνεται στη πιστοποίηση ενός χρήστη στο σύστημα. Για την αποφυγή επιθέσεων με χρήση της τεχνικής ωμής βίας (brute force attacks) ο τρόπος πιστοποίησης πρέπει να ελέγχεται. Ο έλεγχος πρέπει να περιλαμβάνει προσωρινή απαγόρευση εισαγωγής δεδομένων στη φόρμα πιστοποίησης, μετά από μια σειρά αποτυχημένων προσπαθειών. Ένας λογικός αριθμός είναι τρείς ως πέντε προσπάθειες. Όπως παρατηρούμε στην εικόνα 4.26 η ευπάθεια εντοπίζεται στη σελίδα πιστοποίησης (login) της ιστοσελίδας Altoro Mutual.



Εικόνα 4.26 Τμήματα ευπάθειας Inadequate Account Lockout

4.1.2.7 Προβλέψιμα στοιχεία πιστοποίησης (Predictable Login Credentials)

Η ευπάθεια αυτή προκύπτει από την ευκολία που υπάρχει στον να εικάσει ένας κακόβουλος χρήστης τα στοιχεία πιστοποίησης, όνομα χρήστη και κωδικό πρόσβασης, ενός νόμιμου χρήστη του συστήματος. Σε πολλά Web πληροφοριακά συστήματα υπάρχουν εσφαλμένως κάποια εξ ορισμού στοιχεία πιστοποίησης που δεν έχουν αλλαχθεί όπως θα έπρεπε για λόγους ασφάλειας. Τέτοια παραδείγματα είναι το admin/admin για πιστοποίηση με δικαιώματα διαχειριστή στο σύστημα και guest/guest για είσοδο κάποιου επισκέπτη. Κάθε σύστημα θα πρέπει να ελέγχει αυτά τα στοιχεία και να περιορίζει τους χρήστες, ώστε να μην εισάγουν αδύναμα στοιχεία πιστοποίησης από θέμα ασφάλειας. Υπάρχουν διάφορες τεχνικές ώστε να σιγουρευτεί η ασφάλεια των στοιχείων πιστοποίησης από τέτοια ευπάθεια. Ορισμένες από αυτές περιλαμβάνουν, το μέγεθος του κωδικού πρόσβασης να περιλαμβάνει περισσότερους χαρακτήρες από κάποιον αριθμό (π.χ. 6), να υπάρχει διαχωρισμός μεταξύ πεζών και κεφαλαίων γραμμάτων, να μην περιέχει στον κωδικό πρόσβασης κομμάτι του ονόματος χρήστη, να περιέχεται στα στοιχεία πιστοποίησης μια αναλογία από αριθμούς και χαρακτήρες (π.χ. τουλάχιστον ένας αριθμός και ένα γράμμα) και τέλος να γίνεται αλλαγή κωδικού πρόσβασης ανά τακτά χρονικά διαστήματα. Στην εικόνα 4.27 φαίνεται το τμήμα της ιστοσελίδας στο οποίο εντοπίστηκε η ευπάθεια.

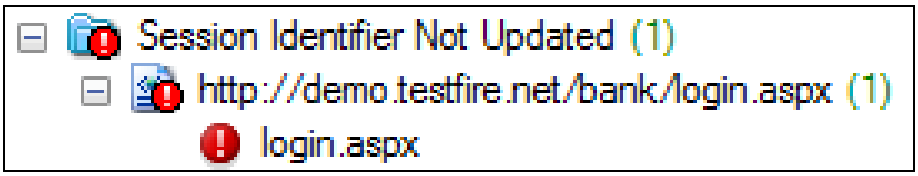


Εικόνα 4.27 Τμήματα ευπάθειας προβλέψιμων στοιχείων πρόσβασης

4.1.2.8 Μη ανανεωμένα αναγνωριστικά συνόδου (Session Identifier Not Updated)

Κάθε φορά που ένας χρήστης επικοινωνεί με το σύστημα, ο τρόπος επικοινωνίας χαρακτηρίζεται από μια σειρά αιτημάτων και απαντήσεων. Αίτημα

ονομάζουμε την είσοδο από τον χρήστη και απάντηση την απόκριση του συστήματος στο συγκεκριμένο αίτημα. Η ευπάθεια των μη ανανεωμένων αναγνωριστικών συνόδου που εντόπισε το πρόγραμμα στην ιστοσελίδα αφορά την πιστοποίηση χρηστών. Υπάρχουν δύο τρόποι διαχείρισης συνόδων ως προς το αναγνωριστικό (ID) τους. Ο πρώτος τρόπος είναι αυτός των “ανεκτικών” συστημάτων όπου το αναγνωριστικό συνόδου (Session ID) καθορίζεται από τον φυλλομετρητή ιστού του χρήστη και ο δεύτερος τρόπος των “αυστηρών” συστημάτων που επιτρέπουν μόνο αναγνωριστικά που δημιουργούνται από την μεριά του εξυπηρετητή (Server-side). Εάν ένας κακόβουλος χρήστης, με ένα είδος επίθεσης όπως Cross-site Scripting, μπορέσει να υποκλέψει το αναγνωριστικό συνόδου, τότε θα αποκτήσει πρόσβαση στο σύστημα μιμούμενος τον νόμιμο χρήστη. Για αυτόν τον λόγο πρέπει να ληφθούν απαραίτητα μέτρα στο σύστημα που να αποτρέπουν τέτοιες ενέργειες. Σε κάθε αίτηση πιστοποίησης τα αναγνωριστικά πρέπει να αλλάζουν σε σχέση με την τιμή που είχαν πριν την πιστοποίηση του χρήστη στο σύστημα. Επίσης θα πρέπει να μην επιτρέπεται η αλλαγή του αναγνωριστικού συνόδου από τους χρήστες. Τέλος πιο ασφαλές είναι το να μην χρησιμοποιούμε τον “ανεκτικό” τρόπο διαχείρισης αναγνωριστικών συνόδου.

