Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών
«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	**Ανασχεδιασμός Ασφάλειας Διαδικτυακών Εφαρμογών με τη χρήση Ανάλυσης Ευπαθειών** **- Μελέτη περίπτωσης σε περιβάλλον .NET** **Web Application Security Reengineering based on Vulnerability Analysis** **– A Case study using the .NET Framework**
Ονοματεπώνυμο Φοιτητή	**Γεώργιος Οικονόμου**
Πατρώνυμο	**Βασίλειος**
Αριθμός Μητρώου	**ΜΠΠΛ 12045**
Επιβλέπων	**Παναγιώτης Κοτζανικολάου, Λέκτορας**

Ημερομηνία Παράδοσης **Δεκέμβριος 2014**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)	(υπογραφή)	(υπογραφή)
Παναγιώτης Κοτζανικολάου Λέκτορας	Δημήτριος Βέργαδος Επίκουρος Καθηγητής	Μιχαήλ Ψαράκης Επίκουρος Καθηγητής

Επιτελική Σύνοψη

Οι διαδικτυακές εφαρμογές επεξεργάζονται συνήθως ευαίσθητα και προσωπικά δεδομένα των χρηστών τους. Για αυτό το λόγο, κατά την ανάπτυξη αυτών των εφαρμογών θα πρέπει να γίνεται προληπτική ανάλυση της ασφάλειας, με σκοπό τον έγκαιρο εντοπισμό και την επιδιόρθωση των ευπαθειών ασφάλειας. Οι υπεύθυνοι ανάπτυξης των εφαρμογών θα πρέπει να θεωρούν την ασφάλεια των εφαρμογών χαρακτηριστικό ίσης προτεραιότητας με την λειτουργικότητά τους. Η ασφάλεια των εφαρμογών μόνο εύκολη υπόθεση δεν θα μπορούσε να χαρακτηριστεί σήμερα όπου υπάρχουν πάρα πολλά προβλήματα, αδυναμίες και απειλές ασφάλειας και συνεχώς εμφανίζονται νέες που παρουσιάζουν μεγαλύτερη πολυπλοκότητα και μειώνουν τον ασφαλή χρόνο ζωής των εφαρμογών.

Οι σημαντικότερες συνέπειες που μπορούν να προκληθούν από πιθανή παραβίαση της ασφάλειας μιας διαδικτυακής εφαρμογής περιλαμβάνουν την υποκλοπή προσωπικών δεδομένων ή την καταστροφή τους, την μη διαθεσιμότητα της εφαρμογής για κάποιο χρονικό διάστημα μέχρι και την μερική καταστροφή της. Η ασφάλεια απασχολεί πια σε μεγάλο βαθμό τους προγραμματιστές κατά την διάρκεια ανάπτυξης της εφαρμογής και στην συνέχεια τους υπευθύνους ασφαλείας όταν αναφερόμαστε σε μεγάλους οργανισμούς ή εταιρείες. Αυτό όμως προϋποθέτει τη γνώση τους σε θέματα ασφάλειας των διαδικτυακών εφαρμογών και την πιστή τήρηση αυτών.

Τα εργαλεία εντοπισμού ευπαθειών ασφάλειας (vulnerability scanners) χρησιμοποιούνται ως ένα εργαλείο για την προληπτική ανάλυση ευπαθειών, και την ενίσχυση της ασφάλειάς τους, στο στάδιο της ανάπτυξης των εφαρμογών. Η παρούσα διατριβή εστιάζει στον ανασχεδιασμό ασφάλειας (security re-engineering) των διαδικτυακών εφαρμογών, με βάση μία προληπτική ανάλυση ευπαθειών. Αρχικά παρουσιάζονται γνωστές αδυναμίες και ευπάθειες των διαδικτυακών εφαρμογών, Στη συνέχεια παρουσιάζεται μία τυπική διαδικτυακή εφαρμογή, η οποία έχει αναπτυχθεί σε περιβάλλον .NET και η οποία θα χρησιμοποιηθεί ως περιβάλλον δοκιμής για μία μελέτη περίπτωσης. Εφαρμόζοντας το γνωστό εργαλείο εντοπισμού ευπαθειών Nessus, γίνεται μία ανάλυση ευπαθειών της εφαρμογής. Τέλος, με βάση τα αποτελέσματα της ανάλυσης, εφαρμόζονται διορθωτικά μέτρα ασφάλειας και γίνεται ανασχεδιασμός της υλοποίησης της δοκιμαστικής εφαρμογής, με σκοπό την μείωση της έκθεσής της σε διαδικτυακές επιθέσεις ασφάλειας.

Abstract

Web applications process sensitive personal and private data of their users. For this reasons, during their development, the web applications should be proactively analyzed for known security vulnerabilities. Developers should have at equal priority the security and the functionality of a web application.. Nowadays, the security of the applications could not be characterized as an easy task as there are many problems, vulnerabilities and threats and new ones always appear which are much more complicated and reduce the duration of the applications' safe functionality.

The most important impacts which can be caused from a possible hacking of a web application include wiretapping of personal data or their destruction, non-availability of the application for a certain period of time as well as the destruction of the application. Security is a very important issue for developers during the development of an application and is also important for security managers when it comes to big organizations and companies. However, this assumes their knowledge in security policies of web applications and their full compliance.

Vulnerability scanners can be applied as a preventive security control, in order to enhance the security of a web application during its development. Then, security measures should be applied, based on the results of a vulnerability analysis. This thesis focuses on the security re-engineering of web applications, based on the proactive vulnerability analysis. We first present a study of known vulnerabilities for web environments. Then we present a web application, developed as a case study for this analysis, using the .NET framework. By using the well-known Nessus vulnerability scanner, we analyze the vulnerabilities of the web application. Finally, based on the results of the vulnerability analysis, we apply corrective security controls and re-engineer the implementation of the test application to reduce its exposure to known security threats.

Πίνακας περιεχομένων

ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

ΠΙΝΑΚΑΣ ΠΙΝΑΚΩΝ

Κεφάλαιο 1°

1 Εισαγωγή στην Ασφάλεια Διαδικτυακών Εφαρμογών

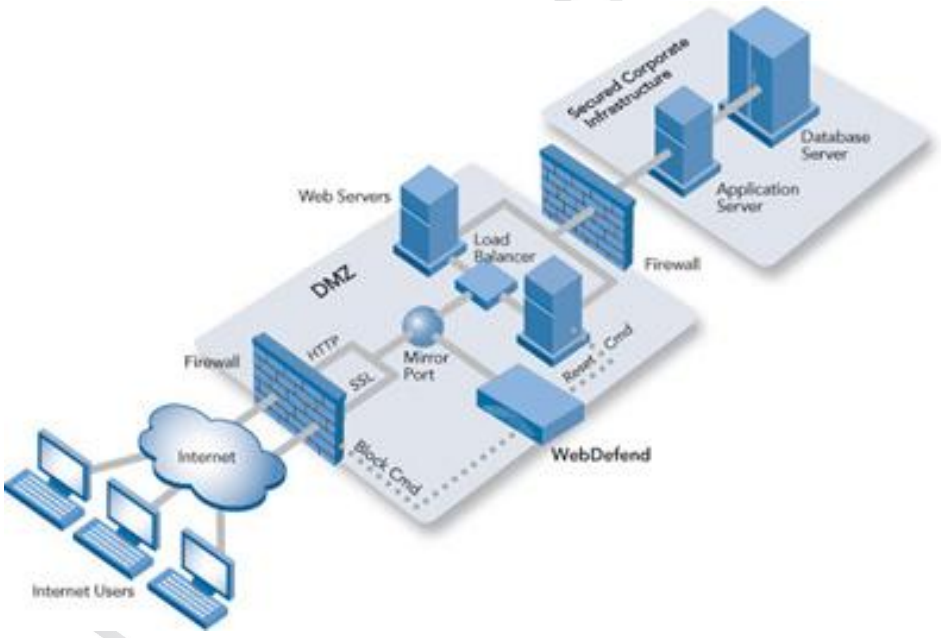
1.1 Εισαγωγή

Το επιστημονικό πεδίο με το οποίο θα ασχοληθεί η εργασία είναι αυτό της Ασφάλειας Διαδικτυακών Εφαρμογών. Είναι ένας κλάδος της Ασφάλειας Πληροφοριών ο οποίος ασχολείται με την ασφάλεια ιστοσελίδων, εφαρμογών Διαδικτύου και υπηρεσιών Διαδικτύου. Βασίζεται δηλαδή στις αρχές που διέπουν την ασφάλεια εφαρμογών οι οποίες όμως προσανατολίζονται στο Διαδίκτυο και στα συστήματα που λειτουργούν σε αυτό.

Σε αυτό το κεφάλαιο θα γίνει μια παρουσίαση του σκοπού και των στόχων της εργασίας. Θα αναλυθούν προκειμένου να γίνουν κατανοητές οι βασικές έννοιες που θα χρησιμοποιηθούν στην παρούσα διπλωματική εργασία και αφορούν την ασφάλεια διαδικτυακών εφαρμογών.

1.2 Αρχιτεκτονική Ασφάλειας 3 Επιπέδων

Ασφάλεια πληροφοριών γενικά είναι η διατήρηση της εμπιστευτικότητας, της ακεραιότητας και της διαθεσιμότητας των πληροφοριών, όπως και της διαθεσιμότητας του συστήματος που χειρίζεται τις πληροφορίες. [1]



Εικόνα 1 . Αρχιτεκτονική ασφάλειας διαδικτυακών εφαρμογών

Για να διασφαλιστεί η ασφάλεια των πληροφοριών σε μια διαδικτυακή εφαρμογή όπως φαίνεται στην εικόνα 1 θα πρέπει να επιτευχθούν οι καλύτερες συνθήκες ασφάλειας στα παρακάτω τρία επίπεδα. [2]

1. Επίπεδο Εφαρμογής
2. Επίπεδο Host
3. Επίπεδο Δικτύου

1.2.1 Επίπεδο Εφαρμογής

Ο βαθμός ασφαλείας που μπορεί να επιτευχθεί στις εφαρμογές εξαρτάται από τον σχεδιασμό και την υλοποίηση τους. Τα προβλήματα ασφαλείας και οι κίνδυνοι που τις απειλούν είναι γνωστά και οι συνθήκες ασφαλείας που πρέπει να εξασφαλιστούν από τους προγραμματιστές αφορούν θέματα που αναφέρονται ενδεικτικά εδώ όπως η αυθεντικοποίηση, η εξουσιοδότηση, η κρυπτογραφία και η διαχείριση ευαίσθητων δεδομένων. Θα πρέπει να ληφθούν όλα υπόψη για την υλοποίηση μιας εφαρμογής που θα μπορεί να χαρακτηριστεί ασφαλής.

1.2.2 Επίπεδο Host

Στο επίπεδο του host η ασφάλεια αφορά τους α) application server β) web server και γ) database server που θα χρησιμοποιηθούν για την εφαρμογή. Θα πρέπει να γίνουν οι κατάλληλες ρυθμίσεις τόσο κατά την αρχικοποίηση τους (π.χ απενεργοποίηση πρωτοκόλλων και υπηρεσιών που δεν χρησιμοποιούνται) αλλά και σε όλη την διάρκεια χρήσης τους (π.χ ενημέρωση με αναβαθμίσεις που αντιμετωπίζουν νέες ευπάθειες) ώστε να επιτυγχάνεται η ασφαλή λειτουργία τους.

1.2.3 Επίπεδο Δικτύου

Το δίκτυο αποτελεί το μέσο που καθιστά την εφαρμογή μας προσβάσιμη στους δυνητικούς χρήστες αλλά και στους επιτιθέμενους. Απαρτίζεται συνήθως από α) routers (δρομολογητές) β) firewalls (τείχη ασφαλείας) και γ) switches (μεταγωγείς). Οι επιθέσεις που δέχεται ένα δίκτυο βασίζονται στο πρωτόκολλο TCP/IP και θα πρέπει να είναι δομημένο έτσι ώστε να μπορεί να τις αντιμετωπίζει. Επίσης πρέπει να διασφαλίζει την ακεραιότητας των δεδομένων που διακινεί.

1.3 Βασικές Αρχές Ασφαλείας

Οι βασικές αρχές ασφαλείας σύμφωνα με τον J.D Meier [3] που θα πρέπει να τηρούνται πιστά για την υλοποίηση μιας διαδικτυακής εφαρμογής είναι οι παρακάτω:

1.3.1 Υιοθέτηση της αρχής των ελάχιστων προνομίων

Διεργασίες που τρέχουν scripts ή εκτελέσιμο κώδικα θα πρέπει να γίνονται με την χρήση λογαριασμών που έχουν όσο το δυνατόν λιγότερα προνόμια (δικαιώματα πρόσβασης) ώστε να ελαχιστοποιηθεί ο κίνδυνος σε περίπτωση που αποκτήσει κάποιος πρόσβαση.

1.3.2 Χρήση της άμυνας σε βάθος

Η τοποθέτηση σημείων ελέγχου σε κάθε ένα από τα στρώματα και τα υποσυστήματα μέσα στην εφαρμογή αποτελούν τους φύλακες που διασφαλίζουν ότι μόνο οι εξουσιοδοτημένοι χρήστες έχουν τη δυνατότητα να μεταβούν στον επόμενο προς τα κάτω στρώμα.

1.3.3 Έλλειψη εμπιστοσύνης στις εκχωρήσεις του χρήστη

Οι εφαρμογές θα πρέπει να μην εμπιστεύονται και να ελέγχουν εξονυχιστικά την εγκυρότητα των εκχωρήσεων (inputs) του χρήστη πριν τις χρησιμοποιήσουν για να πραγματοποιήσουν λειτουργίες.

1.3.4 Χρήση προεπιλογών ασφαλείας

Αν η εφαρμογή απαιτεί να ενεργοποιηθούν χαρακτηριστικά τα οποία αλλάζουν τις αρχικές ρυθμίσεις ασφαλείας τότε πρέπει να ελεγχθεί η επίδραση που θα έχουν πριν να πραγματοποιηθεί η ενεργοποίηση τους.

1.3.5 Η ασφάλεια δεν μπορεί να βασιστεί εξ' ολοκλήρου στην απόκρυψη

Η προσπάθεια απόκρυψης πληροφοριών με την χρήση παραπλανητικών ονομάτων μεταβλητών ή με την αποθήκευση τους σε περίεργες τοποθεσίες αρχείων δεν παρέχει ασφάλεια. Είναι προτιμότερη η χρήση δυνατοτήτων της πλατφόρμας ή αποδεδειγμένων ορθών τεχνικών για την ασφάλεια των δεδομένων.

1.3.6 Έλεγχος του χρήστη στην είσοδο

Με τον έλεγχο από το πρώτο σημείο καθορίζεται στον εξουσιοδοτημένο χρήστη τι δικαιώματα πρόσβασης μπορεί να έχει.

1.3.7 Μη ασφαλή εξωτερικά συστήματα

Τα εξωτερικά συστήματα που χρησιμοποιούνται θα θεωρούνται ως ανασφαλή. Εφόσον δεν ανήκουν στον κάτοχο της εφαρμογής δεν μπορούν να τα θεωρούνται ασφαλή.

1.3.8 Μείωση του μεγέθους της περιοχής ασφαλείας

Αποφυγή της έκθεσης πληροφοριών που δεν είναι απαραίτητη. Αν δεν γίνει αυτό είναι πιθανόν να ανοίγουν πόρτες που μπορεί να οδηγήσουν σε πρόσθετες ευπάθειες.

1.3.9 Κατάσταση ασφαλείας σε περίπτωση αποτυχίας

Αν η εφαρμογή εμφανίσει πρόβλημα λειτουργίας πρέπει να διασφαλιστεί ότι δεν αφήνει απροστάτευτα ευαίσθητα δεδομένα. Επίσης δεν πρέπει να παρέχει πολλές λεπτομέρειες στα μηνύματα λάθους που θα βοηθούσαν τον επιτιθέμενο να εκμεταλλευτεί μια αδυναμία.

1.3.10 Η ασφάλεια αφορά όλα τα επίπεδα και τις βαθμίδες της εφαρμογής

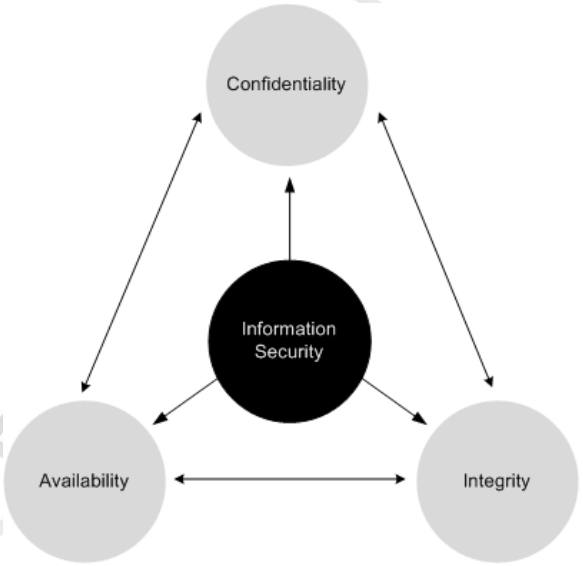
Η αρχή είναι ότι ο βαθμός ασφαλείας εξαρτάται από το επίπεδο ασφαλείας του πιο αδύναμου κρίκου της εφαρμογής.

1.3.11 Ότι δεν χρησιμοποιείται πρέπει να απενεργοποιηθεί

Υπάρχει η δυνατότητα να μειωθούν τα πιθανά σημεία που μπορούν να δεχτούν επίθεση με την απενεργοποίηση ενοτήτων (modules) και συστατικών (components) που η εφαρμογή δεν χρειάζεται.

1.4 Περιγραφή του υπό μελέτη προβλήματος

Ο πρωταρχικός στόχος του ανθρώπου ήταν η επικοινωνία με την χρήση των ηλεκτρονικών υπολογιστών χωρίς να δίνει μεγάλη σημασία στον βαθμό ασφάλειας αυτής της επικοινωνίας. Μέσα σε ένα εύλογο χρονικό διάστημα η επικοινωνία αυτή επετεύχθη και έγινε ένα αναπόσπαστο κομμάτι της καθημερινότητάς μας, με την ταυτόχρονη όμως δημιουργία της ανάγκης υψηλών απαιτήσεων ασφαλείας. Η ανάγκη αυτή δημιουργήθηκε από την χρήση διαδικτυακών εφαρμογών (web applications) για την εξυπηρέτηση ολοένα και περισσότερων καθημερινών μας δραστηριοτήτων (π.χ. τραπεζικές συναλλαγές, ηλεκτρονικές αγορές προϊόντων, συνδιαλλαγές με υπηρεσίες του κράτους). Αποτέλεσμα αυτής της ανάγκης ήταν λέξεις όπως εμπιστευτικότητα, διαθεσιμότητα, ακεραιότητα να ηχούν όλο και πιο οικείες στα αυτιά μας.



Εικόνα 2. Εμπιστευτικότητα Ακεραιότητα Διαθεσιμότητα

Στην εργασία αυτή θα μελετηθεί πώς μπορεί να ελεγχθεί ο βαθμός ασφαλείας μιας διαδικτυακής εφαρμογής. Αρχικά θα αναλυθούν ποιες είναι οι συνηθισμένες αδυναμίες – προβλήματα ασφάλειας τέτοιου είδους εφαρμογών όπως τις παρουσιάζει ο OWASP τα τελευταία χρόνια καθώς και τους τρόπους αποτροπής τους. Θα παρουσιαστεί η εφαρμογή (Online Movies) που αποτελεί την δοκιμαστική εφαρμογή και στην οποία θα γίνουν οι απαραίτητοι έλεγχοι.

Έπειτα Θα παρουσιαστεί αναλυτικά το εργαλείο εντοπισμού αδυναμιών (Nessus) και ο τρόπος λειτουργίας του. Θα δημιουργηθεί η κατάλληλη πολιτική ελέγχου, όπου θα χρησιμοποιηθεί για τον έλεγχο της εφαρμογής και θα αναλυθούν οι αδυναμίες που εντοπίστηκαν. Τέλος θα αναπτυχθούν τα κατάλληλα μέτρα προστασίας για τις συγκεκριμένες αδυναμίες διορθώνοντας τον κώδικα της εφαρμογής (Security reengineering).

1.5 Σκοπός και στόχοι της εργασίας

Οι στόχοι της εργασίας είναι :

- Να αναλυθούν οι κίνδυνοι ασφαλείας των διαδικτυακών εφαρμογών.
- Να αναλυθούν ποιοι είναι οι τρόποι που μπορούν να αποτρέψουν τους ανωτέρω κινδύνους.
- Να περιγραφεί το εργαλείο εντοπισμού ευπαθειών (vulnerability scanner) που θα χρησιμοποιηθεί, δηλαδή το Nessus.
- Να χρησιμοποιηθεί το εργαλείο εντοπισμού ευπαθειών πάνω σε δοκιμαστική διαδικτυακή εφαρμογή .NET.
- Να αναλυθούν και να περιγραφούν οι ευπάθειες (vulnerabilities) που εντοπίστηκαν.
- Να παρουσιαστούν οι τρόποι προστασίας από τις ευπάθειες και να αναπτυχθούν τα αντίστοιχα μέτρα προστασίας με την κατάλληλη τροποποίηση του κώδικα της εφαρμογής.

1.6 Βασικοί ορισμοί

1.6.1 Εμπιστευτικότητα (confidentiality)

Εμπιστευτικότητα είναι η ιδιότητα της προστασίας του περιεχομένου της πληροφορίας από όλους, εκτός από τους χρήστες εκείνους που έχει εγκρίνει ο νόμιμος κάτοχος της πληροφορίας. Οι μη εγκεκριμένοι χρήστες συνήθως καλούνται μη εξουσιοδοτημένοι χρήστες. Άλλοι όροι, όπως η ιδιωτικότητα (privacy), χρησιμοποιούνται σχεδόν συνώνυμα με την εμπιστευτικότητα. Παρόλα αυτά ο όρος ιδιωτικότητα αναπαριστά μια ανθρώπινη ιδιότητα (και όχι μια ιδιότητα της πληροφορίας) η οποία συνήθως δεν είναι εύκολα μετρήσιμη. [4]

1.6.2 Διαθεσιμότητα (Availability)

Η διαθεσιμότητα είναι η ιδιότητα της προστασίας της πληροφορίας από μη εξουσιοδοτημένη, προσωρινή ή μόνιμη παρακράτηση της. Δηλαδή σημαίνει ότι οι υπηρεσίες ενός δικτύου υπολογιστών είναι προσπελάσιμες και χωρίς αδικαιολόγητη καθυστέρηση όποτε απαιτείται η χρήση τους από μια εξουσιοδοτημένη οντότητα. [4]

1.6.3 Ακεραιότητα (Integrity)

Η ακεραιότητα είναι η ιδιότητα της προστασίας της πληροφορίας από την τροποποίησή της από μη εξουσιοδοτημένους χρήστες. Πρόκειται για την επιβεβαίωση ότι τα δεδομένα που έχουν αποσταλεί, παραληφθεί ή αποθηκευτεί είναι πλήρη και δεν έχουν υποστεί αλλοίωση. Επίσης σημαίνει πρόληψη μη εξουσιοδοτημένης μεταβολής πληροφοριών, δηλαδή, πρόληψη από μη εξουσιοδοτημένη εγγραφή ή διαγραφή, συμπεριλαμβανομένης και της μη εξουσιοδοτημένης δημιουργίας δεδομένων. [4]

1.6.4 Απειλή (threat)

Απειλή είναι κάθε δυνητικός κίνδυνος που μπορεί να βλάψει ένα αγαθό. Όταν μια απειλή εκδηλώνεται, το σύστημα ή το δίκτυο είναι «υπό επίθεση». Ο επιτιθέμενος ή φορέας της επίθεσης (attacker, threat agent) είναι κάθε υποκείμενο ή οντότητα που προκαλεί την επίθεση [5], [6].

1.6.5 Επίθεση (attack)

Η εκδήλωση μιας απειλής είναι η επίθεση. Αυτή ορίζεται ως η προσπάθεια ενός δυνητικού εισβολέα εκμεταλλευόμενος τις ευπάθειες των εφαρμογών να τις καταστρέψει, εκθέσει, αλλοιώσει, υποκλέψει ή αποκτήσει σε αυτές μη εξουσιοδοτημένη πρόσβαση ή να τις χρησιμοποιήσει χωρίς εξουσιοδότηση.

1.6.6 Συνέπεια (Impact)

Η συνέπεια της απειλής μετράει την έκταση της απώλειας που θα προκληθεί στο αγαθό ή στον ιδιοκτήτη του αγαθού, εφόσον η απειλή πραγματοποιηθεί εναντίον του αγαθού.

1.6.7 Αδυναμία ασφάλειας (security vulnerability)

Αδυναμία ασφάλειας είναι μια αδυναμία σε ένα προϊόν (π.χ. εφαρμογή), που ο προγραμματιστής δεν είχε σκοπό να εισάγει και πρέπει να διορθώσει όταν ανακαλυφθεί, η οποία μπορεί να επιτρέψει σε έναν εισβολέα να εκθέσει την ακεραιότητα, την διαθεσιμότητα ή την εμπιστευτικότητα του προϊόντος [7].

1.6.8 Ευπάθεια (exploit)

Ο όρος ευπάθεια αναφέρεται σε ένα κακόβουλο πρόγραμμα ή κομμάτι κώδικα, το οποίο καθώς εκτελείται μπορεί και εκμεταλλεύεται κάποιο bug ή κενό ασφαλείας μιας εφαρμογής ή του λειτουργικού συστήματος και παρέχει μη εξουσιοδοτημένη πρόσβαση ή αύξηση δικαιωμάτων στον επιτιθέμενο.

1.6.9 Εργαλείο εντοπισμού αδυναμιών (vulnerability scanner)

Ένα εργαλείο εντοπισμού αδυναμιών είναι ένα πρόγραμμα που έχει σχεδιαστεί να προσδιορίζει, αναγνωρίζει και ταξινομεί τις αδυναμίες ασφάλειας σε υπολογιστές, εξυπηρετητές (server), δίκτυα ή εφαρμογές.