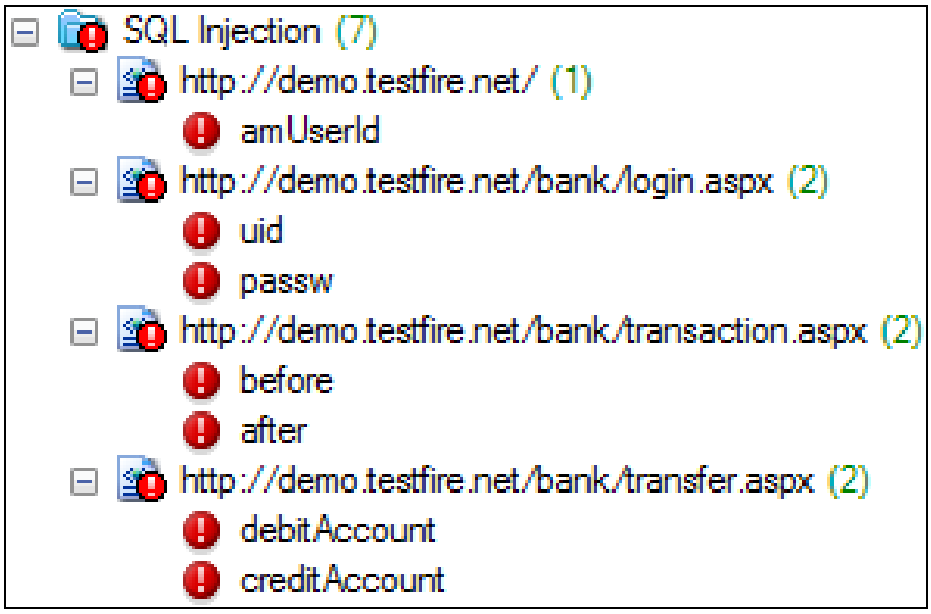


Εικόνα 4.28 Τμήματα ευπάθειας Session Identifier Not Updated

4.1.2.9 Έκχυση SQL (SQL injection)

Μια μεγάλη κατηγορία ευπαθειών σχετίζονται με την έκχυση SQL. Είτε πραγματοποιείται τυφλά είτε έπειτα από συγκέντρωση πληροφοριών, η ευπάθεια έκχυσης SQL είναι από τις σοβαρότερες που ταλαιπωρούν τα σύγχρονα Web πληροφοριακά συστήματα. Όπως είδαμε στην αρχή της ανάλυσης των ευπαθειών της ιστοσελίδας Altoro Mutual, η ευπάθεια 4.1.2.1 με όνομα παράκαμψη πιστοποίησης με χρήση έκχυσης SQL, αποτελεί υποκατηγορία της ευπάθειας

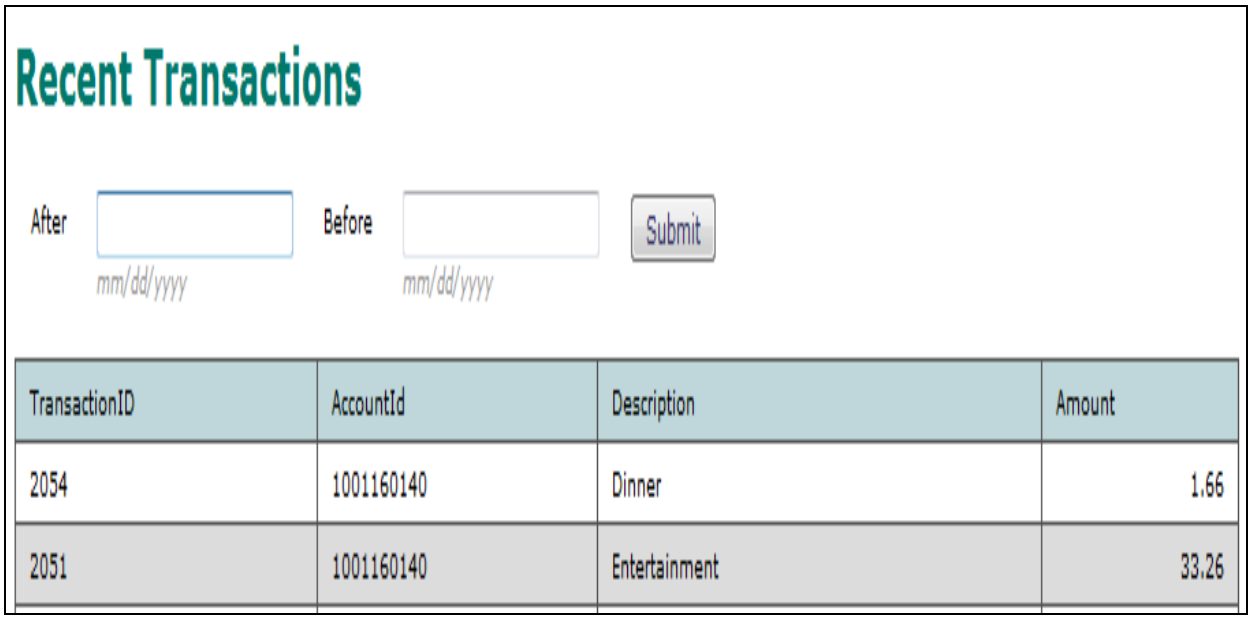
έκχυσης SQL. Υπάρχουν πολλές δυνατότητες με τις οποίες ένας κακόβουλος χρήστης μπορεί να εκμεταλλευτεί αυτήν την ευπάθεια. Εκτός της παράκαμψης της πιστοποίησης θα δώσουμε ένα παράδειγμα όπου ο κακόβουλος χρήστης πετυχαίνει να συλλέξει πολύτιμες πληροφορίες με εξαιρετικά μεγάλες συνέπειες για το σύστημα. Στην εικόνα 4.29 παρατηρούμε τα μέρη της ιστοσελίδας που εντοπίζεται η ευπάθεια.