1.6.10 Αντίμετρα (countermeasures)

Τα αντίμετρα είναι ενέργειες, διεργασίες, διαδικασίες ή συστήματα που μπορούν να αποτρέψουν ή να μετριάσουν τις συνέπειες των επιθέσεων ασφάλειας. Η μη παρουσία του απαραίτητου αντιμέτρου ή μια αδυναμία στην σχεδίαση και υλοποίηση του δίνουν την δυνατότητα να καταστεί μια εφαρμογή ευάλωτη σε επιθέσεις.

1.6.11 Επικινδυνότητα Ασφάλειας (risk)

Επικινδυνότητα ασφάλειας είναι οποιοδήποτε γεγονός ή ενέργεια το οποίο θα μπορούσε να προκαλέσει απώλεια ή ζημιά στο hardware του υπολογιστή, στο λογισμικό του, στα δεδομένα ή σε πληροφορίες του.

1.6.12 HyperText Transfer Protocol (HTTP)

Το http είναι το πρωτόκολλο που χρησιμοποιείται από το διαδίκτυο (world wide web) για την ορθή λειτουργία του. Το http ορίζει ποιά θα είναι η μορφή των μηνυμάτων και πως αυτά θα μεταδοθούν. Επίσης ορίζει τις δράσεις των server και browser στις διάφορες εντολές που απευθύνονται σε αυτούς. Για παράδειγμα όταν εισάγουμε ένα url στον browser, αυτό στην πραγματικότητα στέλνει μια εντολή http στον web server καθοδηγώντας τον ώστε να μας αποστείλει την ζητούμενη ιστοσελίδα.

1.7 Παραδοτέα της εργασίας

Η παρούσα μεταπτυχιακή εργασία αποτελείται από τα παρακάτω:

1. Το έντυπο κείμενο της πτυχιακής εργασίας.
2. Ένα CD που θα περιέχει την δοκιμαστική εφαρμογή (Online Movies), η οποία αναπτύχθηκε σε .NET περιβάλλον, την πτυχιακή εργασία σε ηλεκτρονική μορφή και δυο αρχεία μορφής html που θα περιέχουν τις αναφορές από τους ελέγχους στην εφαρμογή.

1.8 Δομή της εργασίας

Η εργασία έχει την ακόλουθη δομή: Στο Κεφάλαιο 1 (Εισαγωγή στην Ασφάλεια Διαδικτυακών Εφαρμογών), γίνεται μια αναφορά σε βασικούς ορισμούς και αρχές της ασφάλειας. Στο Κεφάλαιο 2 (Επισκόπηση Κινδύνων Ασφάλειας Διαδικτυακών Εφαρμογών) γίνεται εκτενής ανάλυση και κατηγοριοποίηση των ευπαθειών όσον αφορά την ασφάλεια των εφαρμογών και τρόποι αποτροπής των ανωτέρω ευπαθειών. Στο Κεφάλαιο 3 (Περιγραφή του Εργαλείου Εντοπισμού Ευπαθειών Nessus) παρουσιάζονται αναλυτικά οι λειτουργίες και οι δυνατότητες του συγκεκριμένου εργαλείου.

Στο Κεφάλαιο 4 (Ανάπτυξη Δοκιμαστικής Εφαρμογής και Έλεγχος Ευπαθειών) παρουσιάζεται συνοπτικά η εφαρμογή ιστού που στη συνέχεια θα ελεγχθεί και γίνεται χρήση του εργαλείου εντοπισμού ευπαθειών για τον έλεγχο του βαθμού ασφαλείας της εφαρμογής όσον αφορά την εύρεση ευπαθειών. Στο Κεφάλαιο 5 (Ανάλυση Ευπαθειών της Διαδικτυακής Εφαρμογής) γίνεται αναλυτική περιγραφή των ευπαθειών που εντοπίστηκαν και στο Κεφάλαιο 6 (Ανασχεδιασμός Ασφάλειας και Εφαρμογή Μέτρων Ασφαλείας) αναπτύσσονται τα κατάλληλα μέτρα προστασίας στην εφαρμογή που ελέγχθηκε ώστε να καταστεί ασφαλής. Στο Κεφάλαιο 7 (Συμπεράσματα) παρατίθενται τα συμπεράσματα που εξήχθησαν από την παρούσα εργασία και στο Κεφάλαιο 8 (Βιβλιογραφία) παρατίθεται η βιβλιογραφία που χρησιμοποιήθηκε.

1.9 Πλάνο υλοποίησης

Το πλάνο υλοποίησης της παρούσας εργασίας παρατίθεται στον παρακάτω πίνακα αναλυτικά όσον αφορά την παράδοση των κεφαλαίων στις προβλεπόμενες ημερομηνίες.

ΠΑΡΑΔΟΤΕΑ	ΠΕΡΙΓΡΑΦΗ	ΕΝΑΡΞΗ	ΛΗΞΗ
Π1 (Προσχέδιο εργασίας)	Πρόταση εκπόνησης μεταπτυχιακής εργασίας	01/06/2014	25/06/2014
Π2 (Κεφάλαια 1,2)	Θεωρητική μελέτη κενών ασφαλείας σε δικτυακές εφαρμογές	26/06/2014	31/07/2014
Π3 (Κεφάλαιο 3)	Θεωρητική μελέτη εργαλείων εντοπισμού ευπαθειών (vulnerability scanning tools) σε δικτυακές εφαρμογές	01/08/2014	31/08/2014
Π4 (Κεφάλαιο 4)	Εφαρμογή ελέγχου για εύρεση αδυναμιών στην δοκιμαστική εφαρμογή	01/09/2014	30/09/2014
Π5 (Κεφάλαια 5,6)	Εντοπισμός και διόρθωση σφαλμάτων κώδικα	01/10/2014	31/10/2014
Π6 (Κεφάλαια 1-8)	Παράδοση τελικής έκδοσης μεταπτυχιακής εργασίας	01/11/2014	30/11/2014

Πίνακας 1. Πίνακας υλοποίησης μεταπτυχιακής εργασίας

Κεφάλαιο 2ο

2 Επισκόπηση Κίνδυνων Ασφάλειας Διαδικτυακών Εφαρμογών

2.1 Εισαγωγή

Ο σκοπός αυτού του κεφαλαίου είναι η παρουσίαση και κατηγοριοποίηση των κινδύνων ασφαλείας που σχετίζονται με τις διαδικτυακές εφαρμογές. Για την απόκτηση μιας σφαιρικής εικόνας και κατανόησης της διαχρονικής εξέλιξης τους θα αναφερθούν από το 2003 μέχρι και σήμερα οι δέκα πιο σημαντικοί κίνδυνοι σύμφωνα με τα πρότυπα του OWASP [8].

Ο OWASP είναι η σύντμηση του Open Web Application Security Project που είναι ένας παγκόσμιος μη κερδοσκοπικός οργανισμός με σκοπό την βελτίωση της ασφάλειας του λογισμικού [9]. Σκοπός όλων αυτών που τον απαρτίζουν είναι να ενημερώνουν την παγκόσμια κοινότητα για τις πραγματικές επικινδυνότητες ασφάλειας. Τέλος, θα δοθούν οι ορισμοί των κινδύνων για την τελευταία δημοσίευση του (OWASP) που έγινε το 2013 και οι τρόποι που προτείνονται για την αποτροπή τους.

2.2 Δημοσιεύσεις OWASP

Στην πλειοψηφία τους οι οργανισμοί βασίζονται αποκλειστικά σε περιστασιακούς ελέγχους ή δοκιμές διείσδυσης για τη διασφάλιση των εσωτερικών και εξωτερικών εφαρμογών διαδικτύου που χρησιμοποιούν. Η προσέγγιση αυτή έχει υψηλό κόστος και δεν παρέχει πολλές πληροφορίες σχετικά με την αντιμετώπιση των κινδύνων. Όπως και για άλλους τομείς της ασφάλειας έτσι και για την ασφάλεια λογισμικού εφαρμογών απαιτείται ένα πρόγραμμα διαχείρισης της επικινδυνότητας που παρέχει πληροφορίες για τις εφαρμογές και τα στρατηγικά μέτρα ελέγχου που στοχεύουν στη βελτίωση της ασφάλειας.

Από το 2003 ερευνητές και ειδικοί της ασφάλειας λογισμικού εφαρμογών από όλο τον κόσμο παρακολουθούν προσεκτικά μέσα από τον OWASP την κατάσταση της ασφάλειας των διαδικτυακών εφαρμογών και δημοσιεύουν ένα κείμενο ευαισθητοποίησης που αποτελεί πλέον πρότυπο στο οποίο βασίζονται διεθνείς οργανισμοί όπως οι PCI, DOD, FTC, και πολλοί ακόμα.

Το κείμενο αυτό που δημοσιεύει ο OWASP αποτελεί μια ενημερωμένη έκθεση που αποτυπώνει τους δέκα σημαντικότερους κινδύνους που σχετίζονται με τη χρήση διαδικτυακών εφαρμογών σε έναν οργανισμό. Η έκθεση αυτή είναι γεμάτη με παραδείγματα και αναλύσεις που εξηγούν τους κινδύνους αυτούς σε όσους αναπτύσσουν λογισμικό, σε επικεφαλής ομάδων ανάπτυξης και οργανισμών και γενικότερα, σε όσους ενδιαφέρονται για το μέλλον της ασφάλειας στο διαδίκτυο.

2.2.1 Δημοσίευση OWASP 2003

Η πρώτη δημοσίευση του OWASP που πραγματοποιήθηκε το 2003 και αποτέλεσε το έναυσμα για την συνέχεια αυτής της προσπάθειας και των μετέπειτα ενημερώσεων και ανανεώσεων της σχετικής λίστας αποτελείται από τους 10 σημαντικότερους κινδύνους με αριθμό προτεραιότητας από το A1 έως το A10 και παρουσιάζεται στον Πίνακα 2.

OWASP TOP 10 2003	
A1	Unvalidated Parameters (Μη Επικύρωση Παραμέτρων)
A2	Broken Access Control (Εσφαλμένος Έλεγχος Πρόσβασης)
A3	Broken Account and Session Management (Εσφαλμένη Διαχείριση Λογαριασμών και Συνόδων)
A4	Cross Site Scripting (XSS) Flaws (Σφάλματα XSS)
A5	Buffer Overflows (Σφάλματα Υπερχείλισης)
A6	Command Injection Flaws (Σφάλματα Έγχυσης Εντολών)
A7	Error Handling Problems (Προβλήματα Χειρισμού Σφαλμάτων)
A8	Insecure Use of Cryptography (Μη Ασφαλής Χρήση Κρυπτογραφίας)
A9	Remote Administration Flaws (Σφάλματα Απομακρυσμένης Διαχείρισης)
A10	Web and Application Server Misconfiguration (Σφάλματα Διαμόρφωσης Εξυπηρετητών)

Πίνακας 2. Κίνδυνοι Ασφάλειας των Διαδικτυακών Εφαρμογών με βάση την έκδοση 2003 του OWASP

2.2.2 Δημοσίευση OWASP 2004

Στη συνέχεια, το 2004 έγινε η πρώτη ανανέωση, όπως παρουσιάζεται στον Πίνακα 3 μαζί με την συσχέτιση με τους αντιστοίχους κινδύνους του 2003.

NEW OWASP TOP 10 2004		OWASP TOP 10 2003	
A1	Unvalidated Input (Μη Επικύρωση Εκχώρησης)	A1	Unvalidated Parameters (Μη Επικύρωση Παραμέτρων)
A2	Broken Access Control (Εσφαλμένος Έλεγχος Πρόσβασης)	A2 (A9)	Broken Access Control (Εσφαλμένος Έλεγχος Πρόσβασης) (Remote Administration Flaws) (Σφάλματα Απομακρυσμένης Διαχείρισης)
A3	Broken Authentication and Session Management (Εσφαλμένη Διαχείριση Λογαριασμών και Συνόδων)	A3	Broken Account and Session Management (Εσφαλμένη Διαχείριση Λογαριασμών και Συνόδων)
A4	Cross Site Scripting (XSS) Flaws (Σφάλματα XSS)	A4	Cross Site Scripting (XSS) Flaws (Σφάλματα XSS)
A5	Buffer Overflows (Σφάλματα Υπερχείλισης)	A5	Buffer Overflows (Σφάλματα Υπερχείλισης)
A6	Injection Flaws (Σφάλματα Έγχυσης)	A6	Command Injection Flaws (Σφάλματα Έγχυσης Εντολών)
A7	Improper Error Handling (Μη Κατάλληλος Χειρισμός Σφαλμάτων)	A7	Error Handling Problems (Προβλήματα Χειρισμού Σφαλμάτων)
A8	Insecure Storage (Μη Ασφαλής Αποθήκευση)	A8	Insecure Use of Cryptography (Μη Ασφαλής Χρήση Κρυπτογραφίας)
A9	Denial of Service (Άρνηση Εξυπηρέτησης)	A9	
A10	Insecure Configuration Management (Μη Ασφαλής Διαχείριση Διαμόρφωσης)	A10	Web and Application Server Misconfiguration (Σφάλματα Διαμόρφωσης Εξυπηρετητών)

Πίνακας 3. Κίνδυνοι Ασφάλειας των Διαδικτυακών Εφαρμογών με βάση την έκδοση 2004 του OWASP και σύγκριση με την έκδοση 2003

2.2.3 Δημοσίευση OWASP 2007

Στον Πίνακα 4 παρουσιάζεται η ανανεωμένη λίστα του 2007 σε συσχέτιση με τους κινδύνους του 2004.

NEW OWASP TOP 10 2007		OWASP TOP 10 2004	
A1	Cross Site Scripting (XSS) (Ατέλειες Τύπου XSS)	A4	Cross Site Scripting (XSS) Flaws (Σφάλματα XSS)
A2	Injection Flaws (Σφάλματα Έγχυσης)	A6	Injection Flaws (Σφάλματα Έγχυσης)
A3	Malicious File Execution (Εκτέλεση Κακόβουλου Αρχείου)		
A4	Insecure Direct Object Reference (Μη Ασφαλής Απευθείας Αναφορά σε Αντικείμενα)	A2	Broken Access Control - Split in 2007 Top 10 (Εσφαλμένος Έλεγχος Πρόσβασης - Διαχωρίστηκε στο Top 10 του 2007)
A5	Cross Site Request Forgery (CSRF) (Πλαστογράφηση Αίτησης Μεταξύ Θέσεων)		
A6	Information Leakage and Improper Error Handling (Διαρροή Πληροφοριών και Μη Κατάλληλος Χειρισμός Σφαλμάτων)	A7	Improper Error Handling (Μη Κατάλληλος Χειρισμός Σφαλμάτων)
A7	Broken Authentication and Session Management (Εσφαλμένη Διαχείριση Λογαριασμών και Συνόδων)	A3	Broken Authentication and Session Management (Εσφαλμένη Διαχείριση Λογαριασμών και Συνόδων)
A8	Insecure Cryptographic Storage (Μη Ασφαλής Κρυπτογραφημένη Αποθήκευση)	A8	Insecure Storage (Μη Ασφαλής Αποθήκευση)
A9	Insecure Communications (Μη Ασφαλής Επικοινωνίες)		
A10	Failure to Restrict URL Access (Αποτυχία Περιορισμού της Πρόσβασης URL)	A2	Broken Access Control - Split in 2007 Top 10 (Εσφαλμένος Έλεγχος Πρόσβασης - Διαχωρίστηκε στο Top 10 του 2007)
	(Removed in 2007) (Αφαιρέθηκε το 2007)	A1	Unvalidated Input (Μη Επικύρωση Εκχώρησης)
	(Removed in 2007) (Αφαιρέθηκε το 2007)	A5	Buffer Overflows (Σφάλματα Υπερχείλισης)
	(Removed in 2007) (Αφαιρέθηκε το 2007)	A9	Denial of Service (Άρνηση Εξυπηρέτησης)
	(Removed in 2007) (Αφαιρέθηκε το 2007)	A10	Insecure Configuration Management (Μη Ασφαλής Διαχείριση Διαμόρφωσης)

Πίνακας 4. Κίνδυνοι Ασφάλειας των Διαδικτυακών Εφαρμογών με βάση την έκδοση 2007 του OWASP και σύγκριση με την έκδοση 2004

2.2.4 Δημοσίευση OWASP 2010

Η ενημερωμένη έκδοση του 2010 [10] βασίζεται σε περισσότερες πηγές πληροφόρησης για ευπάθειες διαδικτυακών εφαρμογών σε σχέση με την προηγούμενη. Επιπλέον, παρουσιάζει τις πληροφορίες με πιο συνοπτικό τρόπο, ώστε να μπορούν εύκολα να τεθούν σε εφαρμογή αφού περιλαμβάνουν πολλές αναφορές σε νέο, πλούσιο υλικό που μπορεί να χρησιμοποιηθεί για να αντιμετωπίσει το κάθε θέμα, όπως το νέο Enterprise Security API (ESAPI) του OWASP και το Πρότυπο Ελέγχου Επαλήθευσης Ασφάλειας Εφαρμογών (Application Security Verification Standard – ASVS). [11]

Enterprise Security API

Το νέο Enterprise Security API του OWASP [12] είναι μια δωρεάν, ανοιχτού κώδικα, βιβλιοθήκη ελέγχου ασφαλείας των διαδικτυακών εφαρμογών που καθιστά ευκολότερο για τους προγραμματιστές να γράψουν εφαρμογές χαμηλότερου κινδύνου. Οι βιβλιοθήκες του ESAPI έχουν σχεδιαστεί ώστε να είναι ευκολότερο για τους προγραμματιστές να τροποποιούν την ασφάλεια σε υπάρχουσες εφαρμογές. Χρησιμεύουν επίσης ως ένα στερεό θεμέλιο για την ανάπτυξη νέων εφαρμογών.

Εκτός από συγκεκριμένες διαφορές που παρουσιάζονται ανάλογα με την γλώσσα προγραμματισμού που χρησιμοποιήθηκε για την υλοποίηση τους, όλες οι εκδόσεις του OWASP ESAPI έχουν το ίδιο βασικό σχέδιο:

- Υπάρχει ένα σύνολο από διεπαφές ελέγχου ασφαλείας. Καθορίζουν για παράδειγμα τους τύπους των παραμέτρων που περνάνε στα είδη των ελέγχων ασφαλείας.
- Υπάρχει για κάθε έλεγχο ασφαλείας η εκπλήρωση μιας αναφοράς. Η λογική δεν είναι να γίνουν όλα στα πλαίσια ειδικά ενός οργανισμού και ειδικά μιας εφαρμογής. Ένα παράδειγμα είναι μια αναφορά για μια συμβολοσειρά που βασίζεται στην επικύρωση των εκχωρήσεων (string-based valid input).
- Υπάρχουν προαιρετικά υλοποιήσεις για κάθε έλεγχο ασφαλείας, όπου θα ταιριάζουν στην λογική της εφαρμογής που ελέγχεται και θα είναι προσαρμοσμένες για την συγκεκριμένη χρήση.

Αρκετοί οργανισμοί πια έχουν υιοθετήσει το ESAPI για να καταστήσουν ασφαλείς τις διαδικτυακές τους εφαρμογές.

Πρότυπο Ελέγχου Επαλήθευσης Ασφάλειας Εφαρμογών (Application Security Verification Standard – ASVS)

Όσον αφορά το Πρότυπο Ελέγχου Επαλήθευσης Ασφάλειας Εφαρμογών (OWASP ASVS) [13], [14], [15] πρωταρχικός στόχος του είναι να κανονικοποιήσει το εύρος της κάλυψης και τα επίπεδα αυστηρότητας που διατίθενται στην αγορά, όταν πρόκειται να διενεργηθεί επαλήθευση του ελέγχου ασφάλειας των διαδικτυακών εφαρμογών με την χρήση ενός εμπορικά-εφαρμόσιμου ανοιχτού προτύπου. Το πρότυπο παρέχει μια βάση για τεχνικούς ελέγχους ασφάλειας των εφαρμογών, καθώς και τυχόν τεχνικούς ελέγχους ασφαλείας στο περιβάλλον, που χρησιμοποιούνται για την προστασία από ευπάθειες, όπως CROSS-SITE SCRIPTING (XSS) και SQL injection. Αυτό το πρότυπο μπορεί να χρησιμοποιηθεί για να δημιουργηθεί ένα επίπεδο εμπιστοσύνης στην ασφάλεια των διαδικτυακών εφαρμογών. Οι απαιτήσεις αναπτύχθηκαν με σκοπό τους ακόλουθους στόχους:

- Χρήση ως «μετρικό» – Παρέχει στους προγραμματιστές εφαρμογών και στους ιδιοκτήτες εφαρμογών ένα κριτήριο με το οποίο μπορούν να εκτιμήσουν το βαθμό της εμπιστοσύνης που μπορούν να έχουν στις εφαρμογές τους.
- Χρήση ως «οδηγός» – Παρέχει καθοδήγηση στους υπεύθυνους ανάπτυξης ελέγχων ασφάλειας ως προς το τι να ενσωματώσουν στους ελέγχους ασφαλείας προκειμένου να ικανοποιήσουν τις απαιτήσεις της ασφάλειας των εφαρμογών,
- Χρήση κατά τη διάρκεια της προμήθειας - Παρέχουν μια βάση για τον καθορισμό των απαιτήσεων ελέγχου ασφάλειας των εφαρμογών στη σύναψη συμβάσεων.

Στον Πίνακα 5 παρουσιάζονται οι κίνδυνοι του 2010 [16] και η συσχέτιση τους με τους αντίστοιχους του 2007.

NEW OWASP TOP 10 2010		OWASP TOP 10 2007	
A1	Injection (Έγχυση)	A2	Injection Flaws (Σφάλματα Έγχυσης)
A2	Cross Site Scripting (XSS) (Ατέλειες Τύπου XSS)	A1	Cross Site Scripting (XSS) (Ατέλειες Τύπου XSS)
A3	Broken Authentication and Session Management (Εσφαλμένη Διαχείριση Λογαριασμών και Συνόδων)	A7	Broken Authentication and Session Management (Εσφαλμένη Διαχείριση Λογαριασμών και Συνόδων)
A4	Insecure Direct Object Reference (Μη Ασφαλής Απευθείας Αναφορά σε Αντικείμενα)	A4	Insecure Direct Object Reference (Μη Ασφαλής Απευθείας Αναφορά σε Αντικείμενα)
A5	Cross Site Request Forgery (CSRF) (Πλαστογράφηση Αίτησης Μεταξύ Θέσεων)	A5	Cross Site Request Forgery (CSRF) (Πλαστογράφηση Αίτησης Μεταξύ Θέσεων)
A6	Security Misconfiguration (Σφάλματα Διαμόρφωσης Ασφαλείας)		(Was Top 10 2004 A10 - Insecure Configuration Management) (Ήταν το A10 στο Top 10 του 2004 - Μη Ασφαλής Διαχείριση Διαμόρφωσης)
A7	Insecure Cryptographic Storage (Μη Ασφαλής Κρυπτογραφημένη Αποθήκευση)	A8	Insecure Cryptographic Storage (Μη Ασφαλής Κρυπτογραφημένη Αποθήκευση)
A8	Failure to Restrict URL Access (Αποτυχία Περιορισμού της Πρόσβασης URL)	A10	Failure to Restrict URL Access (Αποτυχία Περιορισμού της Πρόσβασης URL)
A9	Insufficient Transport Layer Protection (Ανεπαρκής Προστασία Επιπέδου Μεταφοράς)	A9	Insecure Communications (Μη Ασφαλείς Επικοινωνίες)
A10	Unvalidated Redirects and Forwards (Μη Επικύρωση Ανακατευθύνσεων και Προωθήσεων)		
	(Removed in 2010) (Αφαιρέθηκε το 2010)	A3	Malicious File Execution (Εκτέλεση Κακόβουλου Αρχείου)
	(Removed in 2010) (Αφαιρέθηκε το 2010)	A6	Information Leakage and Improper Error Handling (Διαρροή Πληροφοριών και Μη Κατάλληλος Χειρισμός Σφαλμάτων)

Πίνακας 5. Κίνδυνοι Ασφάλειας των Διαδικτυακών Εφαρμογών με βάση την έκδοση 2010 του OWASP και σύγκριση με την έκδοση 2007

2.2.5 Δημοσίευση OWASP 2013

Ο επόμενος πίνακας παρουσιάζει την τελευταία δημοσίευση του OWASP που έγινε το 2013 και παραθέτει τις αλλαγές σε σχέση με την δημοσίευση του 2010.

NEW OWASP TOP 10 2013		OWASP TOP 10 2010	
A1	Injection (Έγχυση)	A1	Injection (Έγχυση)
A2	Broken Authentication and Session Management (Εσφαλμένη Διαχείριση Λογαριασμών και Συνόδων)	A3	Broken Authentication and Session Management (Εσφαλμένη Διαχείριση Λογαριασμών και Συνόδων)
A3	Cross Site Scripting (XSS) (Ατέλειες Τύπου XSS)	A2	Cross Site Scripting (XSS) (Ατέλειες Τύπου XSS)
A4	Insecure Direct Object Reference (Μη Ασφαλής Απευθείας Αναφορά σε Αντικείμενα)	A4	Insecure Direct Object Reference (Μη Ασφαλής Απευθείας Αναφορά σε Αντικείμενα)
A5	Security Misconfiguration (Σφάλματα Διαμόρφωσης Ασφαλείας)	A6	Security Misconfiguration (Σφάλματα Διαμόρφωσης Ασφαλείας)
A6	Sensitive Data Exposure (Έκθεση Ευαίσθητων Δεδομένων)	A7	Insecure Cryptographic Storage – Merged with A9 (Μη Ασφαλής Κρυπτογραφημένη Αποθήκευση – Συγχωνεύτηκε με το A9)
A7	Missing Function Level Access Control (Έλλειψη στη Λειτουργία Ελέγχου στο Επίπεδο Πρόσβασης)	A8	Failure to Restrict URL Access (Αποτυχία Περιορισμού της Πρόσβασης URL)
A8	Cross Site Request Forgery (CSRF) (Πλαστογράφηση Αίτησης Μεταξύ Θέσεων)	A5	Cross Site Request Forgery (CSRF) (Πλαστογράφηση Αίτησης Μεταξύ Θέσεων)
A9	Using Known Vulnerable Components (Χρήστη στοιχείων που έχουν ευπάθειες)		(Burried in A6 - Security Misconfiguration)
A10	Unvalidated Redirects and Forwards (Μη Επικύρωση Ανακατευθύνσεων και Προωθήσεων)	A10	Unvalidated Redirects and Forwards (Μη Επικύρωση Ανακατευθύνσεων και Προωθήσεων)
	(Merged with 2010 – A7 into 2013 – A6) (Συγχωνεύθηκε με το A7 του 2010 και αποτελούν το A6 του 2013)	A9	Insufficient Transport Layer Protection (Ανεπαρκής Προστασία Επιπέδου Μεταφοράς)

Πίνακας 6. Κίνδυνοι Ασφάλειας των Διαδικτυακών Εφαρμογών με βάση την έκδοση 2013 του OWASP και σύγκριση με την έκδοση 2010

2.3 OWASP Top10 - 2013

Η τελευταία δημοσίευση του OWASP όπως παρουσιάστηκε στον Πίνακα 6, είναι ενημερωμένη με τους σημαντικότερους κινδύνους που σχετίζονται με τις διαδικτυακές εφαρμογές έως και σήμερα και ορίζονται με σειρά προτεραιότητας ως ακολούθως:

2.3.1 A1 – Injection (Έγχυση)

Επιθέσεις ένεσης κώδικα, όπως οι επιθέσεις ένεσης SQL, LDAP κ.α., συμβαίνουν όταν μη έγκυρα δεδομένα στέλνονται στον διερμηνευτή (interpreter) και εκτελούνται ως έγκυρα. Τα δεδομένα αυτά, μπορούν να ξεγελάσουν τον διερμηνευτή στο να εκτελέσει ακούσια εντολές ή να επιτρέψει σε έναν κακόβουλο χρήστη να αποκτήσει πρόσβαση σε εμπιστευτικά δεδομένα.