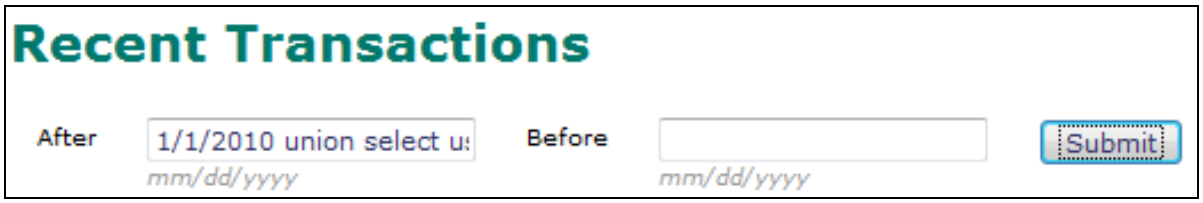
Εικόνα 4.29 Τμήματα ευπάθειας έκχυσης SQL

Στο παράδειγμα που θα περιγραφεί αντιστοιχούν οι εικόνες 4.30 ως 4.32 για καλύτερη κατανόηση του αναγνώστη. Το παράδειγμα έχει να κάνει με το κομμάτι της ιστοσελίδας που αφορά την εμφάνιση των τελευταίων συναλλαγών ενός νόμιμου χρήστη. Οι συναλλαγές κάθε χρήστη βρίσκονται αποθηκευμένες σε μια βάση δεδομένων. Η δυνατότητα που προσφέρει η υπηρεσία των τελευταίων συναλλαγών είναι η εισαγωγή του χρονικού ορίου, όπου ο χρήστης επιθυμεί να εμφανίσει όλες τις συναλλαγές που πραγματοποιήθηκαν σε αυτό. Με μια επίθεση έκχυσης SQL, σε πεδία που υπάρχει η δυνατότητα εισαγωγής σε βάση δεδομένων εάν δεν φιλτράρονται κατάλληλα, ένας κακόβουλος χρήστης μπορεί να αντλήσει κρίσιμες πληροφορίες για το σύστημα. Στο παράδειγμα έχουμε την εισαγωγή των χαρακτήρων **1/1/2010 union select userid,null,username+','+password,null from users--** στη φόρμα όπου το 1/1/2010 εισάγεται επειδή είναι επιτρεπτή τιμή,

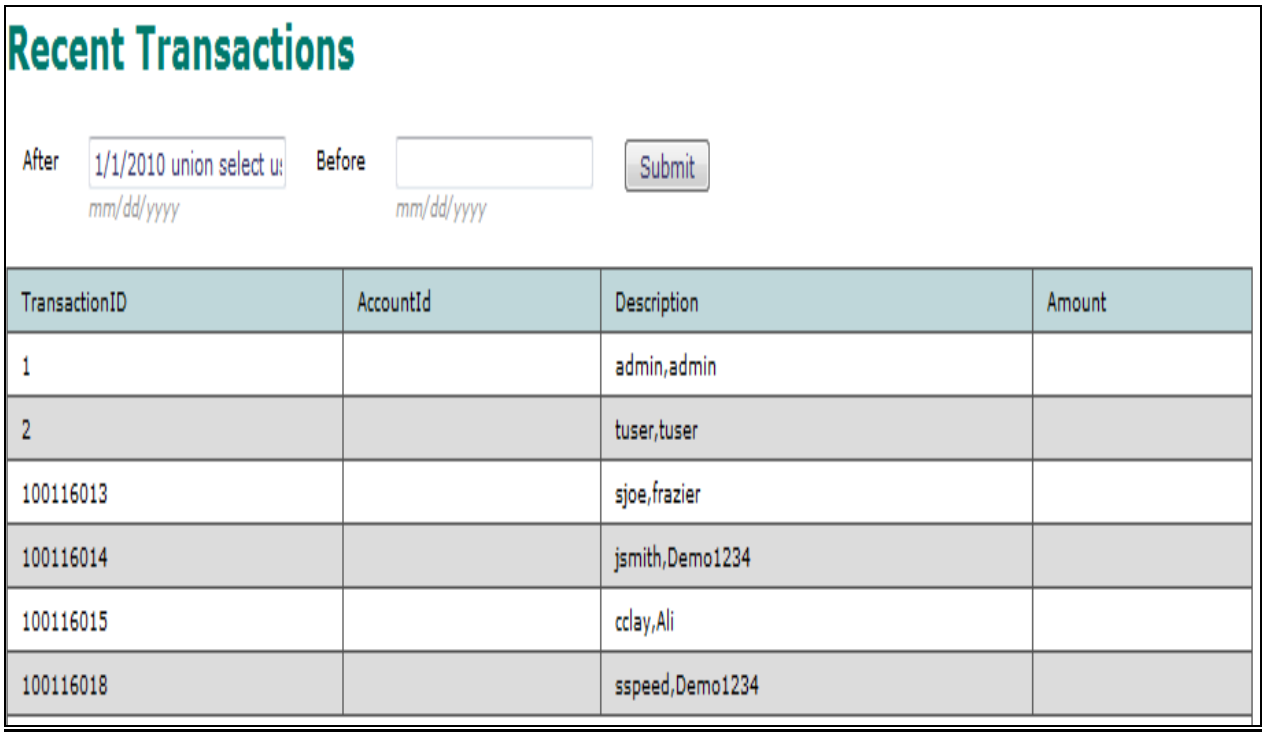
ακολουθούμενο από ένα SQL ερώτημα. Με αυτόν τον τρόπο πετυχαίνουμε την εμφάνιση όλων των χρηστών της ιστοσελίδας με το όνομα χρήστη και τον κωδικό πρόσβασης τους. Στην εικόνα 4.32 που εμφανίζεται βάση του ερωτήματος SQL που εισήχθη έχουμε στην πρώτη στήλη το userid, η δεύτερη στήλη είναι κενή, η τρίτη περιέχει το όνομα χρήστη και τον κωδικό πρόσβασης και η τέταρτη είναι επίσης κενή.



Εικόνα 4.30 Πίνακας συναλλαγών νόμιμου χρήστη



Εικόνα 4.31 Εισαγωγή ερωτήματος SQL



Εικόνα 4.32 Επιστροφή αποτελεσμάτων επίθεσης SQL

Όπως και στις προηγούμενες ευπάθειες που σχετίζονταν με εκχύσεις SQL ο τρόπος αντιμετώπισης παραμένει το φιλτράρισμα ειδικών χαρακτήρων πριν την εισαγωγή από τα πεδία στη βάση δεδομένων.