2.3.2 A2 – Broken Authentication and Session Management (Εσφαλμένη Διαχείριση Λογαριασμών και Συνόδων)

Λειτουργίες εφαρμογών που σχετίζονται με αυθεντικοποίηση και διαχείριση συνόδων συχνά δεν υλοποιούνται σωστά επιτρέποντας σε επιτιθέμενους να υποκλέψουν συνθηματικά, κλειδιά, στοιχεία συνόδων, ή να εκμεταλλευτούν άλλες αδυναμίες για να υιοθετήσουν ταυτότητες άλλων χρηστών.

2.3.3 A3 – Cross-Site Scripting (XSS) (Ατέλειες Τύπου XSS)

Οι επιθέσεις XSS προκαλούνται όταν μια εφαρμογή δέχεται μη έγκυρα δεδομένα και τα στέλνει σε έναν φυλλομετρητή χωρίς να τα ελέγξει και να τα επικυρώσει. Ένας κακόβουλος χρήστης μπορεί να εκτελέσει κώδικα σε έναν φυλλομετρητή και έτσι να ανακατευθύνει ανύποπτους χρήστες σε κακόβουλες ιστοσελίδες και να καταστρέψει ιστοσελίδες [17].

2.3.4 A4 – Insecure Direct Object References (Μη Ασφαλής Απευθείας Αναφορά σε Αντικείμενα)

Μη ασφαλής απευθείας αναφορά αντικειμένου υπάρχει όταν ένας προγραμματιστής εκθέτει μια αναφορά ενός εσωτερικού αντικειμένου όπως ένα αρχείο, ένα κατάλογο, ή το κλειδί μιας βάσης δεδομένων. Χωρίς τον κατάλληλο έλεγχο πρόσβασης και την κατάλληλη προστασία, οι επιτιθέμενοι μπορούν να διαχειρίζονται αυτές τις αναφορές αποκτώντας πρόσβαση σε μη εξουσιοδοτημένα δεδομένα.

2.3.5 A5 – Security Misconfiguration (Σφάλματα Διαμόρφωσης Ασφαλείας)

Οι ορθές πρακτικές ασφάλειας επιτάσσουν σωστές ρυθμίσεις σε εφαρμογές, βάσεις δεδομένων και εξυπηρετητές. Αυτές πρέπει να καθοριστούν, να υλοποιηθούν και να διατηρηθούν αφού δεν διατίθενται στην αγορά ως προεπιλογές. Έτσι απαιτείται η ενημέρωση του ήδη υπάρχοντος λογισμικού όπως και όλων των βιβλιοθηκών που χρησιμοποιούνται από μια εφαρμογή.

2.3.6 A6 – Sensitive Data Exposure (Έκθεση Ευαίσθητων Δεδομένων)

Ένας μεγάλος αριθμός διαδικτυακών εφαρμογών δεν προστατεύει κατάλληλα τα ευαίσθητα δεδομένα όπως είναι για παράδειγμα τις πιστωτικές κάρτες ή τα στοιχεία αυθεντικοποίησης (authentication

credentials). Οι επιτιθέμενοι μπορούν να υποκλέψουν ή να τροποποιήσουν τέτοια δεδομένα που δεν προστατεύονται με τον κατάλληλο τρόπο για να διενεργήσουν απάτες με πιστωτικές κάρτες ή άλλες παράνομες πράξεις. Αυτά τα ευαίσθητα δεδομένα χρειάζονται επιπλέον προστασία όπως είναι η κρυπτογράφηση τόσο όταν είναι αποθηκευμένα ή όταν αποστέλλονται κάπου, καθώς και επιπλέον μέτρα ασφαλείας όταν ανταλλάσσονται μέσω του φυλλομετρητή.

2.3.7 A7 –Missing Function Level Access Control (Έλλειψη στη Λειτουργία Ελέγχου στο Επίπεδο Πρόσβασης)

Οι περισσότερες διαδικτυακές εφαρμογές επαληθεύουν τα δικαιώματα πρόσβασης πριν δώσουν μια συγκεκριμένη λειτουργία στην πλευρά του χρήστη. Παρ' όλα αυτά, οι εφαρμογές χρειάζεται να εκτελούν τους ίδιους έλεγχους πρόσβασης στην πλευρά του εξυπηρετητή (server). Αν δεν γίνεται έλεγχος πρόσβασης (access control) και στην πλευρά του εξυπηρετητή, τότε οι επιτιθέμενοι έχουν την δυνατότητα να δημιουργήσουν κατάλληλα request με σκοπό να αποκτήσουν πρόσβαση στην λειτουργία χωρίς να έχουν την κατάλληλη εξουσιοδότηση.

2.3.8 A8 – Cross-Site Request Forgery (CSRF) (Πλαστογράφηση Αίτησης Μεταξύ Θέσεων)

Μια επίθεση CSRF αναγκάζει τον φυλλομετρητή ενός θύματος να στείλει σε μια ευάλωτη εφαρμογή μια πλαστή αίτηση, συμπεριλαμβάνοντας το cookie της συνόδου καθώς και οποιαδήποτε άλλη πληροφορία. Ο εισβολέας μπορεί έτσι να αναγκάσει το φυλλομετρητή του θύματος να δημιουργήσει αιτήματα τα οποία η ευπαθής εφαρμογή θεωρεί ότι είναι έγκυρα.

2.3.9 A9 - Using Known Vulnerable Components (Χρήστη στοιχείων που έχουν ευπάθειες)

Βοηθητικά στοιχεία, όπως βιβλιοθήκες (libraries), πλαίσια (frameworks), και άλλα στοιχεία λογισμικού (software modules), των διαδικτυακών εφαρμογών σχεδόν πάντα εκτελούνται με δικαιώματα πλήρους πρόσβασης. Αν ένα τέτοιο στοιχείο που μπορεί να έχει κάποια ευπάθεια το εκμεταλλευτεί κάποιος, τότε μπορεί να προκληθεί μια επίθεση η οποία ίσως διευκολύνει μια σοβαρή απώλεια δεδομένων ή την απόκτηση ελέγχου του εξυπηρετητή. Οι εφαρμογές που χρησιμοποιούν βοηθητικά στοιχεία με γνωστές ευπάθειες μπορεί να υπονομεύσουν την ασφάλεια της εφαρμογής και να ενεργοποιήσουν μια σειρά πιθανών επιθέσεων.

2.3.10 A10 - Unvalidated Redirects and Forwards (Μη Επικύρωση Ανακατευθύνσεων και Προωθήσεων)

Οι διαδικτυακές εφαρμογές συχνά ανακατευθύνουν τους χρήστες σε άλλες σελίδες χρησιμοποιώντας μη έγκυρα δεδομένα. Χωρίς τον κατάλληλο έλεγχο ένας επιτιθέμενος μπορεί να ανακατευθύνει τα θύματα του σε ιστοσελίδες phishing ή σε ιστοσελίδες που περιέχουν κακόβουλο λογισμικό.

2.4 Τρόποι αποτροπής των κινδύνων ασφαλείας σύμφωνα με τον OWASP

Αντίστοιχα, η τελευταία δημοσίευση του OWASP παραθέτει και τις ενέργειες στις οποίες πρέπει να προχωρήσουμε προκειμένου να αποτρέψουμε τους συγκεκριμένους κινδύνους και να καταστήσουμε τις εφαρμογές μας πιο ασφαλείς για χρήση.

2.4.1 Τρόποι αποτροπής από A1 - Injection (Έγχυση)

Η αποτροπή της συγκεκριμένης ευπάθειας απαιτεί τον διαχωρισμό των μη έμπιστων δεδομένων από τις εντολές και τις επερωτήσεις (queries).

1. Η προτεινόμενη επιλογή είναι η χρήση μιας ασφαλούς προγραμματιστικής διεπαφής API η οποία αποφεύγει τελείως τη χρήση του διερμηνευτή (interpreter) και παρέχει μια παραμετροποιημένη διεπαφή (interface). Εδώ πρέπει να δοθεί προσοχή σε API's τύπου «αποθηκευμένες διαδικασίες» τα οποία είναι μεν παραμετροποιημένα αλλά μπορούν να παρουσιάσουν και αυτά ευπάθεια τύπου έγχυσης (injection), με όχι ορατό τρόπο. Η μη κατάλληλη χρήση των αποθηκευμένων διαδικασιών, όπως με την εισαγωγή δυναμικού SQL μέσα σε αυτές, τις καθιστούν τόσο επιβλαβείς όπως το δυναμικό SQL σε μια ιστοσελίδα. Όταν χρησιμοποιείται δυναμικό SQL μέσα σε αποθηκευμένες διαδικασίες θα πρέπει η εφαρμογή να έχει την δυνατότητα να ελέγχει τις εκχωρήσεις του χρήστη να ώστε να μειώνει τον κίνδυνο έγχυσης κώδικα. Αν δεν έχει αυτή την δυνατότητα τότε ο χρήστης μπορεί να εισάγει κακόβουλο SQL το οποίο θα εκτελεστεί με την αποθηκευμένη διαδικασία. [18]
2. Αν δεν είναι διαθέσιμο ένα παραμετροποιημένο API, θα πρέπει προσεκτικά να αναιρεθεί η ειδική σημασία των ειδικών χαρακτήρων με την χρήση του ειδικού συντακτικού διαφυγής (escaping) για τον διερμηνευτή (interpreter).
3. Προτείνεται επίσης η θετική ή διαφορετικά «λευκή λίστα» (white-list) επικύρωσης εισόδου, π.χ. με χρήση κανονικών εκφράσεων – με άλλα λόγια η θεώρηση εισόδων που περιέχουν ειδικούς χαρακτήρες ως μη παραδεκτές. Η μέθοδος αυτή ωστόσο, δεν θεωρείται πλήρης άμυνα, καθώς αρκετές εφαρμογές μπορεί να απαιτούν τη δυνατότητα χρήσης ειδικών χαρακτήρων στην είσοδό τους. Το ESAPI του OWASP παρέχει μια εκτεταμένη βιβλιοθήκη με αλγόριθμους επικύρωσης εισόδου λευκής λίστας.

2.4.2 Τρόποι αποτροπής από A2 - Broken Authentication and Session Management (Εσφαλμένη Διαχείριση Λογαριασμών και Συνόδων)

Η κυριότερη σύσταση για την αποτροπή αυτής της ευπάθειας είναι η χρήση ισχυρών εργαλείων ελέγχου αυθεντικοποίησης της συνόδου. Αυτά τα εργαλεία ελέγχου θα πρέπει να:

1. Ικανοποιούν όλες τις απαιτήσεις πιστοποίησης και διαχείρισης συνόδου που ορίζονται από το Application Security Verification Standard (ASVS) του OWASP.
2. Να έχουν μια απλή διεπαφή για τους κατασκευαστές λογισμικού.

Μεγάλες προσπάθειες επίσης πρέπει να γίνουν για να αποφευχθούν ελαττώματα του XSS τα οποία μπορούν να οδηγήσουν σε κλοπή των αναγνωριστικών συνεδρίας (sessions id).

2.4.3 Τρόποι αποτροπής από A3 - Cross-Site Scripting (XSS) (Ατέλειες Τύπου XSS)

Η αποτροπή της ευπάθειας XSS προϋποθέτει τον διαχωρισμό των μη έμπιστων δεδομένων από το ενεργό περιεχόμενο της εφαρμογής πλοήγησης (browser).

1. Η επιλογή που είναι προτιμότερη είναι να γίνει κατάλληλη χρήση διαφυγής (escaping) όλων των μη έμπιστων δεδομένων τα οποία βασίζονται στο πλαίσιο του προτύπου HTML (σώμα, ιδιότητες, Javascript, CSS ή URL) όπου τα δεδομένα μπορούν να τοποθετηθούν σε αυτό.

2. Η θετική ή λευκή λίστα (whitelist) επικύρωσης εισόδου προτείνεται επίσης, καθώς βοηθάει στην προστασία από την ευπάθεια XSS, αλλά αυτή δεν αποτελεί ολοκληρωμένη προστασία, καθώς αρκετές εφαρμογές μπορεί να απαιτούν τη δυνατότητα χρήσης ειδικών χαρακτήρων στην είσοδό τους. Μια τέτοια επικύρωση πρέπει να επικυρώνει το μήκος, τους χαρακτήρες και τη μορφή των δεδομένων εισόδου, προτού να κάνει αποδεκτή την είσοδο.
3. Για τη δυνατότητα χρήσης πλούσιου περιεχομένου, υπάρχουν οι βιβλιοθήκες Antisamy του OWASP ή η Java HTML Sanitizer Project.
4. Για την προστασία όλου του site από την ευπάθεια XSS υπάρχει το Content Security Policy (CSP).

2.4.4 Τρόποι αποτροπής από A4 - Insecure Direct Object References (Μη Ασφαλής Απευθείας Αναφορά σε Αντικείμενα)

Η αποτροπή της επισφαλούς άμεσης αναφοράς αντικειμένου ενός συστήματος απαιτεί την επιλογή μιας προσέγγισης για την προστασία κάθε αντικειμένου που είναι προσβάσιμο από τον χρήστη (π.χ. αριθμός, αντικείμενο, αρχείο):

1. Χρήση μόνο εμμέσων αναφορών αντικειμένου, με διαφορετικές τιμές ανά χρήστη ή ανά σύνοδο. Με τον τρόπο αυτό αποτρέπεται ο επιτιθέμενος να στοχεύει άμεσα σε πόρους στους οποίους δεν έχει εξουσιοδότηση. Για παράδειγμα, σε μια πτυσσόμενη λίστα (drop-down) με έξι αντικείμενα, αντί της χρησιμοποίησης του κλειδιού της βάσεως δεδομένων ως τιμή της επιλογής, είναι προτιμότερο είναι να χρησιμοποιούνται οι αριθμοί από το 1 έως το 6. Στην συνέχεια η εφαρμογή θα πρέπει να αντιστοιχήσει την τιμή που επέλεξε ο χρήστης με την πραγματική τιμή του κλειδιού της βάσης δεδομένων προκειμένου να εκτελεστούν οι υπόλοιπες διαδικασίες.
2. Έλεγχος πρόσβασης. Κάθε χρήση της άμεσης αναφοράς αντικειμένου από μια μη έμπιστη πηγή θα πρέπει να περιλαμβάνει έναν έλεγχο πρόσβασης για να διασφαλίσει ότι ο χρήστης είναι εξουσιοδοτημένος για το αντικείμενο που ζήτησε.

2.4.5 Τρόποι αποτροπής από A5 - Security Misconfiguration (Σφάλματα Διαμόρφωσης Ασφαλείας)

Οι κύριες συστάσεις για την αποτροπή της συγκεκριμένης ευπάθειας είναι να καθιερωθούν τα παρακάτω:

1. Μια επαναλαμβανόμενη διαδικασία ενίσχυσης της ασφάλειας που θα καθιστά γρήγορη και εύκολη την ανάπτυξη άλλου περιβάλλοντος το οποίο θα είναι κατάλληλα διασφαλισμένο. Το περιβάλλον ανάπτυξης, διασφάλισης ποιότητας και παραγωγής θα πρέπει να είναι συντονισμένα μεταξύ τους (με διαφορετικούς κωδικούς (passwords) το κάθε περιβάλλον). Η διαδικασία αυτή θα πρέπει να είναι αυτοματοποιημένη έτσι ώστε να ελαχιστοποιήσει την προσπάθεια που απαιτείται προκειμένου να εγκατασταθεί ένα νέο ασφαλές περιβάλλον.
2. Μια διαδικασία για την ενημέρωση και την εγκατάσταση των ενημερώσεων ασφαλείας και των επιδιορθώσεων του λογισμικού, η οποία θα πραγματοποιείται έγκαιρα σε όλα τα λειτουργούντα περιβάλλοντα. Η διαδικασία αυτή θα πρέπει να περιλαμβάνει οπωσδήποτε όλες τις βιβλιοθήκες κώδικα, οι οποίες συχνά παραλείπονται.
3. Μια ισχυρή αρχιτεκτονική της εφαρμογής η οποία παρέχει ασφαλή ασφάλεια ανάμεσα στα στοιχεία της.

4. Μια διαδικασία περιοδικής εκτέλεσης σαρώσεων και ελέγχων ασφάλειας βοηθά στον εντοπισμό των επισφαλών ρυθμίσεων ασφαλείας ή στον εντοπισμό μη εγκατεστημένων επιδιορθώσεων ασφαλείας.

2.4.6 Τρόποι αποτροπής από A6 - Sensitive Data Exposure (Έκθεση Ευαίσθητων Δεδομένων)

Για όλα τα ευαίσθητα δεδομένα χρειάζεται κατ' ελάχιστον να γίνουν τα παρακάτω:

1. Λαμβάνοντας υπόψη όλες τις απειλές από τις οποίες πρέπει να προστατευτούν τα δεδομένα (π.χ. εσωτερική επίθεση, εξωτερικός χρήστης), να εξασφαλιστεί ότι κρυπτογραφούνται όλα τα ευαίσθητα δεδομένα όταν είναι κάπου αποθηκευμένα όσο και όταν ανταλλάσσονται, με ένα τρόπο που να τα προστατεύει από αυτές τις απειλές.
2. Να μην αποθηκεύονται ευαίσθητα δεδομένα όταν δεν είναι απαραίτητο.
3. Να διασφαλιστεί ότι χρησιμοποιούνται ισχυροί αλγόριθμοι και δυνατά κλειδιά καθώς και ότι υπάρχει σωστή διαχείριση των κλειδιών.
4. Να διασφαλιστεί ότι οι κωδικοί πρόσβασης (passwords) είναι αποθηκευμένοι με την χρήση ενός αλγόριθμου που είναι σχεδιασμένος για την προστασία κωδικών πρόσβασης, όπως είναι ο bcrypt, PBKDF2, ή ο scrypt.
5. Να απενεργοποιηθεί η αυτόματη συμπλήρωση σε φόρμες που συγκεντρώνουν ευαίσθητα δεδομένα και να απενεργοποιηθεί το caching για σελίδες που περιέχουν ευαίσθητα δεδομένα.

2.4.7 Τρόποι αποτροπής από A7 - Missing Function Level Access Control (Έλλειψη στη Λειτουργία Ελέγχου στο Επίπεδο Πρόσβασης)

Η εφαρμογή θα πρέπει να έχει ένα συνεπές και εύκολο για ανάλυση στοιχείο εξουσιοδότησης που καλείται για χρήση από όλες τις επαγγελματικές λειτουργίες. Συχνά τέτοιου είδους προστασία παρέχεται από ένα ή περισσότερα εξωτερικά στοιχεία (components) σε σχέση με τον κώδικα της εφαρμογής.

1. Σκεφτείτε τη διαδικασία διαχείρισης των δικαιωμάτων και διασφαλίστε ότι μπορείτε να ανανεώσετε και να ελέγξετε εύκολα.
2. Οι μηχανισμοί ελέγχου πρόσβασης πρέπει να απαγορεύουν την πρόσβαση σε όλους εξ ορισμού, απαιτώντας ρητές εξουσιοδοτήσεις σε συγκεκριμένους χρήστες για την πρόσβαση σε κάθε λειτουργία.
3. Αν η λειτουργία εμπλέκεται σε μια ροη εργασίας (workflow), ελέγξτε για να βεβαιωθείτε ότι οι συνθήκες είναι σε κατάσταση τέτοια ώστε να επιτρέψουν την πρόσβαση.

2.4.8 Τρόποι αποτροπής από A8 - Cross-Site Request Forgery (CSRF) (Πλαστογράφηση Αίτησης Μεταξύ Θέσεων)

Η αποτροπή του CSRF απαιτεί συνήθως να συμπεριλάβουμε ένα μη προβλέψιμο διακριτικό (Token) στο σώμα κάθε HTTP αίτησης. Τέτοια διακριτικά πρέπει να είναι, τουλάχιστον μοναδικά ανά συνεδρία χρήστη.

1. Η προτιμώμενη επιλογή είναι να συμπεριλάβουμε το μοναδικό διακριτικό σε ένα κρυφό πεδίο. Αυτό έχει ως συνέπεια, η τιμή να στέλνεται στο σώμα του αιτήματος HTTP, αποφεύγοντας την ενσωμάτωση του στο URL το οποίο εκτίθεται.
2. Το μοναδικό διακριτικό μπορεί επίσης να συμπεριληφθεί στο ίδιο το URL ή σε μια παράμετρο αυτού. Μια τέτοια τοποθέτηση όμως, διατρέχει τον κίνδυνο της έκθεσης του URL στον επιτιθέμενο, με αποτέλεσμα να τίθεται σε κίνδυνο το μυστικό διακριτικό.
3. Το CSRF guard του OWASP μπορεί να συμπεριλάβει αυτόματα τέτοια διακριτικά σε εφαρμογές όπως java, .NET ή PHP. Επίσης το ESAPI του OWASP περιλαμβάνει μεθόδους τις οποίες μπορούν να χρησιμοποιούν οι προγραμματιστές για να αποτρέπουν τις ευπάθειες CSRF.
4. Η απαίτηση από τον χρήστη να επαναυθενικοποιείται (reauthenticate) ή το να αποδεικνύει το ότι είναι πραγματικό πρόσωπο (π.χ. με την χρήση του CAPTCHA) μπορούν επίσης να προστατεύσουν από ευπάθειες CSRF.

2.4.9 Τρόποι αποτροπής από A9 - Using Known Vulnerable Components (Χρήστη στοιχείων που έχουν ευπάθειες)

Μια λύση είναι να μην χρησιμοποιούνται εξωτερικά στοιχεία λογισμικού (components). Αλλά αυτό δεν είναι και τόσο ρεαλιστικό. Τα περισσότερα σύνολα στοιχείων λογισμικού (component projects) δεν επιδιορθώνουν τις ευπάθειες για τις παλιές εκδόσεις. Αντί αυτού επιδιορθώνουν το πρόβλημα στην επόμενη έκδοση. Συνεπώς η αναβάθμιση σε αυτές τις νέες εκδόσεις είναι κρίσιμη. Τα προγράμματα λογισμικού θα πρέπει να έχουν μια διαδικασία με σκοπό:

1. Να αναγνωρίζουν όλα τα στοιχεία λογισμικού και τις εκδόσεις που χρησιμοποιούνται συμπεριλαμβανομένων όλων των σχετικών (π.χ. τις εκδόσεις plug in).
2. Να παρακολουθούν την ασφάλεια αυτών των στοιχείων σε ελεύθερα προσπελάσιμες βάσεις δεδομένων, σε λίστες email και λίστες email ασφαλείας και να τις κρατάνε ενήμερες.
3. Να καθιερώνουν πολιτικές ασφάλειας που διέπουν την χρήση των στοιχείων (component), όπως με το να απαιτούν ορισμένες πρακτικές ανάπτυξης λογισμικού, να περνάνε επιτυχώς τους ελέγχους ασφαλείας και να έχουν αποδεκτές άδειες.
4. Όπου είναι κατάλληλο, καλό είναι να προστίθενται περιτυλίγματα ασφαλείας γύρω από τα στοιχεία (components) για να απενεργοποιούν μη χρήσιμες λειτουργικότητες και/ή να ασφαλίζουν αδύναμα ή ευπαθή χαρακτηριστικά των στοιχείων (component).

2.4.10 Τρόποι αποτροπής από A10 - Unvalidated Redirects and Forwards (Μη Επικύρωση Ανακατευθύνσεων και Προωθήσεων)

Ασφαλής χρήση των ανακατευθύνσεων ή των προωθήσεων σε μια εφαρμογή μπορεί να πραγματοποιηθεί με τους παρακάτω τρόπους:

1. Απλά, να σταματήσει η χρήση αυτών των μεθόδων.
2. Αν όμως, δεν μπορεί να σταματήσει η χρήση τους, τότε καλό είναι να μην γίνεται η χρήση των παραμέτρων των χρηστών στον υπολογισμό της σελίδας προορισμού.
3. Αν δεν μπορεί να αποφευχθεί η εισαγωγή των παραμέτρων των χρηστών, πρέπει να εξασφαλιστεί ότι οι τιμές που δίνουν οι χρήστες στο σύστημα είναι έγκυρες και εξουσιοδοτημένες για τον χρήστη. Συνίσταται δε, οι τιμές των παραμέτρων προορισμού να μην είναι το ίδιο το URL αλλά ένας κωδικός, ο οποίος μέσω απεικόνισης στην πλευρά του εξυπερετητή να οδηγεί στο URL στόχο. Οι εφαρμογές μπορούν να χρησιμοποιήσουν το

ESAPI για να παρακάμψουν τη μέθοδο sendRedirect() και να εξασφαλίσουν ότι όλοι οι προορισμοί ανακατευθύνσεων είναι ασφαλείς.

Η αποφυγή τέτοιων λαθών είναι εξαιρετικά σημαντική δεδομένου ότι είναι ένας από τους αγαπημένους στόχους των επιτιθέμενων τέτοιου είδους (phishers), που προσπαθούν να κερδίσουν την εμπιστοσύνη των χρηστών.

Κεφάλαιο 3ο

3 Περιγραφή του Εργαλείου Εντοπισμού Ευπαθειών Nessus

3.1 Εισαγωγή

Ο εντοπισμός ευπαθειών και η παρακολούθηση της αποτελεσματικότητας των μέτρων προστασίας ενός πληροφοριακού συστήματος αποτελούν απαραίτητη προϋπόθεση τόσο κατά την εφαρμογή όσο και κατά τη συντήρηση ενός σχεδίου ασφάλειας. Αποτέλεσμα αυτής της ανάγκης ήταν να αναπτυχθούν εργαλεία εντοπισμού και ανάλυσης ευπαθειών.

Η χρήση τους δίνει την δυνατότητα ελέγχου του επίπεδου ασφάλειας υπολογιστικών συστημάτων, δικτύων και εφαρμογών εντοπίζοντας τα αδύνατα σημεία τους που θα εκμεταλλευτούν οι πιθανοί εισβολείς. Στη συνέχεια μπορούν να αναπτυχθούν όλα τα απαραίτητα αντίμετρα (countermeasures) ώστε να θωρακιστεί ο ελεγχόμενος στόχος από τα κενά ασφαλείας που εντοπίστηκαν και με την επανάληψη αυτής της διαδικασίας σε τακτά χρονικά διαστήματα, υιοθετείται και εφαρμόζεται ένα πλάνο που μειώνει τις πιθανότητες έκθεσης σε κινδύνους.

Σήμερα υπάρχουν διαθέσιμα πολλά εργαλεία εντοπισμού ευπαθειών και υπάρχει δυνατότητα επιλογής εξ αυτών ανάλογα με τις απαιτήσεις του ελέγχου ασφαλείας. Ενδεικτικά αναφέρονται τα παρακάτω που είναι τα πιο δημοφιλή από τους χρήστες και χαρακτηρίζονται ως εργαλεία τελευταίας τεχνολογίας (state-of-the-art):

- Open Vas [19]
- Burp Suite [20]
- Retina [21]
- Nexpose [22]
- MBSA – Microsoft Baselive Security Analyser [23]
- Core Impact [24]
- Nessus [25]
- Nmap [26]
- QualysGuard [27]
- GFI LanGuard [28]

Από τα διαθέσιμα εργαλεία επιλέχτηκε το Nessus ως το καλύτερο για την κάλυψη των αναγκών της παρούσας εργασίας. Το οποίο είναι ένα προϊόν της Tenable Network Security και είναι ένα εργαλείο εντοπισμού ευπαθειών αρκετά «δυνατό» και εύκολο στην χρήση. Έχει ένα μεγάλο αριθμό plugins τα οποία ανανεώνονται σε καθημερινή βάση και τώρα κατατάσσεται ανάμεσα στα κορυφαία προϊόντα της κατηγορίας του σε όλη την βιομηχανία της ασφάλειας των υπολογιστών και εγκρίθηκε από επαγγελματικούς οργανισμούς ασφάλειας πληροφοριών όπως το ινστιτούτο SANS [29]. Παρέχει την δυνατότητα ελέγχου εξ αποστάσεως ενός δικτύου και μπορεί να διαπιστώσει αν αυτό το δίκτυο έχει εκτεθεί σε κίνδυνο ή έχει παραβιαστεί κατά κάποιο τρόπο. Επίσης το Nessus δίνει την δυνατότητα τοπικού ελέγχου ενός συστήματος για ευπάθειες, για την τήρηση προδιαγραφών, για καταστρατήγηση της ακολουθούμενης πολιτικής κ.α.

3.2 Παρουσίαση του Nessus

Η έκδοση του Nessus που θα παρουσιαστεί σε αυτό το κεφάλαιο (Nessus 5.2) είναι η πιο πρόσφατη. Η ανανεωμένη αυτή έκδοση είναι διαθέσιμη προς χρήση από τις 12 Ιουνίου 2014. Από την έκδοση 5 και μετά, η διαχείριση των χρηστών και η παραμετροποίηση του server του Nessus (daemon) γίνεται μέσω της διατιθέμενης διεπαφής χρήστη (Graphic User Interface).

Το GUI του Nessus είναι μια web διεπαφή που δίνει την δυνατότητα για τις απαραίτητες ρυθμίσεις, για την δημιουργία νέων πολιτικών, για τον χειρισμό των ελέγχων και των αναφορών τους. Θα πρέπει να σημειωθεί ότι το GUI είναι ίδιο και δεν επηρεάζεται από το λογισμικό οπότε οι λειτουργίες του που θα παρουσιαστούν εδώ θα ισχύουν για όλα τα υποστηριζόμενα λειτουργικά συστήματα. Το Nessus είναι διαθέσιμο και υποστηρίζεται για τα παρακάτω λειτουργικά συστήματα και πλατφόρμες:

- Debian 6 and 7 (i386 and x86-64)
- Fedora 19 and 20 (i386 and x86-64)
- FreeBSD 9 (i386 and x86-64)
- Mac OS X 10.8 and 10.9 (i386 and x86-64)
- Red Hat ES 4 / CentOS 4 (i386)
- Red Hat ES 5 / CentOS 5 / Oracle Linux 5 (i386 and x86-64)
- Red Hat ES 6 / CentOS 6 / Oracle Linux 6 (i386 and x86-64) [Server, Desktop, Workstation]
- SuSE 10 (x86-64), 11 (i386 and x86-64)
- Ubuntu 10.04 (9.10 package), 11.10, 12.04, and 12.10 (i386 and x86-64)
- Windows XP, Server 2003, Server 2008, Server 2008 R2*, Server 2012, Vista, 7, and 8 (i386 and x86-64)

3.3 Δομή και βασικές λειτουργίες του Nessus

Σύμφωνα με την Tenable Network Security [30] η δομή και οι βασικές λειτουργίες του Nessus παρουσιάζονται παρακάτω.

3.3.1 Intelligent Scanning (Έξυπνη Σάρωση)

Σε αντίθεση με άλλα αντίστοιχα εργαλεία, το Nessus δεν θεωρεί τίποτα δεδομένο (π.χ ότι μια συγκεκριμένη υπηρεσία τρέχει σε μια συγκεκριμένη θύρα). Οπότε αν κάποιος τρέχει τον web server του στην θύρα 1234, το Nessus θα το ανιχνεύσει παρόλο που δεν είναι κλασσική θύρα για τον web server και θα κάνει τους ελέγχους ασφαλείας με τον σωστό τρόπο. Επίσης θα προσπαθήσει να επαληθεύσει μια ευπάθεια εκμεταλλευόμενο ένα exploit όταν είναι πιθανό. Σε περιπτώσεις όπου δεν θα είναι αξιόπιστος ο έλεγχος ή μπορεί επηρεάσει αρνητικά τον ελεγχόμενο στόχο τότε το Nessus θα βασιστεί στις πληροφορίες ενός server banner για να καθορίσει την ύπαρξη της ευπάθειας. Σε κάθε περίπτωση είναι ξεκάθαρο στην εξαγόμενη αναφορά ποια μέθοδος ακολουθήθηκε.

3.3.2 Modular Architecture (Αρθρωτή Αρχιτεκτονική)

Η αρχιτεκτονική client/server (πελάτη/εξυπηρετητή) παρέχει την ευελιξία εγκατάστασης του εργαλείου εντοπισμού (server) και η σύνδεση στη διεπαφή (client) από οποιαδήποτε μηχανή έχει ένα web browser, μειώνοντας τα κόστη διαχείρισης.

3.3.3 CVE Compatible (Συμβατό με CVE)

Τα περισσότερα plugins παρέχουν συνδέσεις CVE στους διαχειριστές ώστε να μπορούν να ανακτήσουν περισσότερες πληροφορίες σε σχέση με τις ευπάθειες που έχουν δημοσιευτεί. Επίσης συχνά περιέχουν αναφορές σε Bugtraq (BID), OSVDB, και προειδοποιήσεις ασφαλείας διαφόρων κατασκευαστών.

3.3.4 Plugin Architecture (Επεκτάσιμη Αρχιτεκτονική)

Κάθε έλεγχος ασφαλείας γράφεται σαν ένα εξωτερικό plugin και στη συνέχεια ομαδοποιείται σε μια από τις 42 οικογένειες. Με αυτό τον τρόπο μπορούν εύκολα να προστεθούν οι ειδικά ανά περίπτωση διαμορφωμένοι έλεγχοι ασφαλείας ,να επιλεγούν συγκεκριμένα plugins, ή μια ολόκληρη οικογένεια από plugins χωρίς να είναι απαραίτητο η γνώση του κώδικα του Nessus server engine, του nessusd. Όλη η λίστα των plugins του Nessus είναι διαθέσιμη στην διεύθυνση http://www.nessus.org/plugins/index.php?view=all.

3.3.5 NASL (Γλώσσα Συγγραφής Scripts για το Nessus)

Το Nessus συμπεριλαμβάνει την NASL (Nessus Attack Scripting Language), μια γλώσσα που σχεδιάστηκε ειδικά για να γράφει ελέγχους ασφαλείας εύκολα και γρήγορα. Οι έλεγχοι ασφαλείας αυτοί χωρίζονται στη συνέχεια σε διαφορετικές ομάδες, στις οικογένειες των plugins. Τα τελευταία χρόνια έχουν γραφτεί πάρα πολλοί έλεγχοι ασφαλείας με την συγκεκριμένη γλώσσα και μπορεί να γίνει η χρήση τους απλά με την εγκατάσταση των αντίστοιχων plugins. Η χρήση της NASL για δημιουργία ελέγχου ασφαλείας για μια ευπάθεια βοηθά στην καλύτερη κατανόηση της ευπάθειας. Επίσης αποτελεί πολύτιμη βοήθεια για τους υπόλοιπους χρήστες αφού θα έχουν έναν ακόμα έλεγχο προς χρήση.

Η τυπική μορφή ενός script [31] με την χρήση της NASL είναι όπως παρακάτω.

```
if(description) {

        script_id(12248);

        script_version ("$Revision: 1.5 $");

        script_name(english: "notes.ini checker");

        desc["english"] = " This plugin attempts to determine the existence of a directory traversal
bug on the remote Lotus Domino Web server.

         Risk factor : High";
```

```
        script_description(english:desc["english"]);

        script_summary (english: "notes.ini checker");

        script_category (ACT_ATTACK);

        script_family(english: "Misc");

        script_copyright(english:"This script is Copyright (C) 2014 Security");

        script_dependencie("http_version.nasl");

        script_require_ports("Services/www", 80);

        exit(0);

}
```

Παράδειγμα ενός script [32] για την ευπάθεια άρνησης εξυπηρέτησης στο Yahoo Messenger

```
if(description) {

        script_id(10326);

        script_version ("$Revision: 1.12 $");

        script_cve_id("CAN-2000-0047");

        name["english"] = "Yahoo Messenger Denial of Service attack";

        script_name(english:name["english"]);

        desc["english"] = " It is possible to cause Yahoo Messenger to crash by sending a few bytes of garbage into its listening port TCP 5010. Solution: Block those ports from outside communication Risk factor : Low";

        script_copyright(english:"This script is Copyright (C) 1999 SecuriTeam"); family["english"] = "Denial of Service";

        script_family(english:family["english"];

        exit(0);

}
```

```
# # Ο κώδικας του script ξεκινά εδώ #

if (get_port_state(5010)) {

        sock5010 = open_sock_tcp(5010);

        if (sock5010) {

                send(socket:sock5010, data:crap(2048));

                close(sock5010);

                sock5010_sec = open_sock_tcp(5010);

                if ( !sock5010_sec ) {

                        security_hole(5010);

                } else close(sock5010_sec);

        }

}
```

3.3.6 Up-to-date Security Vulnerability Database (Ενημερωμένη Βάση Δεδομένων Ευπαθειών)

Το Nessus εστιάζει στην ανάπτυξη ελέγχων ασφαλείας για τις πιο πρόσφατες ευπάθειες που έχουν παρουσιαστεί. Η βάση ελέγχων ασφαλείας του ενημερώνεται σε καθημερινή βάση και οι πιο πρόσφατοι έλεγχοι ασφαλείας είναι διαθέσιμοι στην διεύθυνση http://www.tenable.com/plugins/index.php?view=newest

3.3.7 Tests Multiple Hosts Simultaneously (Έλεγχος Πολλαπλών Στόχων Παράλληλα)

Ανάλογα νε την παραμετροποίηση του Nessus, μπορεί να γίνει έλεγχος σε ένα μεγάλο αριθμό στόχων ταυτόχρονα.

3.3.8 Smart Service Recognition (Έξυπνη Αναγνώριση Υπηρεσιών)

Το Nessus δεν θεωρεί δεδομένο ότι οι ελεγχόμενοι στόχοι θα σέβονται τους κανόνες της IANA και θα τρέχουν στις προκαθορισμένες θύρες. Αυτό σημαίνει ότι θα αναγνωρίζει έναν FTP server που δεν τρέχει στην καθορισμένη θύρα (π.χ. στη θύρα 31337) ή ένα web server που τρέχει στην θύρα 8080 αντί για την 80.

3.3.9 Multiple Services (Πολλαπλές Υπηρεσίες)

Αν δύο ή περισσότεροι web servers τρέχουν στον ελεγχόμενο στόχο (π.χ ο ένας στη θύρα 80 και ο άλλος στη θύρα 8080), το Nessus θα αναγνωρίσει και θα ελέγξει και τους δύο.

3.3.10 Plugin Cooperation (Συνεργασία των Plugin)

Οι έλεγχοι ασφαλείας που γίνονται από τα plugins του Nessus συνεργάζονται ώστε να αποφεύγονται οι μη απαραίτητοι έλεγχοι. Αν ένας ελεγχόμενος FTP server δεν παρέχει ανώνυμα login τότε οι σχετικοί έλεγχοι με τα ανώνυμα login δεν θα πραγματοποιηθούν.

3.3.11 Complete Reports (Πλήρης Αναφορές)

Το Nessus δεν θα αναφέρει μόνο τι ευπάθειες ασφαλείας υπάρχουν στο δίκτυο και το επίπεδο κινδύνου του καθενός (Info, Low, Medium, High, and Critical), αλλά θα προτείνει και λύσεις για την αντιμετώπισή τους.

3.3.12 Full SSL Support (Πλήρης Υποστήριξη SSL)

Το Nessus έχει την ικανότητα να ελέγχει υπηρεσίες που τρέχουν σύμφωνα με το πρωτόκολλο SSL όπως HTTPS, SMTPS, IMAPS και άλλες.

3.3.13 Smart Plugins (Έξυπνα Plugins - προαιρετικό)

Έχει την επιλογή «βελτιστοποίησης» η οποία ορίζει ποια plugins πρέπει ή δεν πρέπει να εκτελεστούν στον ελεγχόμενο στόχο. Για παράδειγμα το Nessus δεν θα ελέγξει ευπάθειες που αφορούν το sendmail όταν ανιχνεύσει το Postfix.

3.3.14 Non-Destructive (Μη Επιβλαβείς Έλεγχοι - προαιρετικό)

Κάποιοι έλεγχοι μπορεί να είναι επιβλαβείς για ορισμένες υπηρεσίες του δικτύου. Αν δεν είναι επιθυμητό να υπάρχει ο κίνδυνος να προκληθεί κάποια βλάβη στο δίκτυο υπάρχει η δυνατότητα ενεργοποίησης της επιλογής «ασφαλών ελέγχων», η οποία θα κάνει το Nessus να βασιστεί σε banners παρά σε πραγματικά προβλήματα του κώδικα για να καθορίσει την παρουσία μιας ευπάθειας.

3.4 Εγκατάσταση του Nessus

3.4.1 Απαιτήσεις Συστήματος

Οι απαιτήσεις συστήματος για την ομαλή λειτουργία του Nessus παρουσιάζονται στον παρακάτω πίνακα.

Scenario	Cpu/Memory	Disk Space
Nessus scanning smaller networks	CPU: 1 Pentium 4 dual-core 2 GHz CPU (dual-core Intel for Mac OS X) Memory: 2 GB RAM (4 GB RAM recommended)	30 GB

Nessus scanning large networks including audit trails and PDF report genaration	CPU: 1 Pentium 4 dual-core 3 GHz CPU (2 dual-core recommended) Memory: 3-4 GB RAM (8 GB RAM recommended)	30 GB

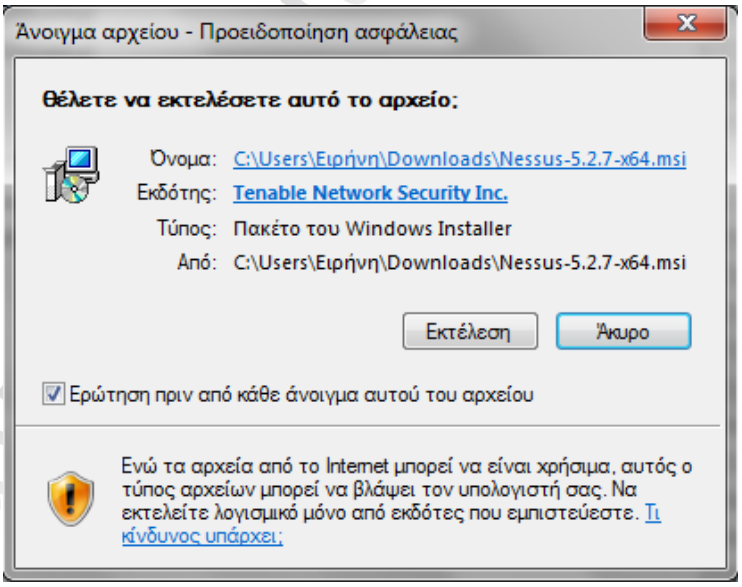
Πίνακας 7. Απαιτήσεις συστήματος Nessus

3.4.2 Εγκατάσταση

Στην ιστοσελίδα του Nessus http://www.tenable.com/products/nessus/select-your-operating-system μπορεί να γίνει επιλογή της έκδοσης ανάλογα με το λειτουργικό σύστημα που χρησιμοποιείται. Η εγκατάσταση του Nessus πρέπει να γίνει με την χρήση ενός λογαριασμού με δικαιώματα διαχειριστή και όχι με έναν απλό λογαριασμό χρήστη γιατί αλλιώς μπορεί να παρουσιαστούν προβλήματα κατά την εγκατάσταση (π.χ να εμφανιστούν λάθη που σχετίζονται με άδειες χρήσης (Access Denied)). Κάποια antivirus μπορεί να αντιληφθούν το Nessus σαν κάποιο ιό ή σαν κάποιο τύπο malware. Αυτό οφείλεται στις πολλές συνδέσεις TCP που δημιουργούνται κατά την διάρκεια μιας σάρωσης.

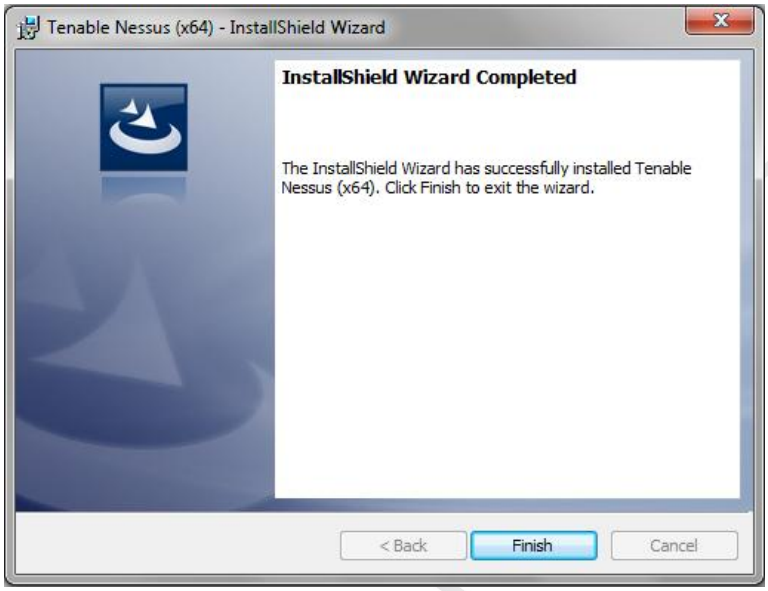
Πριν να ξεκινήσει η εγκατάσταση συνίσταται να έχει αποκτηθεί ένας κωδικός ενεργοποίησης για την τροφοδότηση με τα plugins, καθώς αυτή η πληροφορία θα απαιτηθεί προτού να υπάρχει δυνατότητα αυθεντικοποίησης στην διεπαφή του Nessus. Στην προκειμένη περίπτωση αποκτήθηκε ένας κωδικός ενεργοποίησης τύπου «Home Feed» που έχει περιορισμένα δικαιώματα χρήσης.

Η εγκατάσταση στην παρούσα εργασία αφορά Windows 7 Home Premium (64 bit), οπότε θα επιλεγεί το Nessus-5.2.7-x64.msi. Το αρχείο που θα κατέβει είναι ένα εκτελέσιμο αρχείο εγκατάστασης.



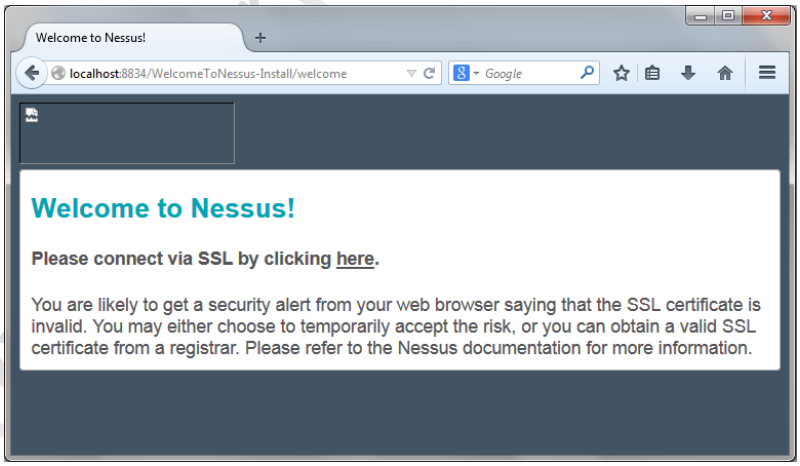
Εικόνα 3. Εκτελέσιμο αρχείο εγκατάστασης Nessus

Με την υλοποίηση των απαραιτήτων βημάτων πραγματοποιείται η εγκατάσταση του Nessus.



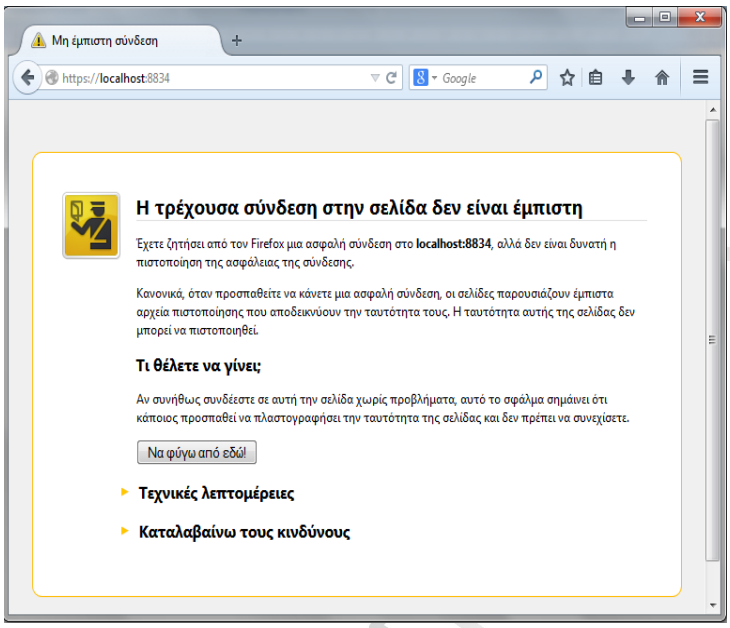
Εικόνα 4. Ολοκλήρωση εγκατάστασης του Nessus

Σε αυτό το σημείο το Nessus θα συνεχίσει με την φόρτωση του προκαθορισμένου web browser ώστε να συνεχίσετε μέσω της web διεπαφής του, την αρχική διαμόρφωση. Όπως φαίνεται στην εικόνα 5, ενημερώνει ότι η σύνδεση πλέον θα γίνεται μέσω του πρωτοκόλλου SSL.



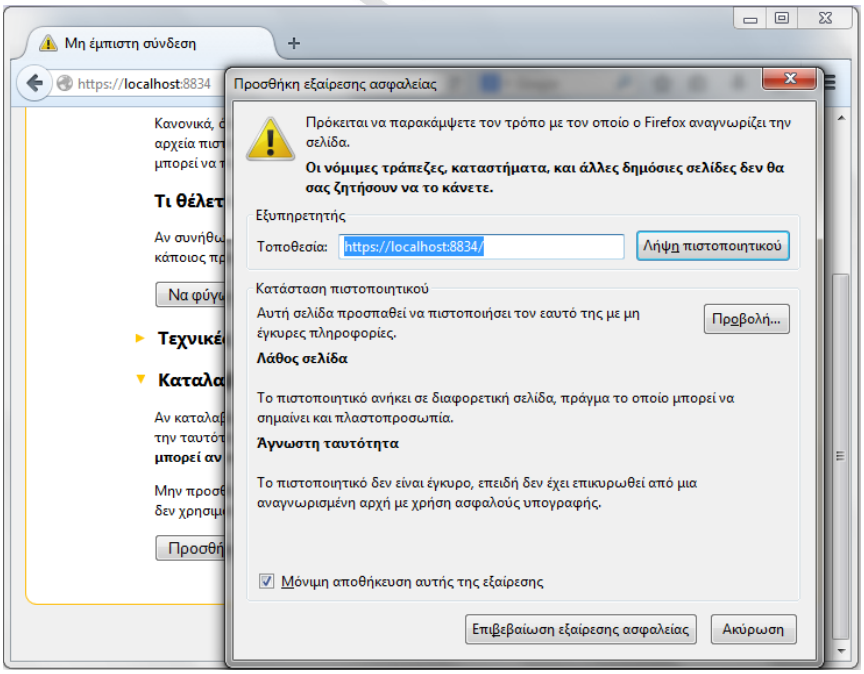
Εικόνα 5. Αρχική οθόνη της διεπαφής του Nessus

Την πρώτη φορά που θα γίνει η σύνδεση με τον web server του Nessus ο browser θα εμφανίσει ένα σφάλμα, προειδοποιώντας ότι η σύνδεση δεν είναι ασφαλής. Αυτό οφείλεται στη χρήση ενός αυτό-υπογεγραμμένου πιστοποιητικού SSL.