4.1.2.10 Μετατροπή παραμέτρων αρχείου των Windows (Windows File Parameter Alteration)

Ακόμα μια κρίσιμη ευπάθεια που εντοπίζεται στην ιστοσελίδα είναι αυτή της μετατροπής παραμέτρων των αρχείων των Windows. Η ευπάθεια αυτή εντοπίζεται όταν από μια απάντηση του συστήματος υπάρχει η δυνατότητα να αποκτηθεί κάποιο αρχείο του συστήματος όπως το boot.ini. Με χρήση κώδικα CGI[8] (Common Gateway Interface) ένας κακόβουλος χρήστης αν δεν υπάρχει έλεγχος από την εφαρμογή μπορεί να αλλάξει τις παραμέτρους του κώδικα και να ζητήσει οποιοδήποτε άλλο αρχείο υπάρχει στον εξυπηρετητή. Έστω ότι έχουμε το παρακάτω κομμάτι κώδικα CGI σε μια φόρμα της διαδικτυακής εφαρμογής:

[8] http://en.wikipedia.org/wiki/Common_Gateway_Interface

```
<FORM METHOD=POST ACTION="/cgi-bin/vulnerable_script.cgi">
...
<INPUT TYPE=HIDDEN NAME="template"VALUE="/dir1/dir2/template.txt">
...
</FORM>
```

Εάν δεν υπάρχει έλεγχος το παραπάνω μπορεί να τροποποιηθεί σε:

```
<FORM METHOD=POST ACTION=http://target/cgi-bin/vulnerable_script.cgi>
...
<INPUT TYPE=HIDDEN NAME="template" VALUE="../../../../../boot.ini">
...
</FORM>
```

όπου με πειραματισμούς στην χρήση των τελεστών **..** που δείχνουν τον ανώτερο κατάλογο (directory) από το σημείο στο οποίο βρίσκεται το αρχείο, ο κακόβουλος χρήστης έχει την δυνατότητα να του επιστραφεί από την φόρμα οποιοδήποτε αρχείο επιθυμεί.

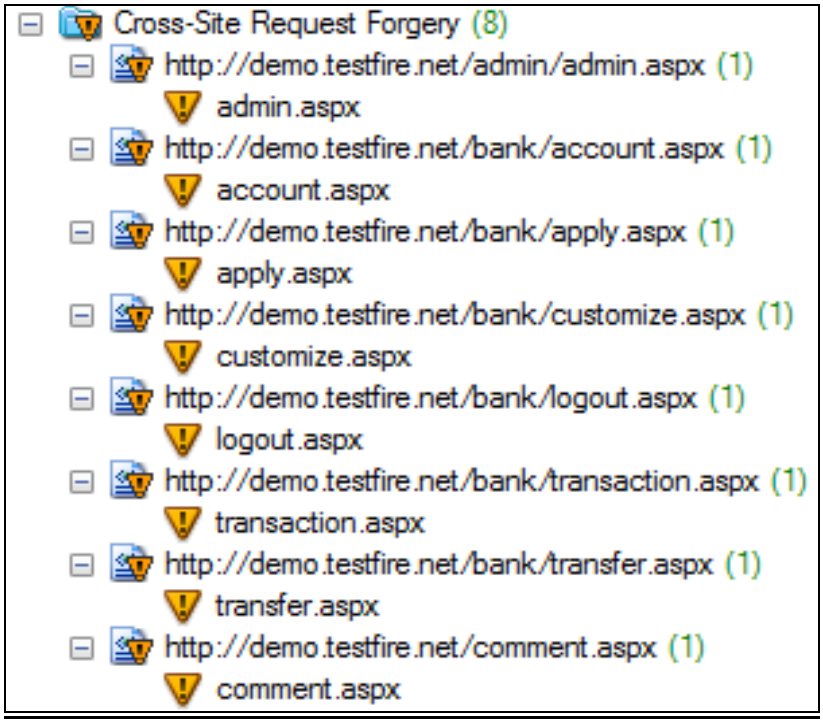
Οι τρόποι για να αντιμετωπιστεί αυτή η ευπάθεια είναι να σιγουρευτούμε ότι το αρχείο στην φόρμα βρίσκεται στην εικονική (virtual) διαδρομή του εξυπηρετητή ιστού, να μην επιτρέπεται η εισαγωγή ειδικών χαρακτήρων και να γίνεται έλεγχος ώστε να επιτρέπεται μόνο σε συγκεκριμένες επεκτάσεις να χρησιμοποιηθούν.

4.1.2.11 Έκχυση XPath (XPath Injection)

Ομοίως με τις ευπάθειες έκχυσης SQL που αφορούν τις SQL βάσεις δεδομένων, η ευπάθεια έκχυσης XPath έχει να κάνει με κενά ασφάλειας στα XML έγγραφα. Με τις κατάλληλες εντολές ένας κακόβουλος χρήστης μπορεί να αποκτήσει πρόσβαση σε όλες τις πληροφορίες που περιέχονται σε ένα XML έγγραφο ακριβώς με τον ίδιο τρόπο που είδαμε στις τυφλές εκχύσεις SQL. Ο τρόπος αντιμετώπισης της ευπάθειας είναι το φιλτράρισμα των ειδικών χαρακτήρων που εισάγει ο χρήστης ώστε να μην μπορούν να αποτελέσουν ερωτήματα.

4.1.2.12 Πλαστογραφία αίτησης Cross-Site (Cross-Site Request Forgery)

Η ευπάθεια της πλαστογραφίας αίτησης Cross-Site στη γενική της μορφή περιγράφθηκε στο πρώτο κεφάλαιο. Σε αυτή την ευπάθεια το θύμα οδηγείται από τον κακόβουλο χρήστη στην αποστολή ενός αιτήματος στη διαδικτυακή εφαρμογή. Για το αίτημα αυτό χρησιμοποιούνται τα cookies του φυλλομετρητή του θύματος, οπότε ο κακόβουλος χρήστης έχει πρόσβαση μέσω του νόμιμου χρήστη για οποιαδήποτε ενέργεια στο Web πληροφοριακό σύστημα. Ευπάθεια υπάρχει στο σύστημα όταν δεν ελέγχεται η πηγή του αιτήματος ή το σύστημα δεν επαληθεύει πως ο χρήστης θέλει να εκτελέσει τη συγκεκριμένη ενέργεια. Οι τρόποι που το θύμα οδηγείται στην αποστολή του αιτήματος είναι η αποστολή του κακόβουλου συνδέσμου(malicious link) που περιέχει το αίτημα μέσω ηλεκτρονικού ταχυδρομείου, η ανάρτηση του κακόβουλου συνδέσμου σε μια ιστοσελίδα του επιτιθέμενου με την παραπλανητική μορφή μιας ενδιαφέρουσας εικόνας ή ενός κανονικού συνδέσμου και η ανάρτηση του κακόβουλου συνδέσμου σε κάποιο φόρουμ. Στην εικόνα 4.33 φαίνονται τα τμήματα της ιστοσελίδας Altoro Mutual που πάσχουν από την ευπάθεια της πλαστογραφίας αίτησης Cross-Site.



Εικόνα 4.33 Τμήματα ευπάθειας πλαστογραφίας αίτησης Cross-Site

Ένα απλό παράδειγμα για την κατανόηση της ευπάθειας είναι το εξής: Υποθέτουμε πως μια ιστοσελίδα ηλεκτρονικών τραπεζικών συναλλαγών χρησιμοποιεί cookies για να ελέγχει πως ένας χρήστης έχει πιστοποιηθεί στο σύστημα. Το αίτημα του χρήστη για μεταφορά χρημάτων έχει την μορφή http://mybank/transfer?toAcct=123&sum=10, έπειτα ο κακόβουλος χρήστης χρησιμοποιώντας έναν από τους τρόπους που περιγράψαμε παραπάνω οδηγεί το θύμα στον σύνδεσμο <img src='http://mybank/transfer?toAcct=attack&sum=100> όπου ο φυλλομετρητής του θύματος θα αποστείλει το αίτημα με το cookie του θύματος για πιστοποίηση και τα χρήματα θα μεταφερθούν στο λογαριασμό του κακόβουλου χρήστη.

Ο τρόπος προστασίας ενός Web πληροφοριακού συστήματος από αυτή την ευπάθεια, είναι να χρησιμοποιηθεί μια μέθοδος ελέγχου με ένα μοναδικό αναγνωριστικό που είναι δύσκολο να μαντέψει ο επιτιθέμενος. Ένα τέτοιο αναγνωριστικό είναι το αναγνωριστικό συνόδου (session id). Ο λόγος που το αναγνωριστικό συνόδου είναι μοναδικό και αποτρέπει την εκμετάλλευση της ευπάθειας από κακόβουλους χρήστες είναι η πολιτική της κοινής προέλευσης (Same Origin Policy) των εξυπηρετητών, όπου αίτημα και αναγνωριστικό πρέπει να έχουν κοινή προέλευση. Έτσι ένας κακόβουλος χρήστης δεν μπορεί να πλαστογραφήσει ένα αίτημα που θα φανεί γνήσιο στον εξυπηρετητή για να απαντήσει εφόσον χρειάζεται και το αναγνωριστικό συνόδου του χρήστη.

4.1.2.13 Διαχωρισμός απαντήσεων HTTP (HTTP Response Splitting)

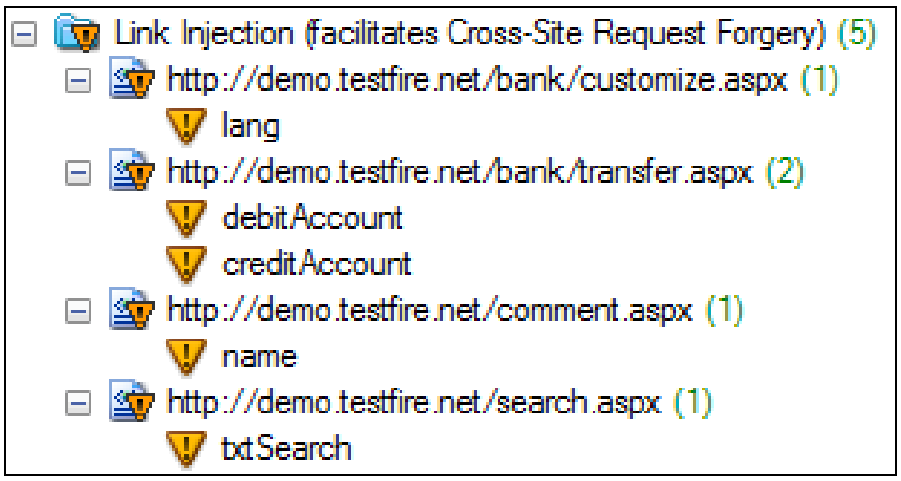
Η ευπάθεια διαχωρισμού απαντήσεων HTTP αφορά την εισαγωγή εμφωλευμένων δεδομένων χρήστη, στις HTTP απαντήσεις. Εάν στα εμφωλευμένα δεδομένα ένας κακόβουλος χρήστης εισάγει μια συγκεκριμένη ακολουθία χαρακτήρων, μπορεί να πετύχει δύο απαντήσεις HTTP αντί για μία από το ρεύμα εξόδου (output stream) του εξυπηρετητή ιστού. Στην πρώτη ο επιτιθέμενος έχει μερικό έλεγχο, όμως στην δεύτερη έχει πλήρη έλεγχο από την γραμμή κατάστασης (status line) HTTP μέχρι το τελευταίο byte του σώματος της απάντησης HTTP. Συνήθως ο επιτιθέμενος τερματίζει την πρώτη απάντηση και δημιουργεί την

δεύτερη απάντηση ώστε να εμφανίσει οτιδήποτε επιθυμεί. Αυτό έχει ως αποτέλεσμα Cross-Site Scripting επιθέσεις, παραμορφώσεις ιστοσελίδων (website defacement) και λήψη ολοκληρωτικού ελέγχου (hijacking) της διαδικτυακής εφαρμογής από την μεριά του κακόβουλου χρήστη.