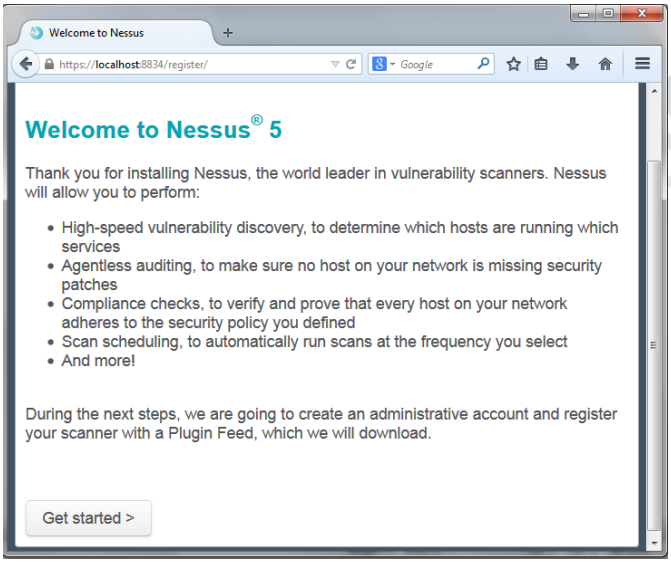
Εικόνα 6. Προειδοποίηση μη ασφαλούς σύνδεσης

Για την πρώτη φορά που γίνεται η σύνδεση πρέπει να γίνει επιβεβαίωση της εξαίρεσης ασφαλείας ώστε στη συνέχεια να γίνει παραμετροποίηση και μετέπειτα ανάλογα με τον browser που χρησιμοποιείται υπάρχει διαδικασία που ακολουθείται για την εγκατάσταση του κατάλληλου πιστοποιητικού.



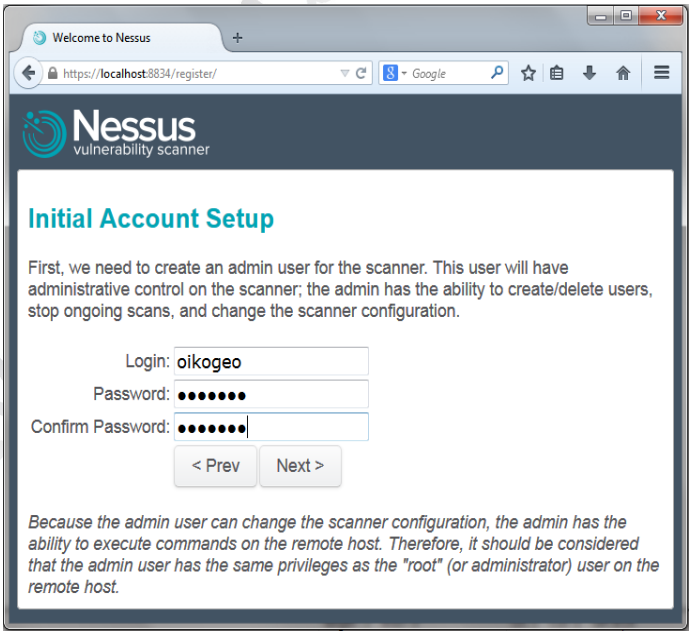
Εικόνα 7. Επιβεβαίωση εξαίρεση ασφαλείας

Καθώς θα γίνει επιβεβαίωση της εξαίρεσης ασφαλείας θα γίνει ανακατεύθυνση στην αρχική ιστοσελίδα για να ξεκινήσει η εγγραφή.



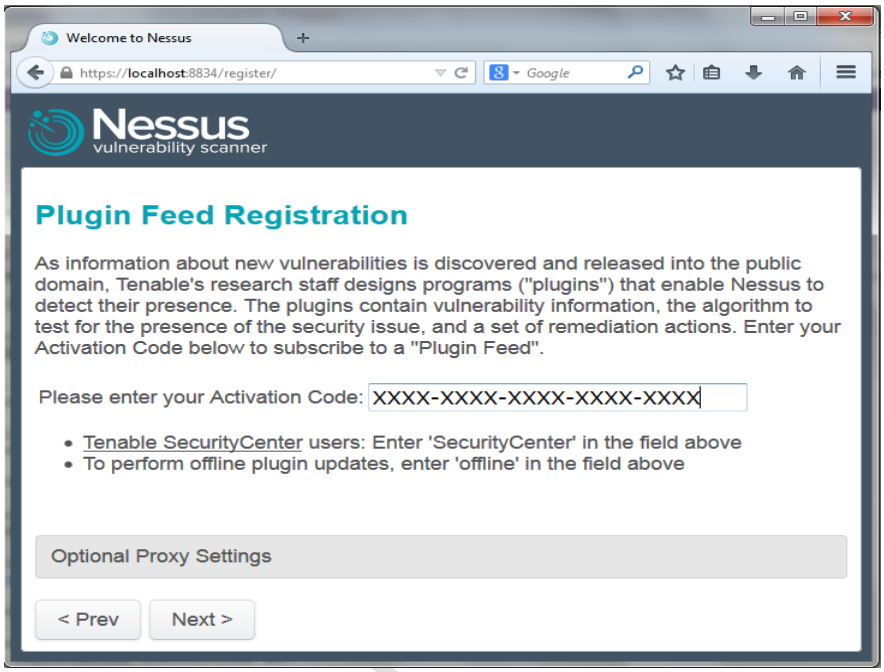
Εικόνα 8. Αρχική σελίδας εγγραφής

Η πρώτη ενέργεια είναι η δημιουργία ενός λογαριασμού για τον server του Nessus. Ο λογαριασμός αυτός θα έχει δικαιώματα διαχειριστή.



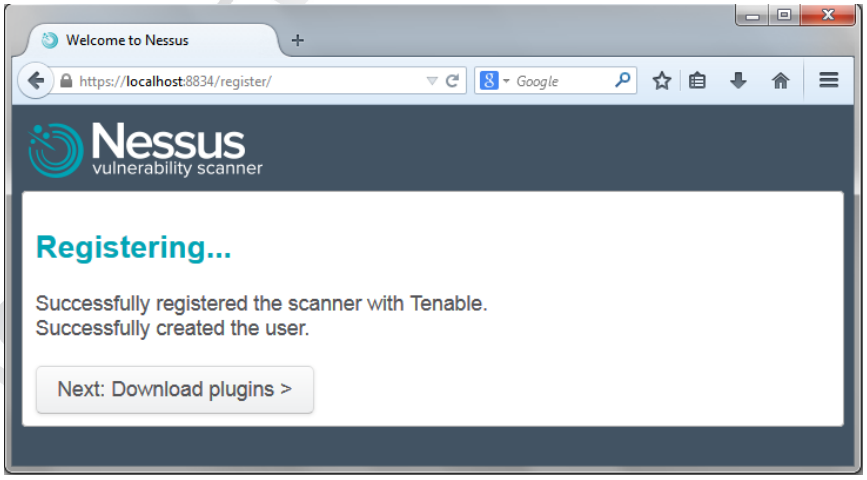
Εικόνα 9. Σελίδα δημιουργίας λογαριασμού χρήστη

Εφόσον δημιουργηθεί ο λογαριασμός, η επόμενη ενέργεια που απαιτείται είναι η εισαγωγή του κωδικού ενεργοποίησης των plugins. Όπως έχει αναφερθεί ο κωδικός που θα χρησιμοποιηθεί στην παρούσα εργασία είναι τύπου «Home Feed».



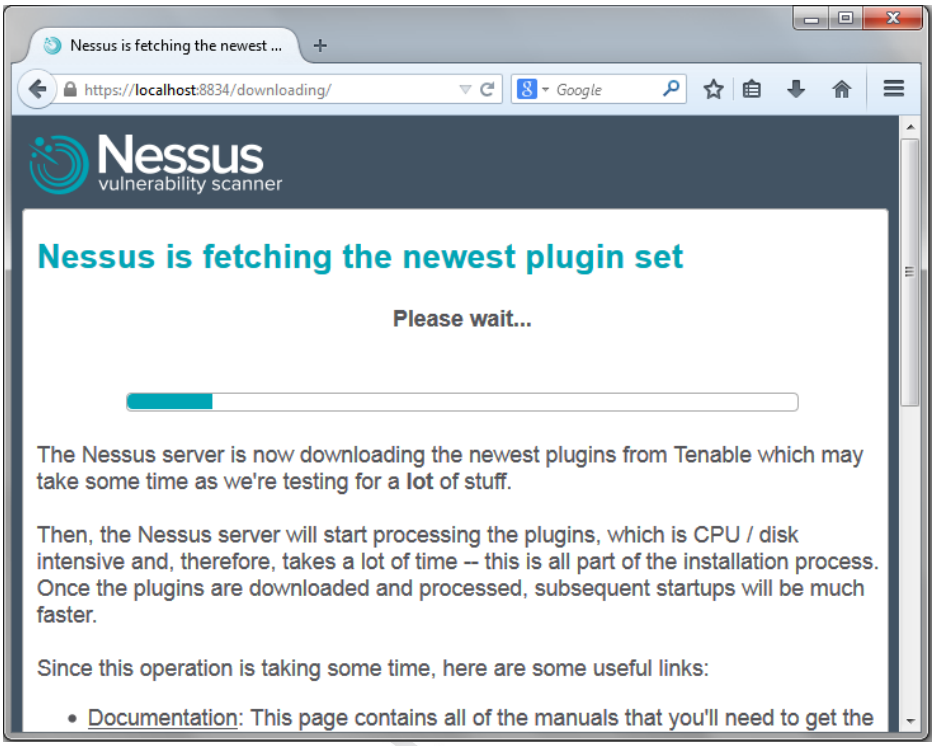
Εικόνα 10. Σελίδα εισαγωγής κωδικού ενεργοποίησης plugins

Στη συνέχεια όπως φαίνεται, γίνεται επιτυχής εγγραφή του σαρωτή με την Tenable και επιτυχής δημιουργία του χρήστη.



Εικόνα 11. Επιβεβαίωση εγγραφής σαρωτή και δημιουργίας λογαριασμού

Μετά την εγγραφή, το Nessus πρέπει να μεταφορτώσει τα plugins από την Tenable στην εσωτερική του βάση.



Εικόνα 12. Μεταφόρτωση plugins

Όταν τελειώσει η διαδικασία με τα plugins θα γίνει αρχικοποίηση του Nessus και ο server του Nessus θα ξεκινήσει, οπότε θα είναι έτοιμο προς χρήση.

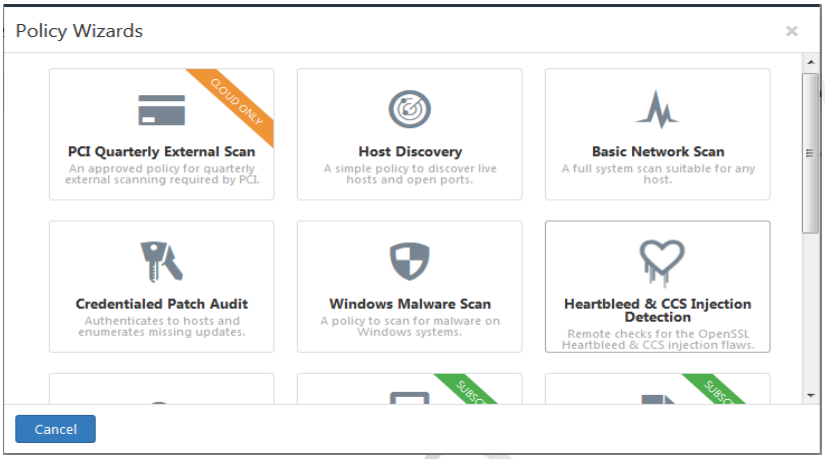
3.5 Επιλογές παραμετροποίησης πολιτικών του Nessus

Μια πολιτική (policy) του Nessus [33] αποτελείται από το σύνολο των επιλογών παραμετροποίησης που σχετίζονται με την εκτέλεση μιας σάρωσης για ευπάθειες (vulnerabilities). Αυτές οι επιλογές περιέχουν τα παρακάτω (αλλά δεν περιορίζονται μόνο σε αυτά):

- Παραμέτρους οι οποίοι ελέγχουν τεχνικά θέματα όπως τα χρονικά όρια, τον αριθμό των στόχων, ο τύπος του σαρωτή για τις θύρες και άλλα.
- Στοιχεία αυθεντικοποίησης για σαρώσεις τοπικές (π.χ. Windows, SSH), σε αυθεντικές βάσεις Oracle, σε HTTP, FTP, POP, IMAP ή βασισμένες σε αυθεντικοποίηση Kerberos.
- Προδιαγραφές σαρώσεων βασισμένες σε μεμονωμένα plugin ή οικογένειες αυτών
- Βάσεις δεδομένων συμβατές με πολιτικές ελέγχου, εκτενείς αναφορές, ανίχνευση υπηρεσιών για ρυθμίσεις σάρωσης, ελέγχους για Unix και άλλα.

3.5.1 Δημιουργία νέας πολιτικής

Μετά από τη σύνδεσή στην διεπαφή του server του Nessus, μπορεί να δημιουργηθεί μια ειδικά διαμορφωμένη πολιτική με την επιλογή «Πολιτικές» στην πάνω γραμμή και στη συνέχεια το κουμπί «+ Νέα Πολιτική» στα αριστερά. Η οθόνη προσθήκης πολιτικών θα είναι η ακόλουθη.



Εικόνα 13. Οθόνη προσθήκης πολιτικών

3.5.2 Χρήση του οδηγού πολιτικών (Policy Wizard)

Η πρώτη επιλογή είναι η προαιρετική χρήση του οδηγού πολιτικών, ώστε να βοηθήσει στην διαμόρφωση μιας πολιτικής που θα έχει ένα συγκεκριμένο σκοπό. Κάποια προκαθορισμένα πρότυπα είναι τα παρακάτω:

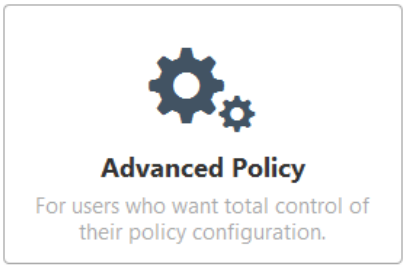
Όνομα Οδηγού Πολιτικής	Περιγραφή
PCI Quarterly External Scan	Μια αποδεκτή πολιτική για τρίμηνες εξωτερικές σαρώσεις οι οποίες απαιτούνται από το PCI. Αυτή η πολιτική προσφέρεται μόνο από το Nessus Enterprise Cloud.
Host Discovery	Εντοπίζει τους ενεργούς στόχους και τις ανοιχτές θύρες.
Basic Network Scan	Για χρήστες που σαρώνουν εσωτερικούς ή εξωτερικούς χρήστες.
Credentialed Patch Audit	Είσοδος στο σύστημα και απαρίθμηση των ανανεώσεων λογισμικού που εκκρεμούν.
Web Application Tests	Για χρήστες που εκτελούν γενικές σαρώσεις διαδικτυακών εφαρμογών.
Windows Malware Scan	Για χρήστες που αναζητούν κακόβουλο λογισμικό σε συστήματα Windows.

Mobile Device Scan	Για χρήστες του Apple Profile Manager, ADSI, MobileIron, ή Good MDM.
Offline Config Auditing	Ανεβάζει (upload) στο δίκτυο και ελέγχει το αρχείο config μιας συσκευής δικτύου.
Amazon AWS Audit	Για χρήστες που θέλουν να ελέγξουν διαχειριζόμενα συστήματα υποδομής AWS.
Prepare for PCI DSS Audits	Για διαχειριστές που προετοιμάζονται για έναν έλεγχο συμμόρφωσης PCI DSS.
Advanced Policy	Για χρήστες που θέλουν πλήρη έλεγχο της διαμόρφωσης της πολιτικής τους, αυτή αποτελεί μια προκαθορισμένη σάρωση.

Πίνακας 8. Οδηγός πολιτικών (Policy Wizard)

3.5.3 Δημιουργία Προηγμένης Πολιτικής (Advanced Policy)

Αν δεν είναι επιθυμητή η χρήση ενός οδηγού πολιτικής, η επιλογή “Advanced” δίνει την δυνατότητα να δημιουργηθεί μια πολιτική με τον παραδοσιακό τρόπο, έχοντας τον πλήρη έλεγχο διαμόρφωσης όλων των επιλογών από την αρχή.



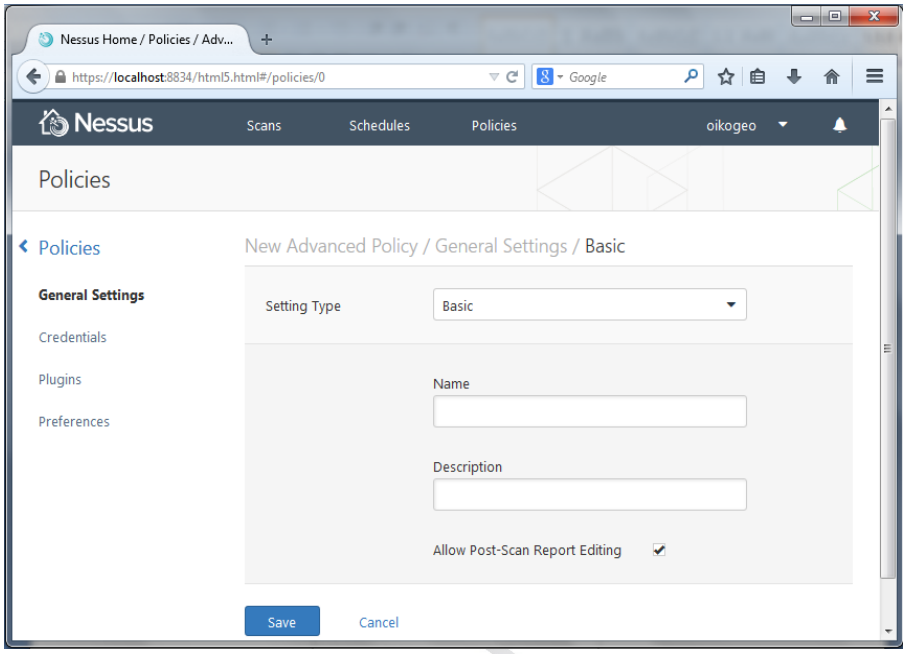
Εικόνα 14. Επιλογή Advanced Policy

Υπάρχουν τέσσερις επιλογές διαμόρφωσης : General Settings, Credentials, Plugins, και Preferences.

1. General Settings

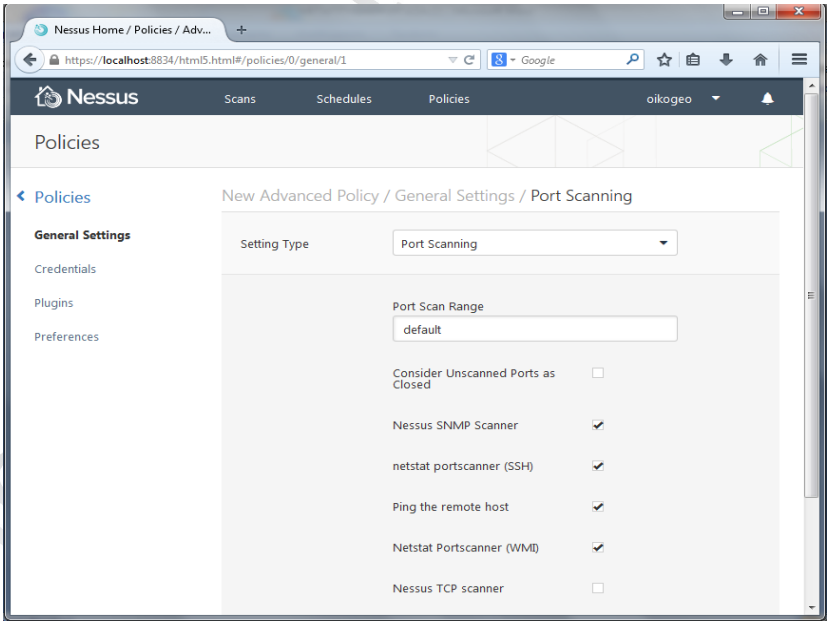
Η επιλογή “General Settings” δίνει την δυνατότητα ονομασίας της πολιτικής και διαμόρφωσης των λειτουργιών που είναι σχετικές με την σάρωση. Υπάρχουν τέσσερα μενού αναδίπλωσης (drop-down menu) τα οποία καθορίζουν την συμπεριφορά του σαρωτή.

Η βασική (“Basic”) οθόνη χρησιμοποιείται για να ορίζει πτυχές της πολιτικής από μόνη της.



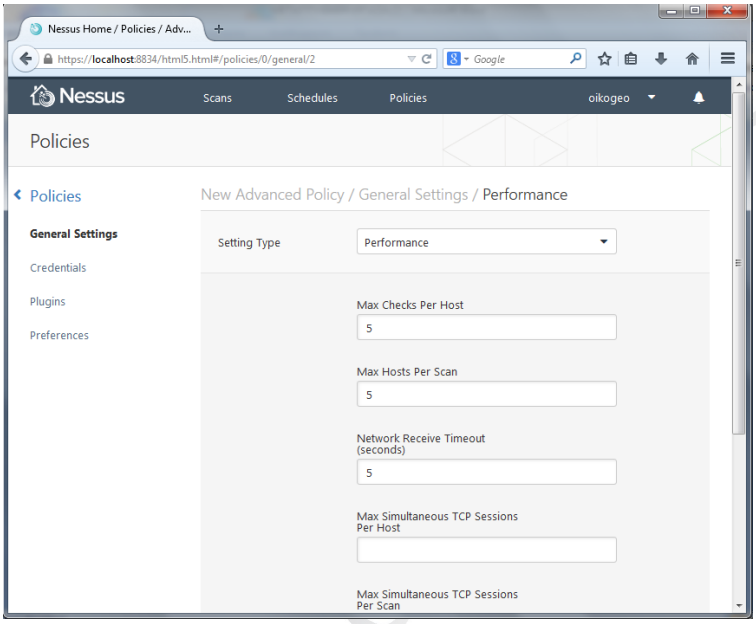
Εικόνα 15. Οθόνη General Settings-Basic

Το μενού “Port Scanning” έχει επιλογές σχετικές με τη σάρωση θυρών συμπεριλαμβανομένου τον ειδών των θυρών και των μεθόδων.



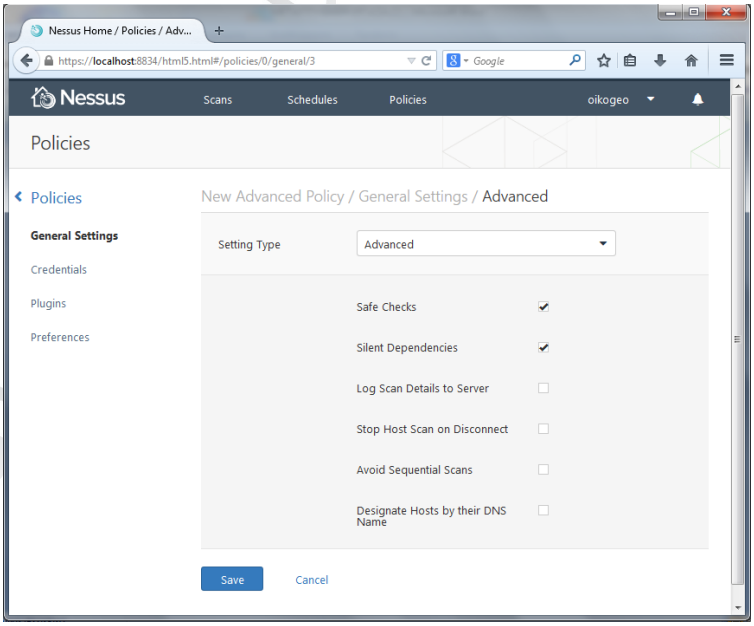
Εικόνα 16. Οθόνη General Settings-Port Scanning

Το μενού “Performance” παρέχει επιλογές που καθορίζουν πόσες σαρώσεις θα πραγματοποιηθούν. Αυτές οι επιλογές είναι πιθανόν οι πιο σημαντικές όταν διαμορφώνεται μια σάρωση καθώς έχουν την μεγαλύτερη επίδραση στους χρόνους της σάρωσης και στην δραστηριότητα του δικτύου.



Εικόνα 17. Οθόνη General Settings-Performance

Το μενού “Advanced” καθορίζει επιπλέον επιλογές σχετικές με την συμπεριφορά του σαρωτή.