Ο τρόπος αντιμετώπισης της ευπάθειας είναι το φιλτράρισμα των ειδικών χαρακτήρων που μπορεί ένας χρήστης να εισάγει. Με αυτόν τον τρόπο οι κακόβουλοι χρήστες δεν θα έχουν την δυνατότητα να χωρίσουν τις HTTP απαντήσεις σε περισσότερες της μιας.

4.1.2.14 Έκχυση συνδέσμων (Link Injection)

Όταν σε μια διαδικτυακή εφαρμογή ο χρήστης μπορεί να εισάγει έναν οποιοδήποτε σύνδεσμο προς μια ιστοσελίδα, τότε η διαδικτυακή εφαρμογή πάσχει από την ευπάθεια της έκχυσης συνδέσμων. Μπορεί από πρώτη ματιά να φαίνεται αθώα η δυνατότητα να εισάγουμε συνδέσμους στο περιεχόμενο μιας διαδικτυακής εφαρμογής, όμως οι κακόβουλοι χρήστες εκμεταλλεύονται αυτήν την ευπάθεια για να εξαπολύσουν επιθέσεις Cross-Site Scripting και πλαστογραφία αίτησης Cross-Site. Στην εικόνα 4.34 φαίνονται τα τμήματα της ευπάθειας έκχυσης συνδέσμων.

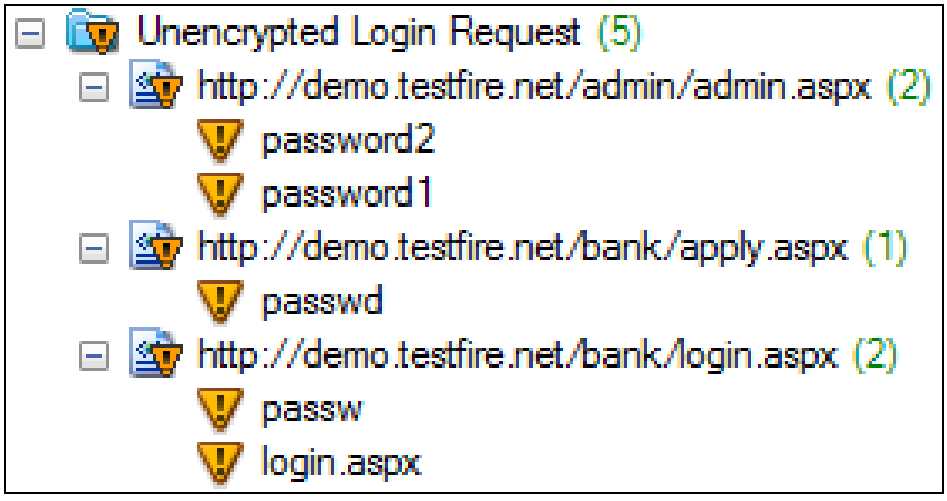


Εικόνα 4.34 Τμήματα ευπάθειας έκχυσης συνδέσμων

Ο τρόπος αντιμετώπισης της ευπάθειας της έκχυσης συνδέσμων είναι να μην επιτρέπεται στον χρήστη να εισάγει χαρακτήρες που μπορούν να χρησιμοποιηθούν για την δημιουργία ενός συνδέσμου στα πεδία εισαγωγής δεδομένων.

4.1.2.15 Αίτηση πιστοποίησης χωρίς κρυπτογράφηση (Unencrypted Login Request)

Τα δεδομένα που μεταφέρονται από και προς την διαδικτυακή εφαρμογή, όπως είπαμε και στο πρώτο κεφάλαιο θα πρέπει να κρυπτογραφούνται. Χωρίς κρυπτογράφηση κακόβουλοι χρήστες μπορούν να συλλέξουν πληροφορίες για την εφαρμογή. Η ευπάθεια αίτηση πιστοποίησης χωρίς κρυπτογράφηση αφορά την απουσία κρυπτογράφησης σε σημαντικά δεδομένα, για την πιστοποίηση των χρηστών στη διαδικτυακή εφαρμογή. Τέτοιου είδους δεδομένα είναι το όνομα χρήστη, ο κωδικός πρόσβασης και δεδομένα που εξυπηρετούν τις ηλεκτρονικές συναλλαγές όπως κωδικοί καρτών. Στην εικόνα 4.35 βλέπουμε τα τμήματα της ιστοσελίδας που πάσχουν από αυτή την ευπάθεια. Η ευπάθεια εντοπίστηκε από το Rational Appscan επειδή παράμετροι όπως ο κωδικός πρόσβασης βρέθηκαν πως δεν στέλνονται μέσω SSL.



Εικόνα 4.35 Τμήματα ευπάθειας αίτησης πιστοποίησης χωρίς κρυπτογράφηση

Ο καλύτερος τρόπος αντιμετώπισης αυτής της ευπάθειας είναι να σιγουρευτούμε πως όλες οι διαδικασίες πιστοποίησης γίνονται μέσω SSL και επίσης ευαίσθητα δεδομένα όπως, ο κωδικός πρόσβασης, το όνομα χρήστη, οι διευθύνσεις ηλεκτρονικού ταχυδρομείου, ο ταχυδρομικός κωδικός, οι κωδικοί καρτών και άλλα μεταφέρονται επίσης μέσω SSL.

4.1.2.16 Ύπαρξη δοκιμαστικών τμημάτων της διαδικτυακής εφαρμογής (Application Test Script Found)

Η συγκεκριμένη ευπάθεια εντοπίζεται όταν υπάρχουν δοκιμαστικά ή ημιτελή τμήματα της διαδικτυακής εφαρμογής στο περιβάλλον λειτουργίας. Συχνά οι προγραμματιστές ξεχνούν να αφαιρέσουν δοκιμαστικά τμήματα μιας εφαρμογής με αποτέλεσμα ένας χρήστης να μπορέσει να έχει πρόσβαση σε αυτά. Τέτοια τμήματα είναι πιθανό να περιέχουν ευαίσθητα δεδομένα, οπότε πρέπει να γίνεται σχολαστικός έλεγχος στην διαδικτυακή εφαρμογή για την απομάκρυνση τους από το περιβάλλον λειτουργίας.

4.1.2.17 Άμεση πρόσβαση σε σελίδες διαχείρισης (Direct Access To Administration Pages)

Σε μια διαδικτυακή εφαρμογή οι χρήστες μετακινούνται μέσω των συνδέσμων. Υπάρχει όμως η περίπτωση, περιοχές που έχουν δικαιώματα διαχείρισης και δεν ενώνονται με την υπόλοιπη εφαρμογή μέσω συνδέσμων, να προσπελαθούν από χρήστες που πληκτρολογούν διευθύνσεις στον ενιαίο εντοπιστή πόρων (URL). Σε αυτήν την περίπτωση έχουμε την ευπάθεια της άμεσης πρόσβασης σε σελίδες διαχείρισης.

Ο τρόπος αντιμετώπισης αυτής της ευπάθειας είναι να μην επιτρέπεται η πρόσβαση σε τμήματα διαχείρισης της διαδικτυακής εφαρμογής χωρίς την απαραίτητη πιστοποίηση.

4.1.2.18 Κοινοποίηση ευαίσθητων πληροφοριών μέσω σχόλιων HTML (HTML Comments Sensitive Data Disclosure)

Η ευπάθεια αυτή συναντάται όταν μέσα στο σώμα HTML μιας διαδικτυακής εφαρμογής υπάρχουν σχόλια τα οποία κοινοποιούν ευαίσθητα δεδομένα. Συνήθως τέτοια σχόλια είναι ξεχασμένα από τους προγραμματιστές της εφαρμογής για επιπλέον βοήθεια στην αποσφαλμάτωση. Επιπλέον έλεγχοι πρέπει να γίνονται ώστε τέτοια σχόλια να απομακρύνονται από τα μέρη της διαδικτυακής εφαρμογής.

4.1.2.19 Λοιπές ευπάθειες ιστοσελίδας Altoro Mutual

Το τελευταίο μέρος των ευπαθειών που εντοπίζονται από το Rational Appscan, αποτελείται κυρίως από το κομμάτι των παρατηρήσεων και περιγράφεται σε αυτό το υποκεφάλαιο. Οι παρατηρήσεις είναι ευπάθειες που όμως αποτελούν μικρό κίνδυνο για την διαδικτυακή εφαρμογή. Πάραυτα πρέπει να αντιμετωπίζονται από τους προγραμματιστές έτσι ώστε η εφαρμογή να είναι όσο το δυνατόν πιο ασφαλής. Παραδείγματα παρατηρήσεων στην ιστοσελίδα Altoro Mutual αποτελεί η παρότρυνση να μην εμφανίζεται η διεύθυνση ηλεκτρονικού ταχυδρομείου στη ιστοσελίδα, σε σημείο που φαίνεται από όλους τους χρήστες ώστε να ελαττωθεί η πιθανότητα χρήσης της ως προορισμό ανεπιθύμητης αλληλογραφίας (spam). Επιπλέον σε καμία περίπτωση δεν πρέπει να μένει εκτεθειμένη η απόλυτη διαδρομή κάποιων αρχείων της ιστοσελίδας, επειδή μπορεί ένας κακόβουλος χρήστης να εισάγει άλλες διαδρομές που προκύπτουν από αυτήν και να αποκτήσει πρόσβαση σε ευαίσθητα δεδομένα. Πολύ σημαντική είναι η παρουσία ελέγχων στα πεδία εισαγωγής, τόσο στο να προκαθορίζεται ο αριθμός των επιτρεπόμενων χαρακτήρων όσο και στο να εισάγονται δεδομένα με την μορφή που αναμένεται από την διαδικτυακή εφαρμογή.

4.3 Σύνοψη

Με το σενάριο χρήσης του προγράμματος Rational Appscan στην ιστοσελίδα Altoro Mutual κατανοήσαμε το μέγεθος των ευπαθειών που έχει μια διαδικτυακή εφαρμογή στην εποχή μας. Αν και το παράδειγμα της συγκεκριμένης ιστοσελίδας, που έχει κατασκευαστεί για εκπαιδευτικούς σκοπούς, περιείχε ένα μέρος μονάχα του συνόλου των ευπαθειών που επηρεάζουν τα Web πληροφοριακά συστήματα, περιγράφθηκαν σε γενικές γραμμές οι κυριότερες από τις ευπάθειες. Τελειώνοντας την ανάγνωση αυτού του κεφαλαίου ο αναγνώστης κατανοεί την δράση των διαγνωστικών προγραμμάτων, όπως το Rational Appscan, και αντιλαμβάνεται την σημαντικότητα που έχει η αντιμετώπιση των ευπαθειών που εντοπίζουν.

ΚΕΦΑΛΑΙΟ 5: Συμπεράσματα

Στόχος της παρούσας πτυχιακής εργασίας είναι ο προσδιορισμών των ευπαθειών των Web πληροφοριακών συστημάτων, η κατασκευή ενός μεθοδολογικού πλαισίου αξιολόγησης τους και η δημιουργία σεναρίου χρήσης του διαγνωστικού προγράμματος IBM Rational Appscan. Με αυτό τον τρόπο αναλύεται πλήρως το μεγάλο κεφάλαιο της ασφάλειας των Web πληροφοριακών συστημάτων με ειδικότερη έμφαση στις ευπάθειες που παρουσιάζονται στις διαδικτυακές εφαρμογές τους.