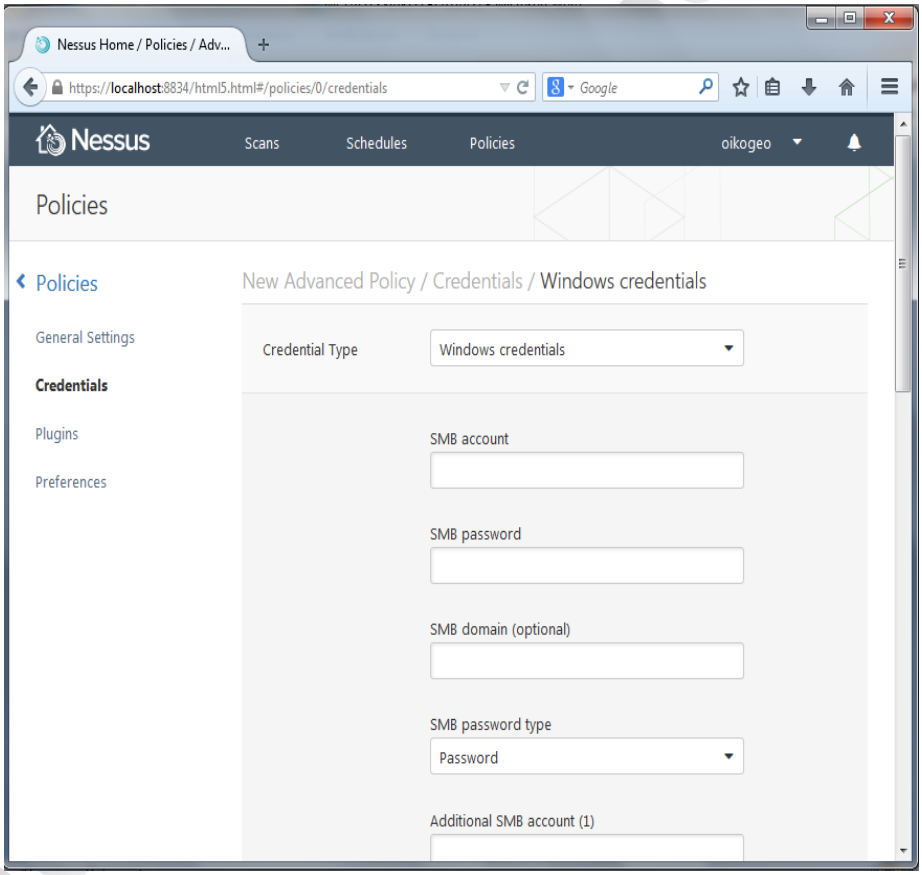


Εικόνα 18. Οθόνη General Settings-Advanced

2. Credentials

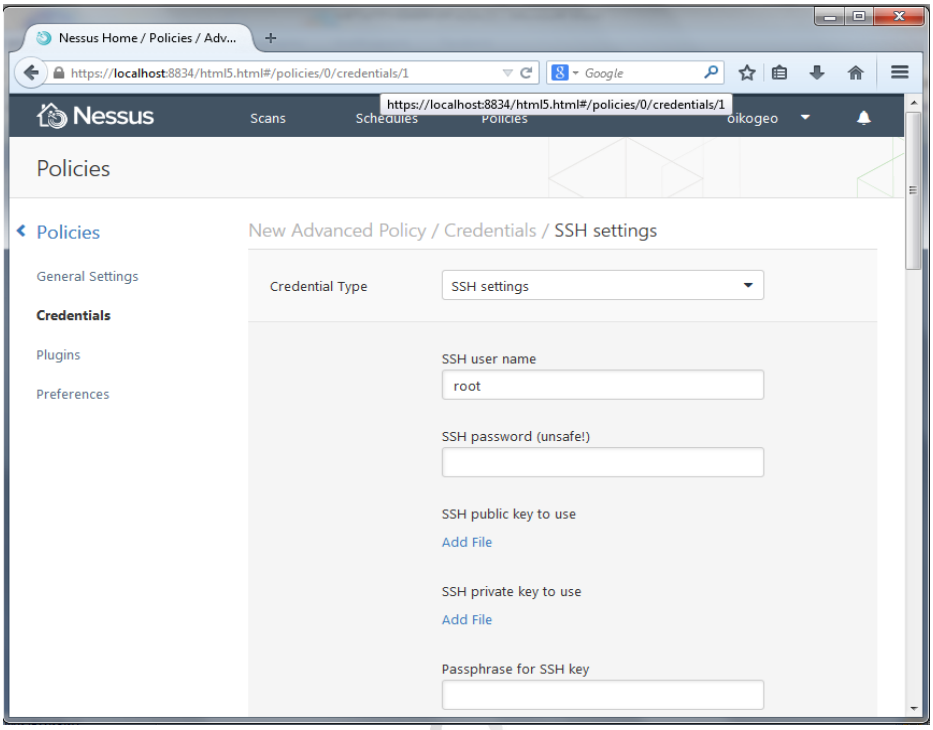
Η επιλογή “Credentials” δίνει την δυνατότητα διαμόρφωσης του σαρωτή Nessus ώστε να χρησιμοποιεί παραμέτρους αυθεντικοποίησης (Authentication Credentials) κατά την διάρκεια της σάρωσης. Αυτή η δυνατότητα επιτρέπει στο Nessus να πραγματοποιεί μια ευρύτερη ποικιλία ελέγχων που οδηγούν σε πιο ακριβή αποτελέσματα των διενεργούμενων σαρώσεων.

Η επιλογή “Windows credentials” στο μενού αναδίπλωσης έχει ρυθμίσεις που παρέχουν στο Nessus πληροφορίες όπως : το όνομα λογαριασμού του SMB, τον κωδικό πρόσβασης και το domain name. Το Server Message Block (SMB) είναι ένα αρχείο πρωτοκόλλου διαμοιρασμού το οποίο επιτρέπει στους υπολογιστές να μοιράζονται πληροφορίες σε όλο το δίκτυο με διαύγεια. Παρέχοντας αυτή την πληροφορία στο Nessus του επιτρέπεται να βρει την τοπική πληροφορία που σχετίζεται με ένα απομακρυσμένο στόχο ο οποίος χρησιμοποιεί Windows. Για παράδειγμα μπορεί να χρησιμοποιηθεί για να καθορίσει αν οι τελευταίες ενημερώσεις ασφαλείας είναι εγκαταστημένες.



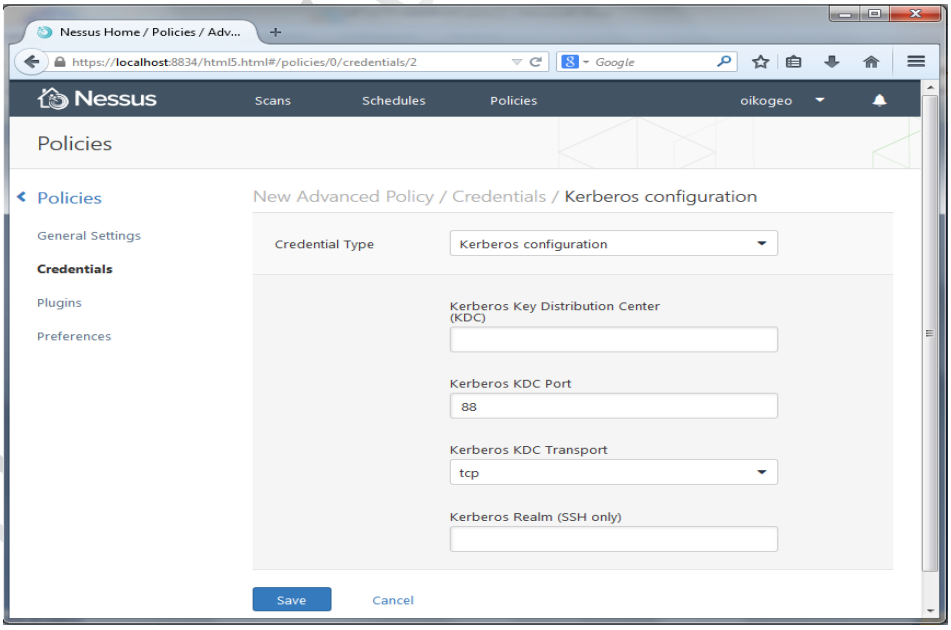
Εικόνα 19. Οθόνη Credentials-Windows credentials

Με την χρήση της επιλογής “SSH settings” και τις κατάλληλες ρυθμίσεις δίνεται η δυνατότητα σάρωσης συστημάτων Unix.



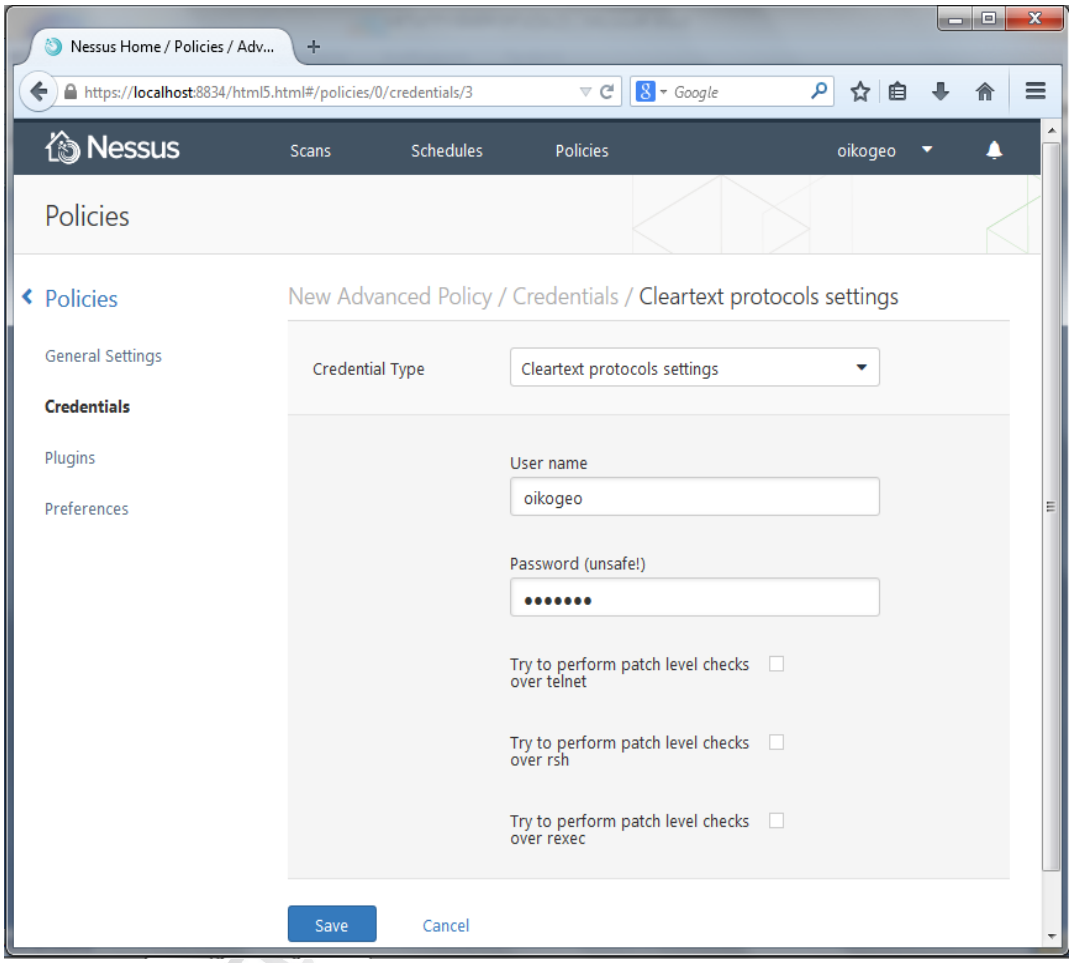
Εικόνα 20. Οθόνη Credentials-SSH Settings

Η επιλογή “Kerberos configuration” επιτρέπει να καθοριστούν τα credentials με την χρήση κλειδιών Kerberos από κάποιο απομακρυσμένο σύστημα.



Εικόνα 21. Οθόνη Credentials-Kerberos configuration

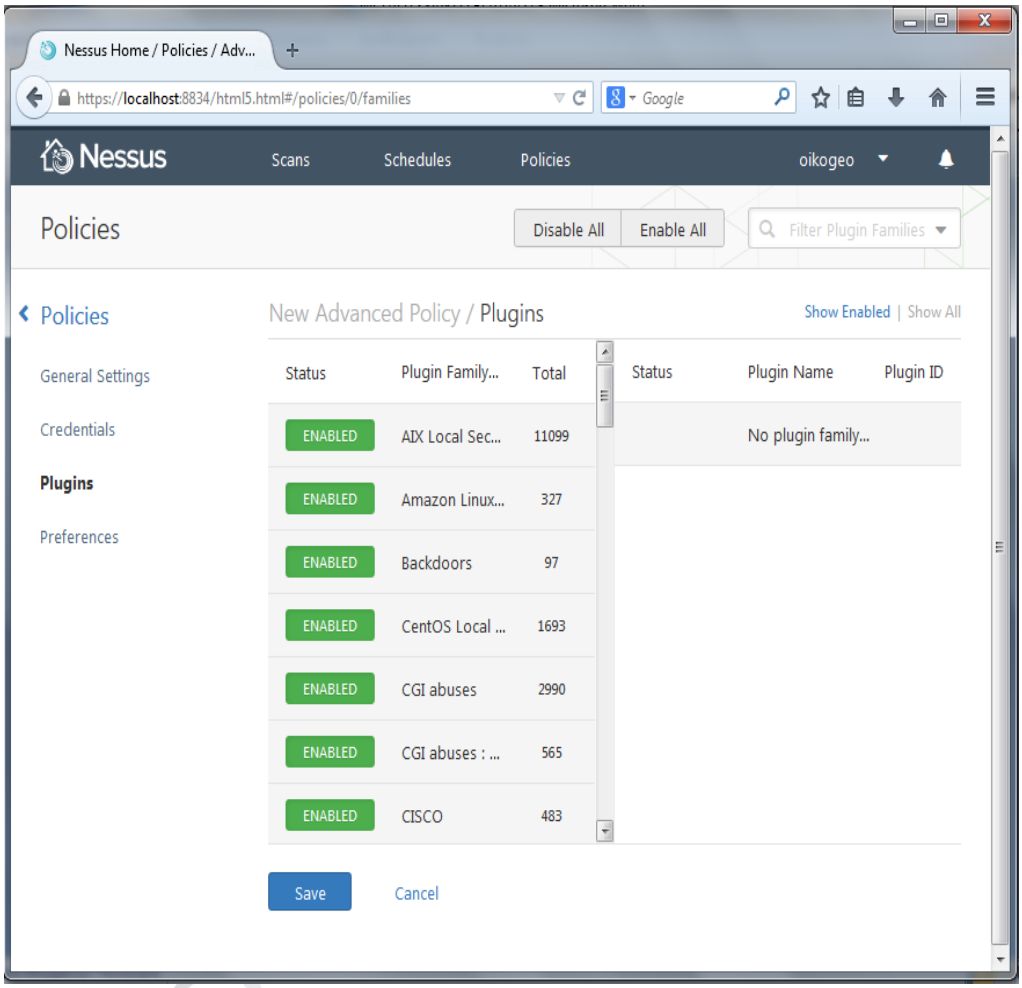
Τέλος με την επιλογή “Cleartext protocol settings” μπορεί να χρησιμοποιηθεί το Nessus για ελέγχους με μη ασφαλή πρωτόκολλα, αν δεν είναι διαθέσιμη η ασφαλής μέθοδος για ελέγχους αυθεντικοποίησης που χρησιμοποιούν credentials.



Εικόνα 22. Οθόνη Credentials-Cleartext protocol settings

3. Plugins

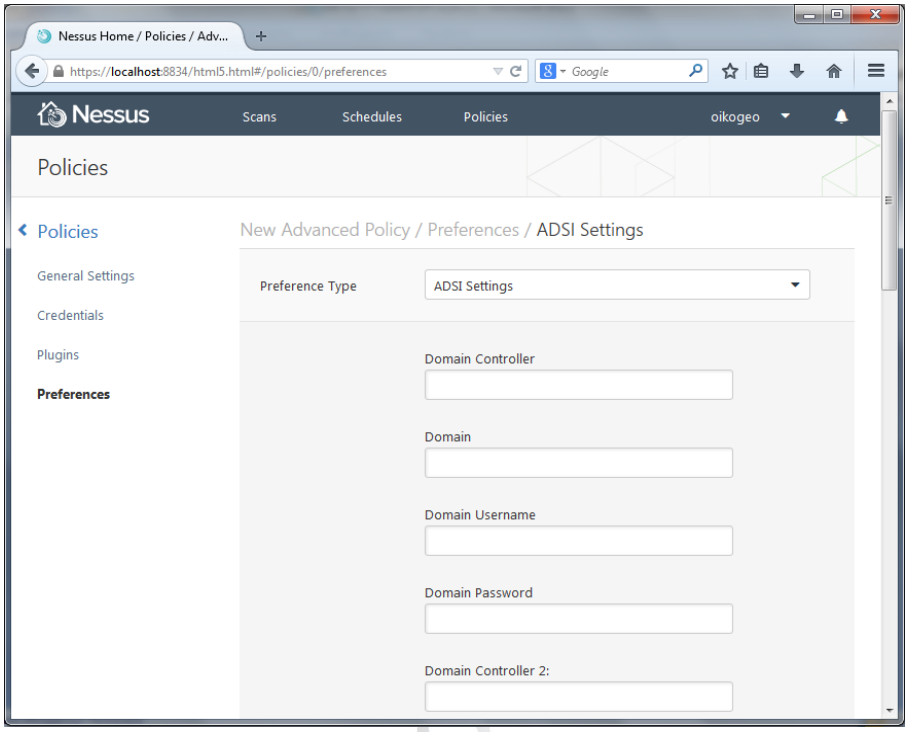
Η επιλογή “Plugins” δίνει την δυνατότητα επιλογής ειδικών ελέγχων ασφαλείας που χρησιμοποιούν μεμονωμένα Plugins ή μια οικογένεια αυτών.



Εικόνα 23. Οθόνη Plugins

4. Preferences

Η επιλογή “Preferences” περιέχει την δυνατότητα για την πιο λεπτομερή διαμόρφωση των ρυθμίσεων μιας πολιτικής σάρωσης.



Εικόνα 24. Οθόνη Preferences

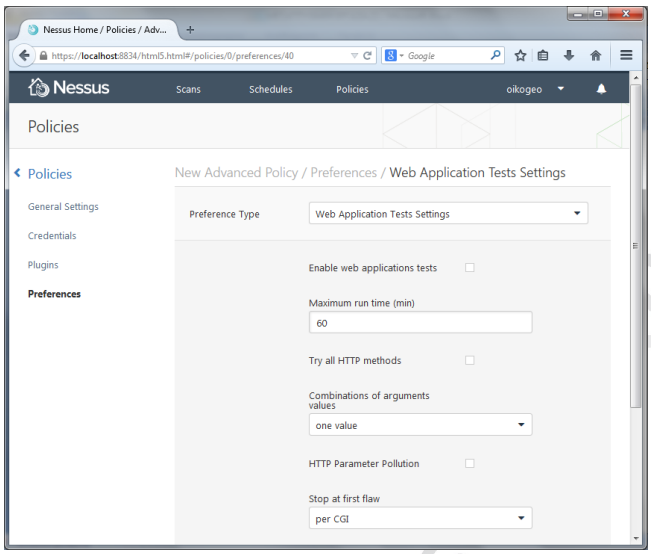
3.5.4 Web Application Tests Settings

Όσον αφορά την παρούσα εργασία οι συγκεκριμένες ρυθμίσεις που αφορούν τις διαδικτυακές εφαρμογές είναι οι πλέον σημαντικές. Το μενού των "Web Application Tests Settings" ελέγχει για αδυναμίες του απομακρυσμένου CGI (Common Gateway Interface) που έχουν ανακαλυφθεί με την διαδικασία web mirroring, κατά την διάρκεια της προσπάθειας εκτέλεσης κώδικα που περιέχει κοινά προγραμματιστικά λάθη CGI όπως είναι cross-site scripting, remote file inclusion, command execution, traversal attacks, και SQL injection. Αυτοί οι έλεγχοι εξαρτώνται από τα παρακάτω plugins:

- 11139 (CGI Generic SQL Injection), 42424 (CGI Generic SQL Injection-blind), 42479 (CGI Generic SQL Injection- 2nd pass), 42426 (CGI Generic SQL Injection – HTTP Cookies), 42427 (CGI Generic SQL Injection – HTTP Headers), 43160 (CGI Generic SQL Injection-blind,time based) – **SQL Injection (CGI abuses).** Αυτά τα plugins ελέγχουν για ευπάθειες που ανήκουν στην κατηγορία A1 – Injection του Top10 του OWASP 2013.
- 39465 (CGI Generic Command Execution), 44967 (CGI Generic Command Execution-time based) **– Command Execution (CGI abuses).** Αυτά τα plugins ελέγχουν για ευπάθειες που ανήκουν στην κατηγορία A3 – Malicious File Execution του Top10 του OWASP 2007.
- 39466 (CGI Generic XSS - quick test), 47831 (CGI Generic XSS comprehensive test), 42425 (CGI Generic XSS - persistent), 46193 (CGI Generic XSS - HTTP Headers), 49067 (CGI

Generic HTML Injections - quick test) – **Cross-Site Scripting (CGI abuses: XSS).** Αυτά τα plugins ελέγχουν για ευπάθειες που ανήκουν στην κατηγορία A3 – Cross-Site Scripting (XSS) του Top10 του OWASP 2013.

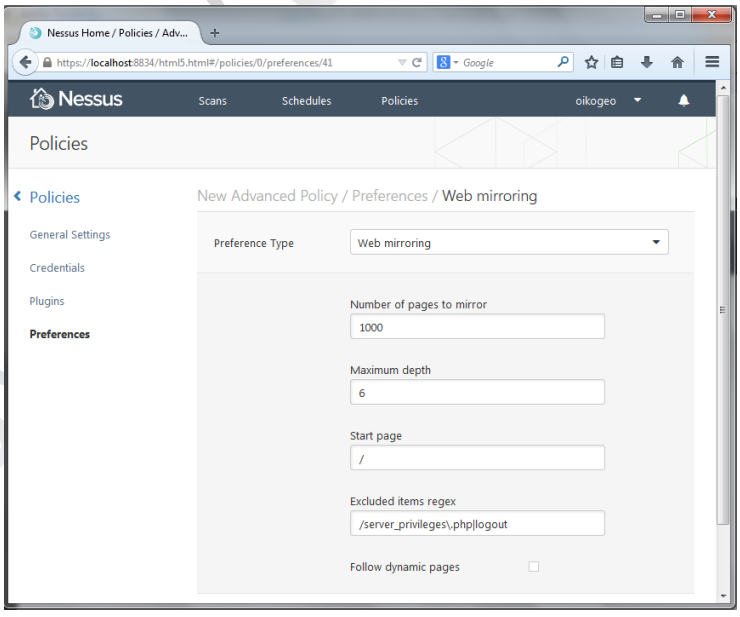
- 39467 (CGI Generic Path Traversal), 46195 (CGI Generic Path Traversal - extended test), 46194 (CGI Generic Path Traversal - write test) – **Directory Traversal (CGI abuses).** Αυτά τα plugins ελέγχουν για ευπάθειες που ανήκουν στην κατηγορία A6 – Information Leakage and Improper Error Handling του Top10 του OWASP 2007.
- 39468 (CGI Generic Header Injection)– **HTTP Header Injection (CGI abuses: XSS).** Αυτό το plugin ελέγχει για ευπάθειες που ανήκουν στην κατηγορία κατηγορία A1 – Injection του Top10 του OWASP 2013.
- 39469 (CGI Generic Remote File Inclusion), 42056 (CGI Generic Local File Inclusion), 42872 (CGI Generic Local File Inclusion - 2nd pass) – **File Inclusion (CGI abuses).** Αυτό το plugin ελέγχει για ευπάθειες που ανήκουν στην κατηγορία A6 – Information Leakage and Improper Error Handling του Top10 του OWASP 2007.
- 42055 (CGI Generic Format String) – **Format String (CGI abuses).** Αυτό το plugin ελέγχει για ευπάθειες μορφοποίησης συμβολοσειρών που δεν ανήκουν σε κάποια από τις κατηγορίες του OWASP.
- 42423 (CGI Generic SSI Injection - HTTP headers), 42054 (CGI Generic SSI Injection) – **Server Side Includes (CGI abuses).** Αυτά τα plug-ins ελέγχουν για ευπάθειες που ανήκουν στην κατηγορία A6 – Information Leakage and Improper Error Handling του Top10 του OWASP 2007.
- 44136 (CGI Generic Cookie Injection Scripting) – **Cookie Manipulation (CGI abuses).** Αυτό το plugin ελέγχει για ευπάθειες χειραγώγησης των cookies που δεν ανήκουν σε κάποια από τις κατηγορίες του OWASP.
- 46196 (CGI Generic XML Injection) **– XML Injection (CGI abuses).** Αυτό το plugin ελέγχει για ευπάθειες που ανήκουν στην κατηγορία A1 – Injection του Top10 του OWASP 2013.
- 40406 (CGI Generic Tests HTTP Errors), 48926 (CGI Generic 2nd Order SQL Injection Detection - potential), 48927 (CGI Generic SQL Injection Detection - potential, 2nd order, 2nd pass) – **Error Messages.** Αυτά τα plug-ins ελέγχουν για ευπάθειες που ανήκουν στην κατηγορία A6 – Information Leakage and Improper Error Handling του Top10 του OWASP 2007.
- 47830 (CGI Generic Injectable Parameter), 47832 (CGI Generic On Site Request Forgery (OSRF)), 47834 (CGI Generic Open Redirection), 44134 (CGI Generic Unseen Parameters Discovery) **– Additional attacks (CGI abuses).** Αυτά τα plug-ins ελέγχουν για ευπάθειες επιπλέον επιθέσεων που δεν ανήκουν σε κάποια από τις κατηγορίες του OWASP



Εικόνα 25. Οθόνη Preferences-Web Application Tests Settings

Web mirroring

Το μενού του “Web mirroring” καθορίζει την παραμετροποίηση του “mirroring“ (αντικατοπτρισμού) του web server του Nessus. Το Nessus θα αντικατοπτρίσει το περιεχόμενο μιας διαδικτυακής εφαρμογής για να αναλύσει καλύτερα τα περιεχόμενα της για ευπάθειες και θα βοηθήσει να ελαχιστοποιηθούν οι επιπτώσεις στον server.