Στο πρώτο μέρος της εργασίας γίνεται αναφορά στις ευπάθειες που έχουν χαρακτηριστεί συχνότερες σε βαθμό εμφάνισης από το πρόγραμμα OWASP. Από αυτό συμπεραίνουμε πως το κομμάτι των ευπαθειών μελετάτε εδώ και πολλά χρόνια και καθώς αναλύονται οι ευπάθειες παρατηρούμε τους κινδύνους που απορρέουν από την ύπαρξη τους. Έπειτα γίνεται κατασκευή του μεθοδολογικού πλαισίου αξιολόγησης των ευπαθειών για την κατηγοριοποίηση τους. Στην κατηγοριοποίηση διαχωρίζουμε τις ευπάθειες ανάλογα με τα τμήματα του Web πληροφοριακού συστήματος που επηρεάζουν. Αυτή η κατηγοριοποίηση βοηθάει στον προσδιορισμό των ευπαθειών. Το κάθε επίπεδο του μεθοδολογικού πλαισίου περιέχει ευπάθειες που αντιμετωπίζονται με παρόμοιο τρόπο. Καθώς κατανοούμε καλύτερα την λειτουργία των τμημάτων ενός Web πληροφοριακού συστήματος προοδεύουμε και στον τομέα της αντιμετώπισης των ευπαθειών τους.

Στο δεύτερο μέρος της εργασίας περιγράφεται το διαγνωστικό πρόγραμμα IBM Rational Appscan. Το εργαλείο αυτό εξετάζει διαδικτυακές εφαρμογές για ευπάθειες και προτείνει τρόπους επίλυσης τους. Σενάριο χρήσης του πραγματοποιείτε για λογαριασμό της ιστοσελίδας Altoro Mutual, στην οποία εντοπίζονται οι ευπάθειες που περιέχει. Μέσα από τα παραδείγματα της ανάλυσης των ευπαθειών ο αναγνώστης αντιλαμβάνεται τους κινδύνους που έχουν. Με χρήση βασικών γνώσεων και χωρίς πολύπλοκα εργαλεία στην διάθεση τους, κακόβουλοι χρήστες μπορούν να προξενήσουν τεράστια ζημιά στις διαδικτυακές εφαρμογές ή να αποκτήσουν πρόσβαση σε προσωπικά δεδομένα και άλλες μεγάλης σημασίας πληροφορίες. Η χρήση προγραμμάτων όπως το IBM Rational

Appscan κρίνεται απαραίτητη για την ασφάλεια του τομέα των διαδικτυακών εφαρμογών των Web πληροφοριακών συστημάτων.

Τέλος με την ολοκλήρωση της πτυχιακής εργασίας έχει καλυφθεί το μεγαλύτερο μέρος των ευπαθειών που πλήττουν τις διαδικτυακές εφαρμογές. Ο τομέας τους συνεχώς εξελίσσεται και αυτό υποδηλώνει η ανά τριετία ανανέωση της λίστας με τις συχνότερα εμφανιζόμενες ευπάθειες του OWASP. Επειδή όμως οι διαδικτυακές εφαρμογές αποτελούν μονάχα ένα επίπεδο του μεθοδολογικού πλαισίου αξιολόγησης ευπαθειών των Web πληροφοριακών συστημάτων, υπάρχει χώρος για επιπλέον μελέτη στα άλλα επίπεδα. Μελλοντική επέκταση του αντικειμένου της παρούσας πτυχιακής εργασίας θα μπορούσε να θεωρηθεί η ανάπτυξη ενός εργαλείου, στο οποίο θα γίνεται έλεγχος ευπαθειών και των άλλων επιπέδων του μεθοδολογικού πλαισίου(Επίπεδο δικτύου, επίπεδο host, επίπεδο βάσεων δεδομένων, επίπεδο κώδικα). Η εξέταση και ανάλυση των ευπαθειών των άλλων επιπέδων σε συνδυασμό με όσα αναπτύχθηκαν στην παρούσα εργασία, θα δημιουργούσε έναν οδηγό για την ασφάλεια των Web πληροφοριακών συστημάτων. Μόνο με την αντιμετώπιση των ευπαθειών κάθε επιπέδου, μπορεί ένας μηχανικός πληροφορικής να σιγουρευτεί για τη συνολική ασφάλεια ενός Web πληροφοριακού συστήματος.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Έντυπη

Άρθρα/Δημοσιεύσεις:

1) Amit Klein (2008), IBM Rational AppScan: Cross-site scripting explained, March 2008

2) Chris Anley (2002) , Advance SQL Injection in SQL Server Applications.

3) Danny Allan (2008), IBM Web Application Security Paper. January 2008, pp 2-3.

4) Edward W. Felten and William Zeller. (2008) "Cross-Site Request Forgeries: Exploitation and Prevention". October 2008.

5) Larry Suto (2010), Analyzing the Accuracy and Time Costs of Web Application Security Scanners, February (2010), pp 3-4.

6) Shawn Miller (2007), Discovering the Value of Verifying Web Application Security Using IBM Rational AppScan, pp 3-61.

Βιβλία:

1) Βεσκούκης Βασίλειος (2000), <Τεχνολογία λογισμικού> pp 87-88.

2) Γκρίτζαλη Σ, Κάτσικα Σ, Γκρίτζαλη Γ. (2003) Ασφάλεια Δικτύων Υπολογιστών.

3) Τασόπουλος Αναστάσιος (2005), «Πληροφοριακά συστήματα. Οργάνωση, μεθοδολογία, εφαρμογές», pp 5-6.

4) Eckerson, Wayne W. (1995), "Three Tier Client/Server Architecture: Achieving Scalability, Performance, and Efficiency in Client Server Applications." Open Information Systems, Vol 10, No 1, pp 3-20.

5) Fogie, Grossman, Hansen and Rager (2007), "*Cross Site Scripting Attacks: XSS Exploits and Defense*" .

6) M. Howard and D. LeBlanc. (2002) "Writing Secure Code". Chapter 4, "Authentication" pp 109.

7) Michael Howard, David LeBlanc and John Viega (2010), "24 Deadly Sins of Software Security". pp 185,194,253.

8) William Stalling (2005) Cryptography and Network Security Principles and Practices, Fourth Edition.

Ηλεκτρονική

1) IBM (http://www.ibm.com)
2) OWASP(2007) "Top 10 2007" (www.owasp.org/index.php/Top_10_2007)
3) US-CERT (http://www.us-cert.gov/)
4) Wikipedia (http://en.wikipedia.org)