Εικόνα 26. Οθόνη Preferences-Web Mirroring

Κεφάλαιο 4ο

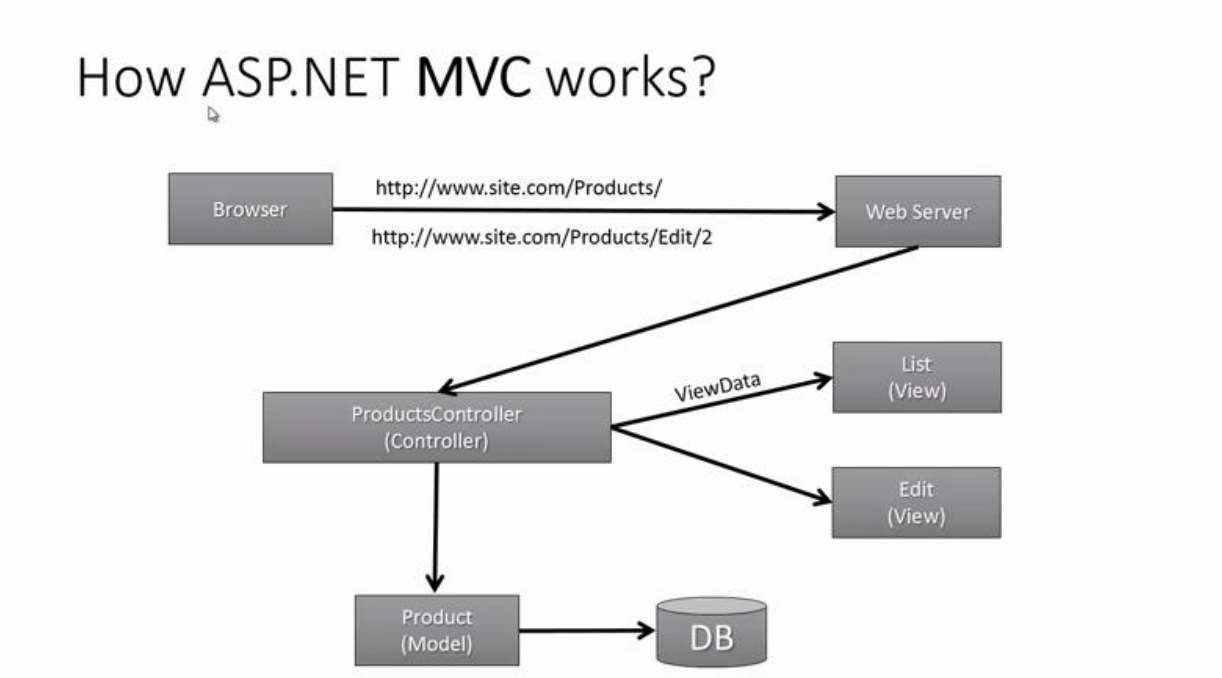
4 Ανάπτυξη Δοκιμαστικής Εφαρμογής και Έλεγχος Ευπαθειών

4.1 Εισαγωγή

Στο πλαίσιο εκπόνησης της παρούσας διπλωματικής εργασίας θα χρησιμοποιηθεί η διαδικτυακή εφαρμογή Online Movies η οποία θα ελεγχθεί με το Nessus για να εντοπιστεί η παρουσία πιθανών ευπαθειών. Η εφαρμογή δημιουργήθηκε με σκοπό να παρουσιάσει τα μέτρα ασφαλείας που μπορούν να ληφθούν κατά την ανάπτυξη μιας διαδικτυακής εφαρμογής τόσο προληπτικά όσο και εκ των υστέρων μετά από έλεγχο αυτής. Αυτά τα μέτρα θα παρουσιαστούν παρακάτω με απώτερο σκοπό να καταστεί απροσπέλαστη από τους δυνητικούς επιτιθέμενους.

4.2 Παρουσίαση διαδικτυακής εφαρμογής Online Movies

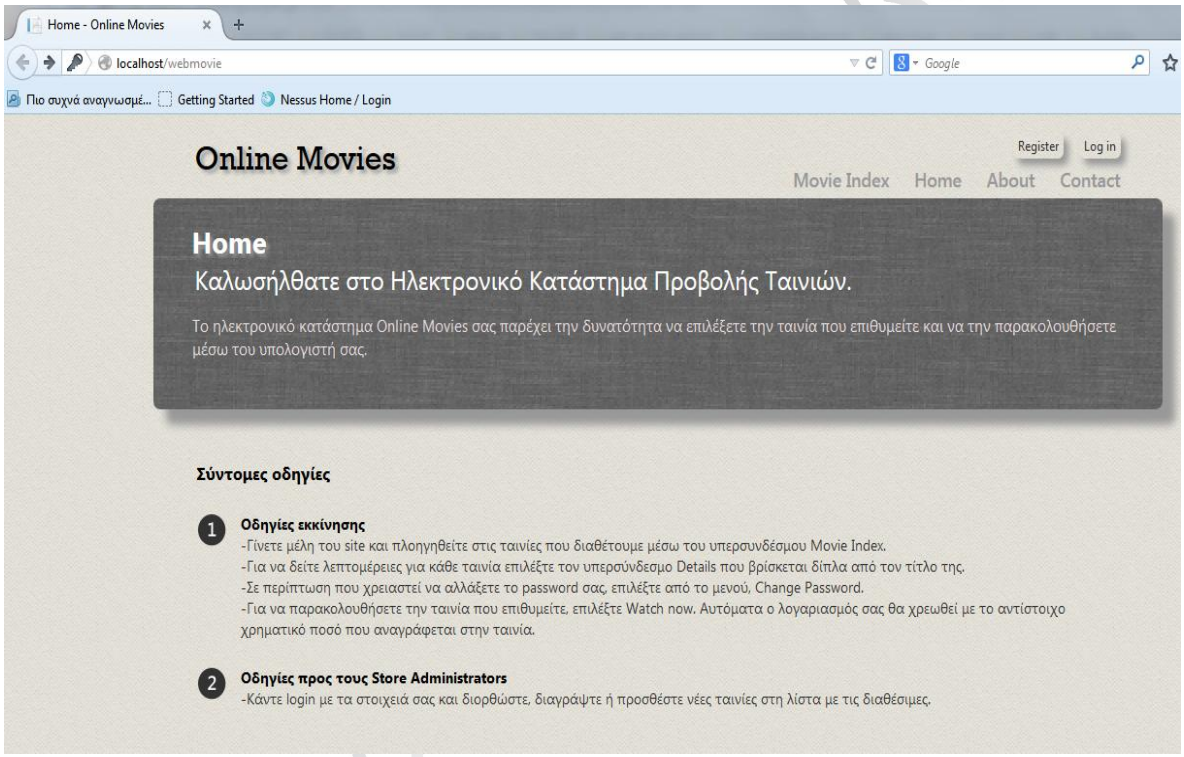
Η διαδικτυακή εφαρμογή Online Movies αποτελεί ένα ηλεκτρονικό κατάστημα ενοικίασης κινηματογραφικών ταινιών. Υλοποιήθηκε σε περιβάλλον Microsoft Visual Studio 2012 Ultimate (VS12) με την χρήση του μοντέλου αρχιτεκτονικής λογισμικού Model-View-Controller (MVC) [34] και της γλώσσας προγραμματισμού C# .



Εικόνα 27. Λειτουργία αρχιτεκτονικής MVC

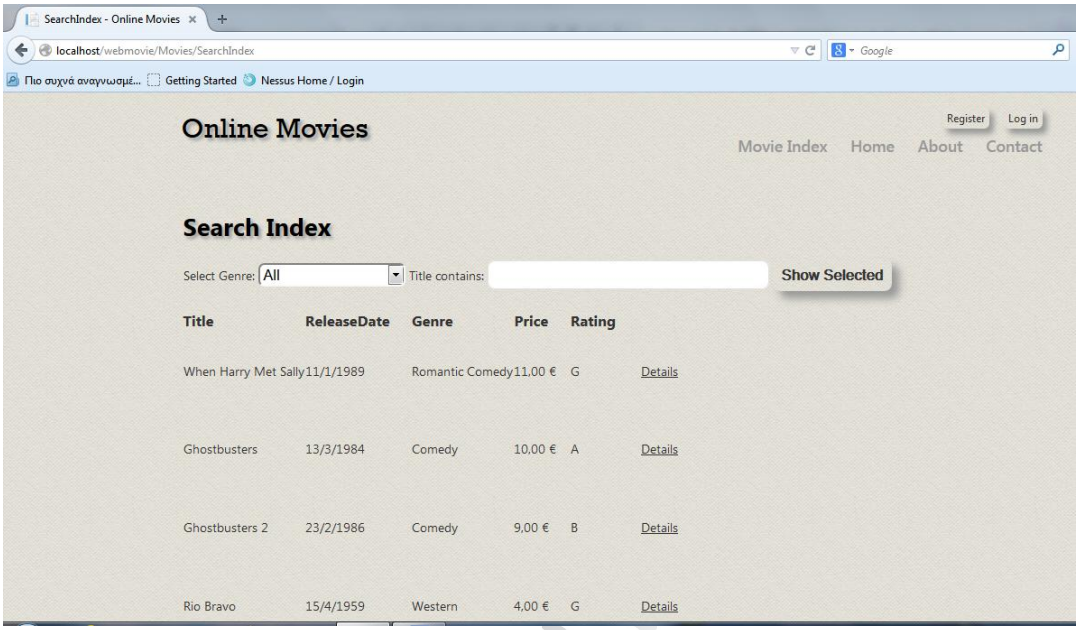
Το λειτουργικό σύστημα είναι Windows 7 Home Premium και για την υλοποίηση διαφόρων χαρακτηριστικών της όσον αφορά την ασφάλεια ρυθμίστηκαν και χρησιμοποιήθηκαν οι servers IIS (Microsoft Internet Information Services) 7.5 και SQL Express 11.0.

Αποτελείται από την αρχική ιστοσελίδα (Home) που εμφανίζεται όταν ο χρήστης θέλει να χρησιμοποιήσει τον ιστότοπο Online Movies. Είναι η εισαγωγική ιστοσελίδα που καλωσορίζει τους χρήστες και έχει σύντομες οδηγίες. Σε αυτή υπάρχουν οι σύνδεσμοι των ιστοσελίδων (Movie Index), (About) και (Contact). Όλες οι προαναφερόμενες ιστοσελίδες μπορούν να χρησιμοποιηθούν χωρίς να απαιτείται να είναι κάποιος εγγεγραμμένος χρήστης και όλες παρέχουν την δυνατότητα στον επισκέπτη μέσω του συνδέσμου login να συνδεθεί εφόσον διαθέτει λογαριασμό χρήστη ή να δημιουργήσει λογαριασμό μέσω του συνδέσμου register.



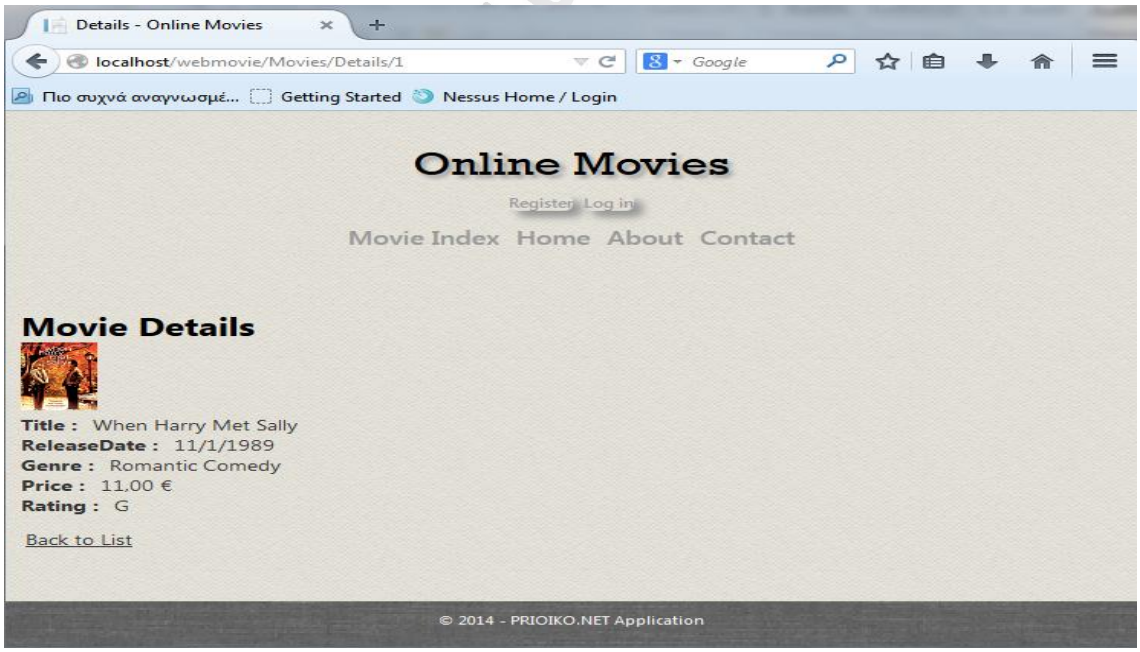
Εικόνα 28. Ιστοσελίδα Home εφαρμογής Online Movies

Η ιστοσελίδα (Movie Index) περιέχει τον κατάλογο των διαθέσιμων ταινιών που μπορεί να ενοικιάσει ο χρήστης. Παρέχει πληροφορίες για κάθε ταινία και την δυνατότητα αναζήτησης κάποιας ταινίας με βάση το όνομα της ή την κατηγορία της.



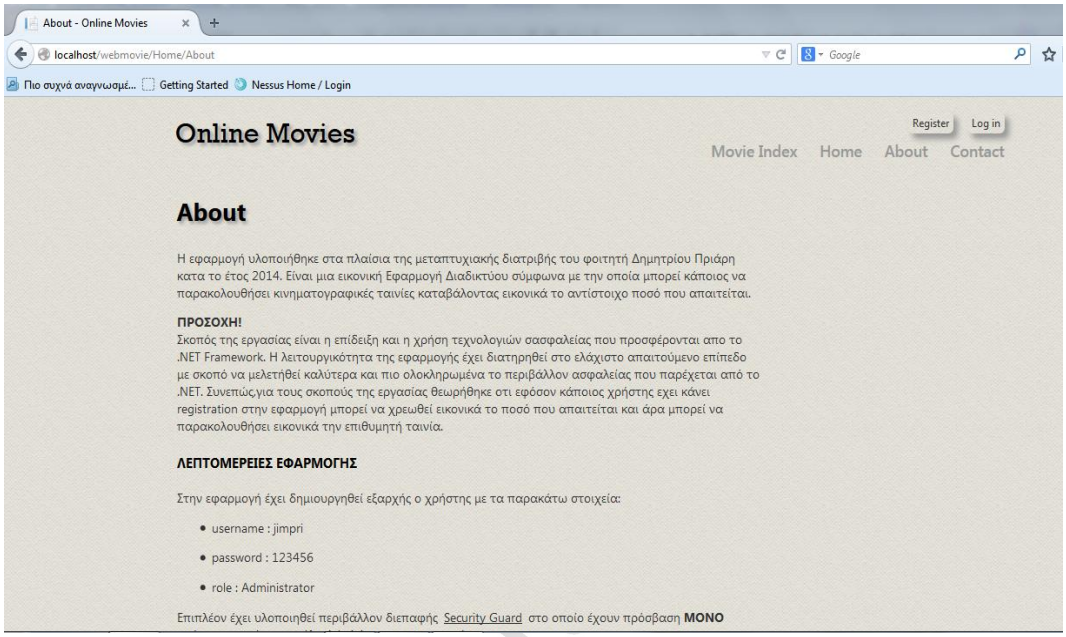
Εικόνα 29. Ιστοσελίδα Movie Index εφαρμογής Online Movies

Επίσης περιέχει τον σύνδεσμο (Details) που επιλέγοντας τον ο χρήστης μπορεί να ενημερωθεί αναλυτικά για κάθε ταινία χωριστά.



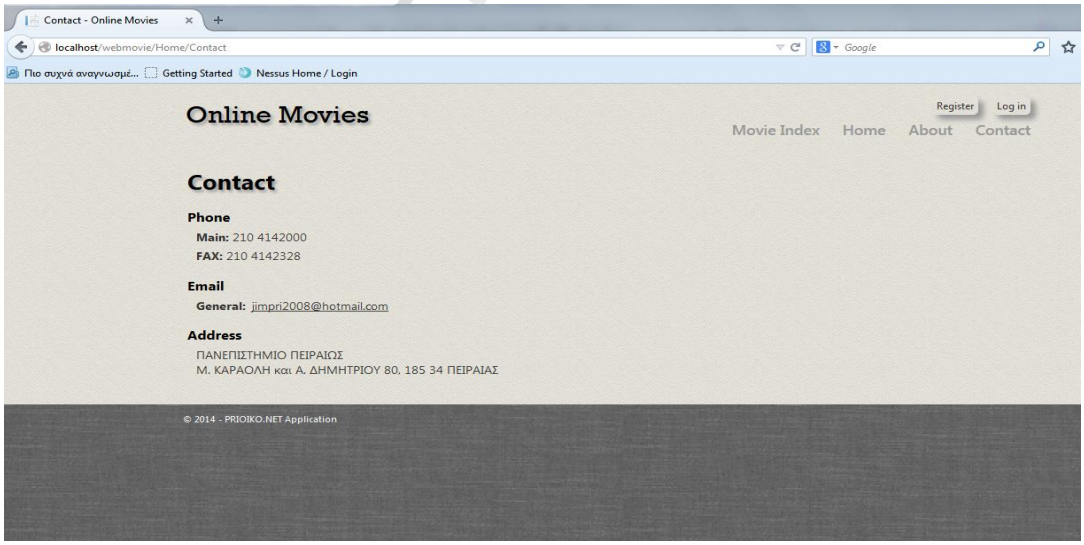
Εικόνα 30. Ιστοσελίδα Details εφαρμογής Online Movies

Η ιστοσελίδα (About) εξηγεί τον λόγο για τον οποίο δημιουργήθηκε η εφαρμογή και ότι σκοπός της είναι η επίδειξη και η χρήση τεχνολογιών ασφάλειας που διατίθενται από το .NET Framework [35]. Επίσης ενημερώνει τον χρήστη για τις λεπτομέρειες της εφαρμογής όσον αφορά τον διαχειριστή ασφαλείας (Administrator) του ιστότοπου και το περιβάλλον διεπαφής Security Guard [36].



Εικόνα 31. Ιστοσελίδα About εφαρμογής Online Movies

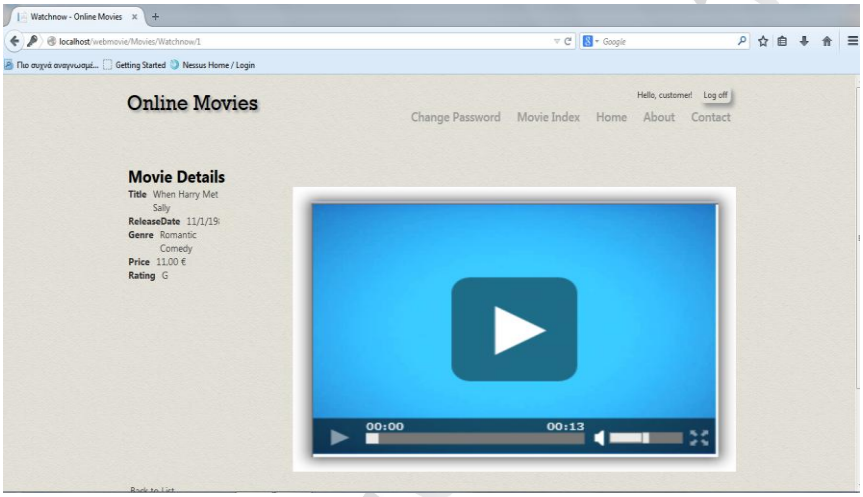
Τέλος η ιστοσελίδα (Contact) περιέχει τα στοιχεία επικοινωνίας του ιστότοπου.



Εικόνα 32. Ιστοσελίδα Contact εφαρμογής Online Movies

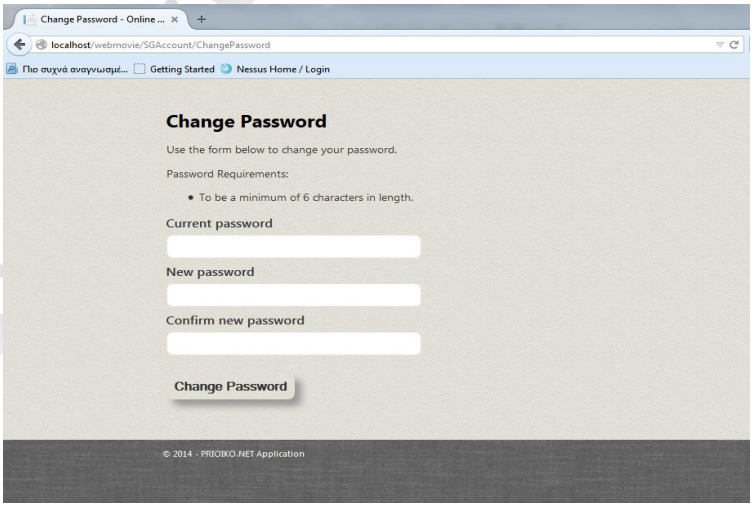
Στους εγγεγραμμένους χρήστες ο ιστότοπος παρέχει περισσότερες δυνατότητες χρήσης. Οι ρόλοι που υπάρχουν σε αυτή την εφαρμογή είναι οι εξής: α) Customer β) Store Administrator και γ) Administrator. Ο ρόλος Administrator δημιουργήθηκε κατά την ανάπτυξη της εφαρμογής όπου αποτελεί και τον διαχειριστή ασφαλείας του ιστοτόπου. Στη συνέχεια με την χρήση της διεπαφής Security Guard [36] δημιουργήθηκαν οι ρόλοι Customer και Store Administrator.

Ο χρήστης (Customer) μπορεί να επιλέξει να παρακολουθήσει κάποια ταινία μέσω του συνδέσμου (Watch now) με την καταβολή του αντίστοιχου χρηματικού αντιτίμου,



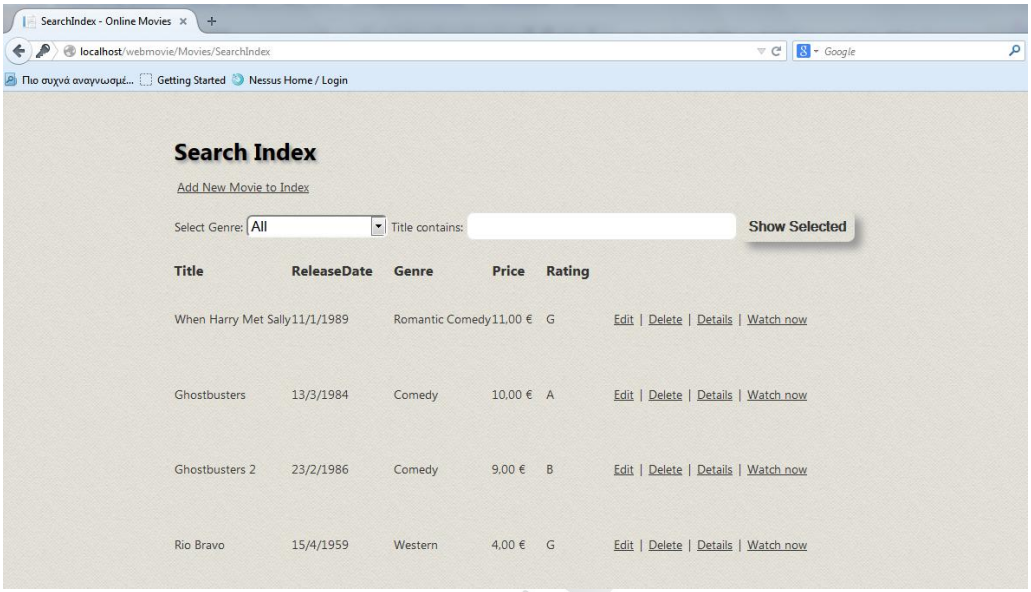
Εικόνα 33. Ιστοσελίδα Watch now εφαρμογής Online Movies

ή να αλλάξει το κωδικό πρόσβασης (password) που χρησιμοποιεί για τον συγκεκριμένο λογαριασμό μέσω του συνδέσμου (Change Password).



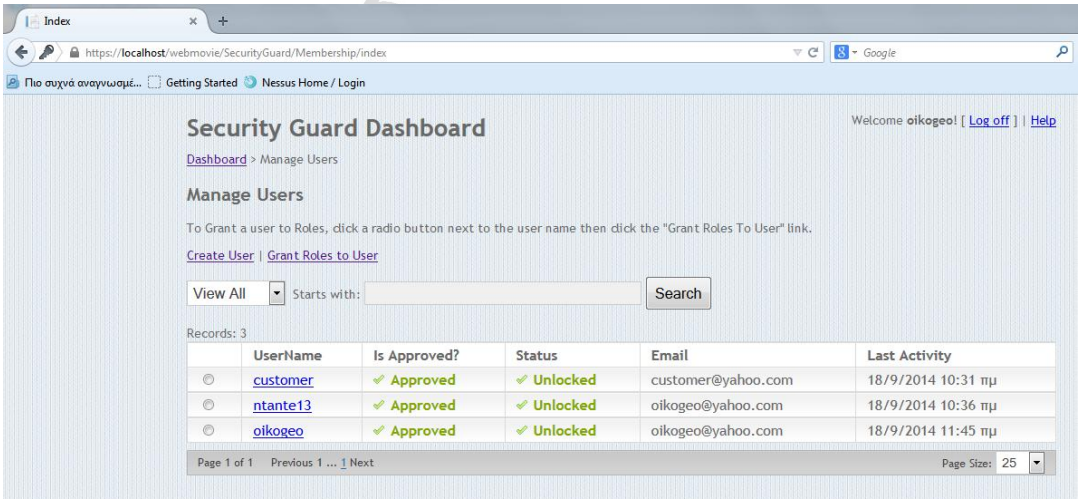
Εικόνα 34. Ιστοσελίδα Change Password

Ο χρήστης (Store Administrator) έχει όλες τις δυνατότητες χρήσης του (Customer) και επιπλέον μπορεί να διαχειριστεί τις ταινίες που περιέχει ο ιστότοπος με την πρόσθεση νέων (Create). Μπορεί να διαγράψει (Delete) ταινίες και να αλλάξει τα στοιχεία (Edit) τους μέσω της σελίδας (Movie Index) και την χρήση των αντίστοιχων συνδέσμων.



Εικόνα 35. Ιστοσελίδα Movie Index αυθεντικοποιημένου χρήστη (Store Administrator) εφαρμογής Online Movies

Ο χρήστης (Administrator) έχει δικαιώματα χρήσης όλων των δυνατοτήτων της εφαρμογής. Επιπλέον μέσω του συνδέσμου (Security Guard) μπορεί να διαχειριστεί όλους τους χρήστες, να δημιουργήσει νέους και να αναθέσει και να διαχειριστεί τους ρόλους των χρηστών.



Εικόνα 36. Ιστοσελίδα Security Guard εφαρμογής Online Movies

Τέλος, πρέπει να επισημανθεί ότι όταν κάποιος χρήστης κάνει registration δεν έχει αυτόματα τη δυνατότητα να χρησιμοποιήσει τις δυνατότητες που του παρέχει ο ρόλος του, πριν την ανάθεση του ρόλου σε αυτόν από το διαχειριστή χρηστών (Administrator).

4.3 Μέτρα ασφάλειας εφαρμογής

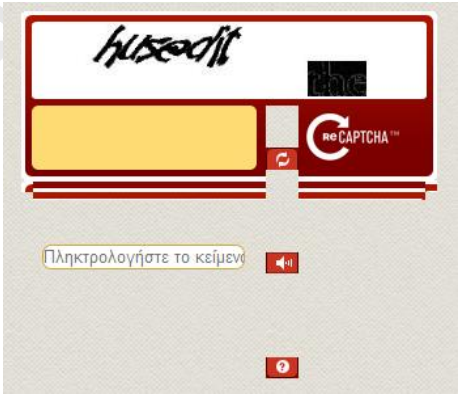
Κατά την διάρκεια ανάπτυξης της εφαρμογής ελήφθησαν τα παρακάτω μέτρα ασφαλείας. Όπως αναφέρθηκε και στη περιγραφή της εφαρμογής, χρησιμοποιήθηκε ο IIS server 7.5 που δίνει την δυνατότητα χρήσης της υπηρεσίας SSL (Secure Sockets Layer). Παραμετροποιήθηκε κατάλληλα ώστε να ενεργοποιηθεί το πρωτόκολλο https στη θύρα 443 και το http στη θύρα 80. Έτσι δόθηκε η δυνατότητα χρήσης σύνδεσης με κρυπτογραφία για κάποιες σελίδες του ιστότοπού μας.

Για την αυθεντικοποίηση (authentication) των χρηστών εγκαταστάθηκε και ρυθμίστηκε κατάλληλα το πρόσθετο στοιχείο (component) Security Guard. Αυτό δίνει την δυνατότητα εγγραφής (register) ενός νέου χρήστη ή της σύνδεσης (login) ενός υφιστάμενου. Επίσης, την εξουσιοδότηση (authorization) των χρηστών την υποστηρίζει με τους ρόλους που μπορεί να δημιουργήσει και να διαχειριστεί ο administrator. Όταν οι χρήστες δεν είναι κατάλληλα αυθεντικοποιημένοι ή εξουσιοδοτημένοι τότε ο web browser τους προτρέπει να εισάγουν τα στοιχεία του λογαριασμού τους ώστε να ταυτοποιηθούν. Η σωστή εγκατάσταση και λειτουργία του Security Guard απαίτησε την εγκατάσταση και παραμετροποίηση του SQL server Express 11.0.

Για την ασφάλεια των χρηστών, κατά την διαδικασία εγγραφής τους προστέθηκε η απόρρητη ερώτηση ασφαλείας (secret question) και η απόρρητη απάντηση ασφαλείας (secret answer) ώστε να μπορεί να γίνει επαναφορά του κωδικού ασφαλείας σε περίπτωση που υπάρχει απώλεια του από τον χρήστη.

Εικόνα 37. Ερώτηση και Απάντηση Ασφαλείας επαναφοράς κωδικού

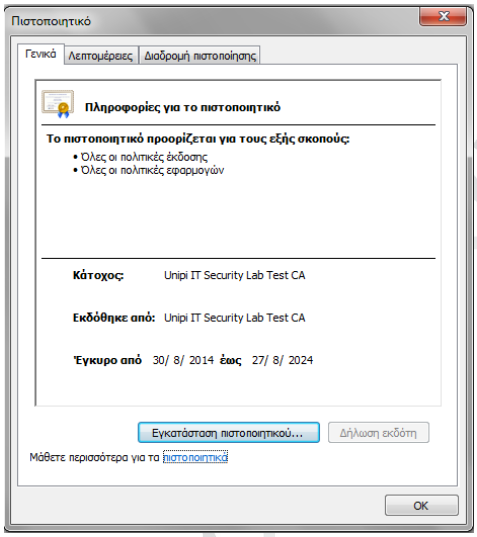
Προστέθηκε επίσης η λειτουργία Captcha για να αποτραπούν επιθέσεις τύπου «bots» όπου με αυτοματοποιημένα scripts ο επιτιθέμενος θα γέμιζε τη βάση με έγγραφές δημιουργώντας πρόβλημα στον ιστότοπο.



Εικόνα 38. Λειτουργία Captcha

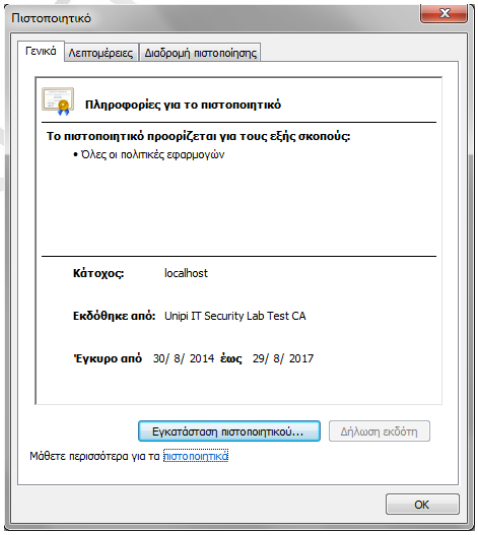
Επιπλέον, οι χρήστες με την υπηρεσία αλλαγής κωδικού πρόσβασης (change password) μπορούν όποτε θελήσουν να αλλάξουν τον κωδικό τους για να αποφύγουν παραβιάσεις του λογαριασμού τους.

Δημιουργήθηκε επίσης πιστοποιητικό για τον web server. Αυτό έγινε με την χρήση του εργαλείου OpenSSL όπου δημιουργήθηκε η Αρχή Πιστοποίησης με όνομα Unipi IT Security Lab Test CA με αυτο-υπογεγραμμένο Πιστοποιητικό.



Εικόνα 39. Πιστοποιητικό Αρχής Πιστοποίησης

Εγκαταστάθηκε αυτό το πιστοποιητικό ώστε η Αρχή Πιστοποίησης να είναι έμπιστη στο σύστημά και στη συνέχεια δημιουργήθηκε το πιστοποιητικό για τον server με όνομα localhost, υπογεγραμμένο από την Αρχή Πιστοποίησης Unipi IT Security Lab Test CA.



Εικόνα 40. Πιστοποιητικό localhost

Τέλος πραγματοποιήθηκε η εγκατάσταση του πιστοποιητικού στον server. Αυτό έγινε με την μετατροπή του πιστοποιητικού σε localhost.pfx ώστε να περιλαμβάνει το ιδιωτικό κλειδί. Επομένως τώρα ο ιστότοπος μπορεί να υποστηρίζει το πρωτόκολλο htpps.

4.4 Παραμετροποίηση του σαρωτή Nessus

Όπως παρουσιάστηκε και στο προηγούμενο κεφάλαιο, ο σαρωτής Nessus δίνει πολλές δυνατότητες στη διαμόρφωση των χαρακτηριστικών των διενεργούμενων ελέγχων στις διαδικτυακές εφαρμογές. Στην παρούσα εργασία θα διαμορφωθούν δύο διαφορετικές πολιτικές ελέγχου όπου και θα χρησιμοποιηθούν για τον έλεγχο της εφαρμογής. Η πρώτη θα είναι μια πολιτική σάρωσης για διαδικτυακές εφαρμογές χωρίς αυθεντικοποίηση και η δεύτερη θα είναι με αυθεντικοποίηση.

4.4.1 Διαμόρφωση πολιτικής σάρωσης χωρίς αυθεντικοποίηση

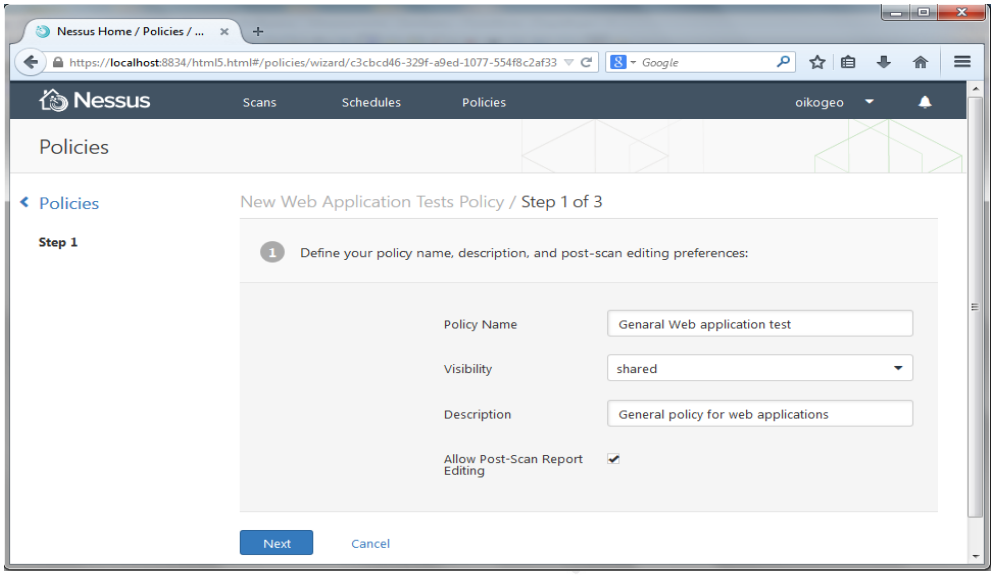
Η διαμόρφωση της συγκεκριμένης πολιτικής δίνει την δυνατότητα ελέγχου της εφαρμογής σε πρώτο στάδιο χωρίς να μπορεί να ελέγξει το σύνολό των λειτουργιών της. Με τον τρόπο αυτό μπορεί να διαπιστωθεί η πιθανή ύπαρξη ευπαθειών και με βάση τα αρχικά αποτελέσματα να διαμορφωθεί μια πιο εξειδικευμένη πολιτική σάρωσης που με την χρήση των διαπιστευτηρίων της αυθεντικοποίησης θα ελέγξει τις επιπλέον λειτουργίες της εφαρμογής.

Διαμόρφωση πολιτικής

Εφόσον γίνει η σύνδεση στο Nessus με τον λογαριασμό που έχει δημιουργηθεί, γίνεται επιλογή της καρτέλας Policies και στην συνέχεια New Policy όπως παρουσιάστηκε στο προηγούμενο κεφάλαιο. Γίνεται επιλογή του Web Application Tests από τις επιλογές που δίνει.

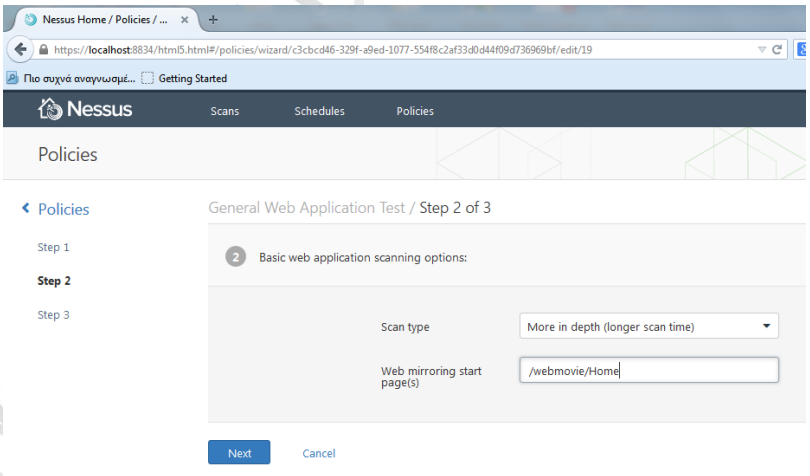
Εικόνα 41. Επιλογή δημιουργίας πολιτικής σάρωσης με βάση το Web Application Tests

Δίνεται η ονομασία General Web Application Test στην πολιτική και ως περιγραφή η ονομασία «General policy for web applications».



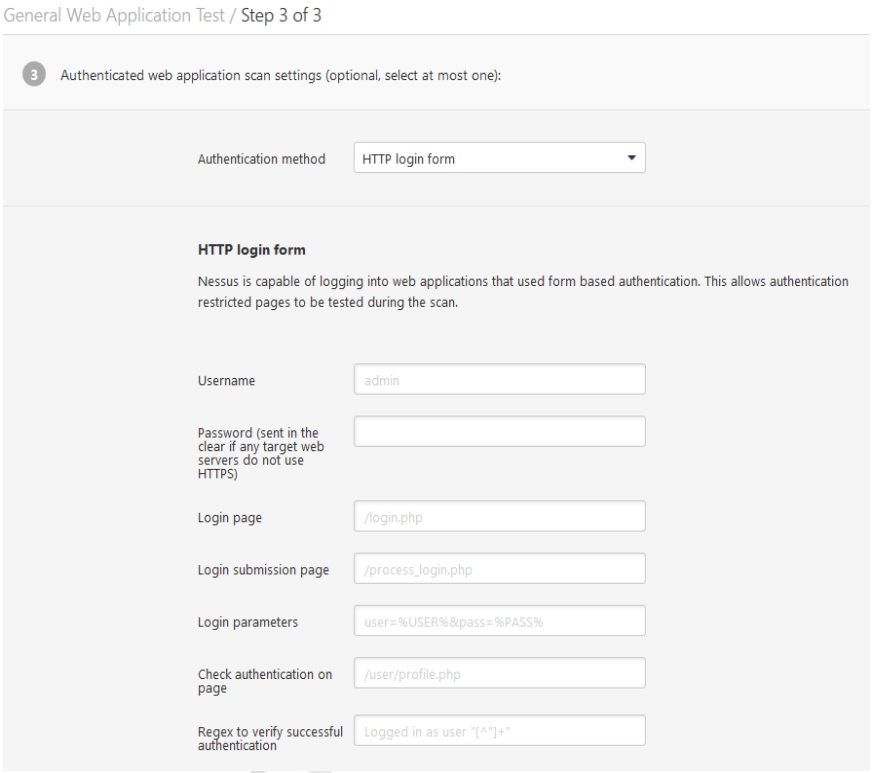
Εικόνα 42. Δημιουργία πολιτικής σάρωσης Step 1

Στο βήμα 2 η επιλογή στο Scan type είναι το More in depth (longer scan time) και στο Web mirroring start pages γίνεται εισαγωγή της webmovie/Home.



Εικόνα 43. Δημιουργία πολιτικής Step 2

Στο βήμα 3 στη μέθοδο αυθεντικοποίησης (Authenticated method), γίνεται η επιλογή της HTTP login form και προς το παρόν δεν θα γίνει η συμπλήρωση των στοιχείων της φόρμας αφού η διενέργεια του πρώτου ελέγχου της εφαρμογής θα πραγματοποιηθεί χωρίς αυθεντικοποίηση.

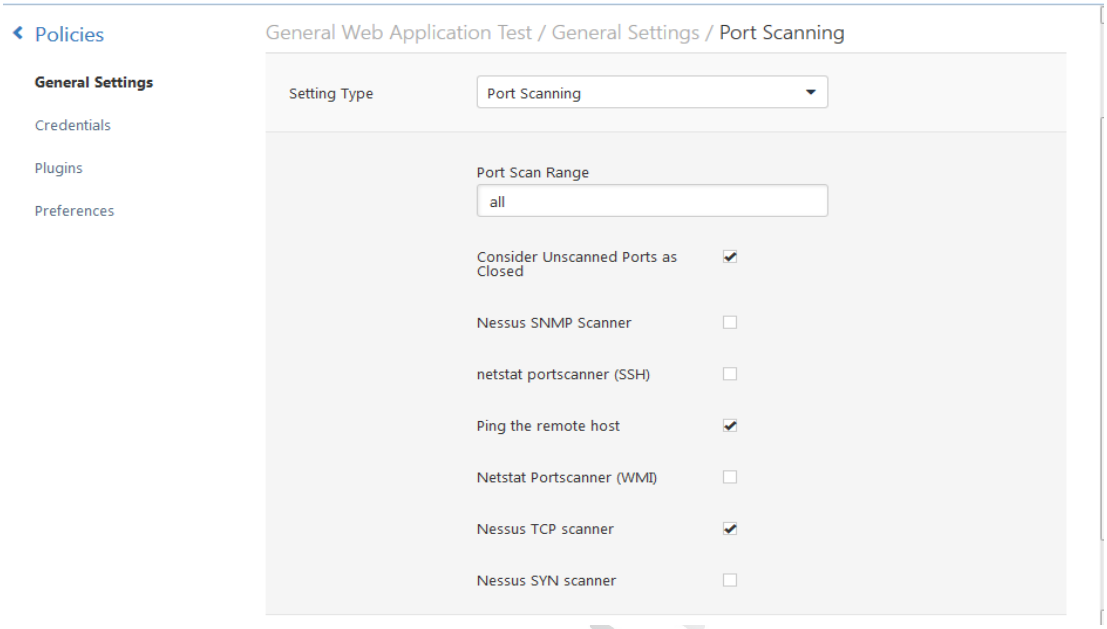


Εικόνα 44. Δημιουργία πολιτικής σάρωσης Step 3

Διαμόρφωση καρτέλας General Settings

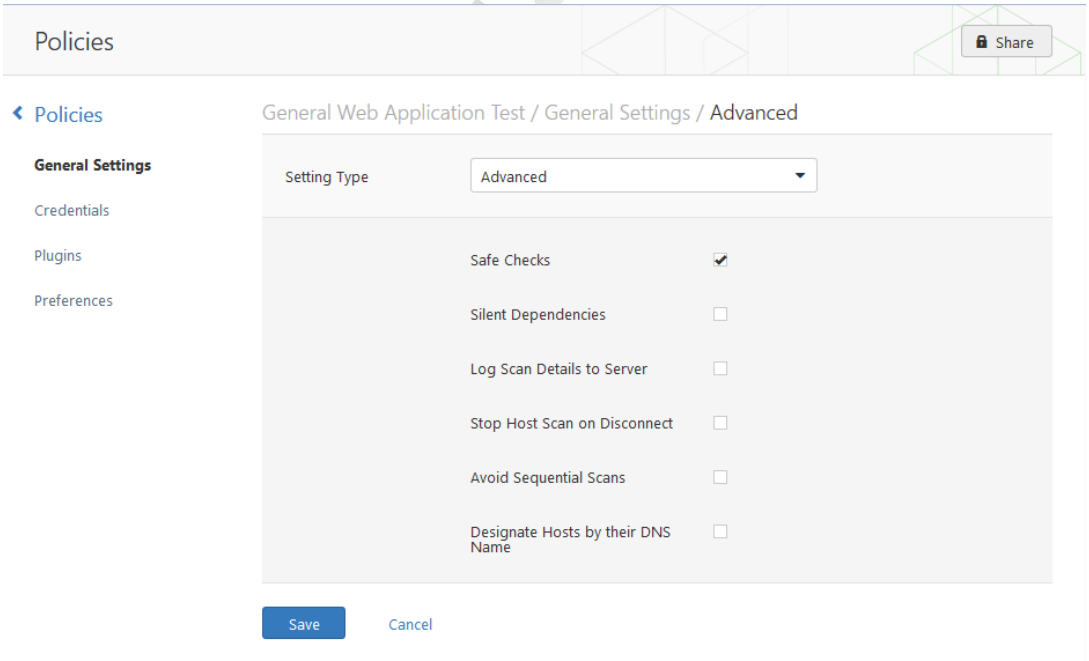
Με αυτό τον τρόπο δημιουργήθηκε η πολιτική σάρωσης και τώρα με την χρήση της επιλογής «Advanced Mode» θα γίνει η παραμετροποίηση της. Στις General Settings και στην συνέχεια στο Setting type γίνεται επιλογή του Port Scanning. Στο Port Scan Range εισάγεται η τιμή all και στη συνέχεια επιλέγονται τα παρακάτω :

- Consider Unscanned Ports as Closed,
- Ping the remote host
- Nessus TCP scanner.



Εικόνα 45. Διαμόρφωση General Settings\Port Scanning πολιτικής σάρωσης

Έπειτα πάλι στο Setting type επιλέγονται τα Advanced και Safe Checks όπου και τελειώνουν οι ρυθμίσεις όσον αφορά το Setting type.

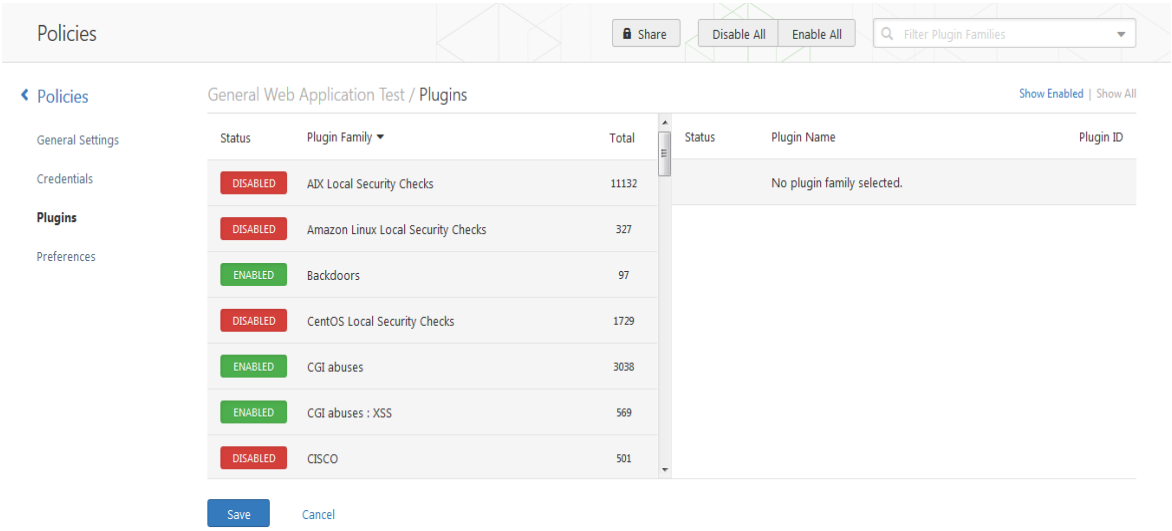


Εικόνα 46. Διαμόρφωση General Settings\Advanced πολιτικής σάρωσης

Διαμόρφωση καρτέλας Plugins

Οι επόμενες ρυθμίσεις αφορούν τις οικογένειες των plugins [37] που θα ενεργοποιηθούν για να χρησιμοποιηθούν στον έλεγχο. Θα απενεργοποιηθούν όλα και θα επιλεχθούν μόνο τα παρακάτω:

- Backdoors – Αυτά τα Plug-ins ανιχνεύουν κερκόπορτες (backdoors) υψηλού προφίλ όπως προγράμματα Δούρειοι Ίπποι (Trojan Horse), μεταδόσεις ελίκων (Worm infections), και συστήματα με σημάδια ότι έχουν παραβιαστεί.
- CGI abuses - Αυτά τα Plug-ins περιέχουν ελέγχους για ευπάθειες τύπου SQL injection, Local File Inclusion (LFI), Remote File Inclusion (RFI), Directory Traversal, και κάποιους ακόμα. Δεν περιέχουν ελέγχους για ευπάθειες cross-site scripting (XSS).
- CGI abuses : XSS - Αυτά τα Plug-ins περιέχουν ελέγχους για cross-site scripting (XSS) ευπάθειες.
- Databases - Αυτά τα Plug-ins ελέγχουν την παρουσία ευπαθειών στο λογισμικό των βάσεων δεδομένων όπως είναι της IBM DB2, του Microsoft SQL Server, της MySQL, της Oracle Database, της PostgreSQL, και κάποιων άλλων.
- Gain a shell remotely - Αυτά τα Plug-ins ελέγχουν το μεγαλύτερο ποσοστό των λογισμικών για την παρουσία ευπαθειών που επιτρέπουν την εκτέλεση εντολών (command execution) ή απομακρυσμένου κώδικα (remote code).
- General – Μια σειρά από ελέγχους που συγκεντρώνουν πληροφορίες για το απομακρυσμένο σύστημα όπως είναι το λειτουργικό σύστημα και το επίπεδο δικτύου.
- Misc. - Αυτά τα Plug-ins ελέγχουν μια μεγάλη ποικιλία λογισμικού συμπεριλαμβανομένων θέματα της πλευράς του πελάτη (client-side) και του server.
- Service detection – Έλεγχοι ασφαλείας που επιτρέπουν στο Nessus να ανιχνεύει μια μεγάλη ποικιλία υπηρεσιών στον απομακρυσμένο στόχο.
- Settings - Αυτά τα Plug-ins ελέγχουν την συμπεριφορά του Nessus κατά την διάρκεια της σάρωσης.
- Web Servers - Αυτά τα Plug-ins ελέγχουν για ευπάθειες σε web servers όπως ο Apache HTTP Server, ο IBM Lotus Domino, ο Microsoft IIS, και για αρκετούς άλλους.
- Windows – Έλεγχοι για λογισμικό που έχει εγκατασταθεί σε συστήματα Microsoft Windows συμπεριλαμβανομένων των Adobe Reader, Adobe Flash, Antivirus software, web browsers, iTunes, και αρκετά ακόμα.
- Windows : Microsoft Bulletins - Έλεγχοι ασφαλείας που δοκιμάζουν συστήματα Microsoft Windows τοπικά αν τα διαπιστευτήρια αυθεντικοποίησης παρέχονται στο Nessus.

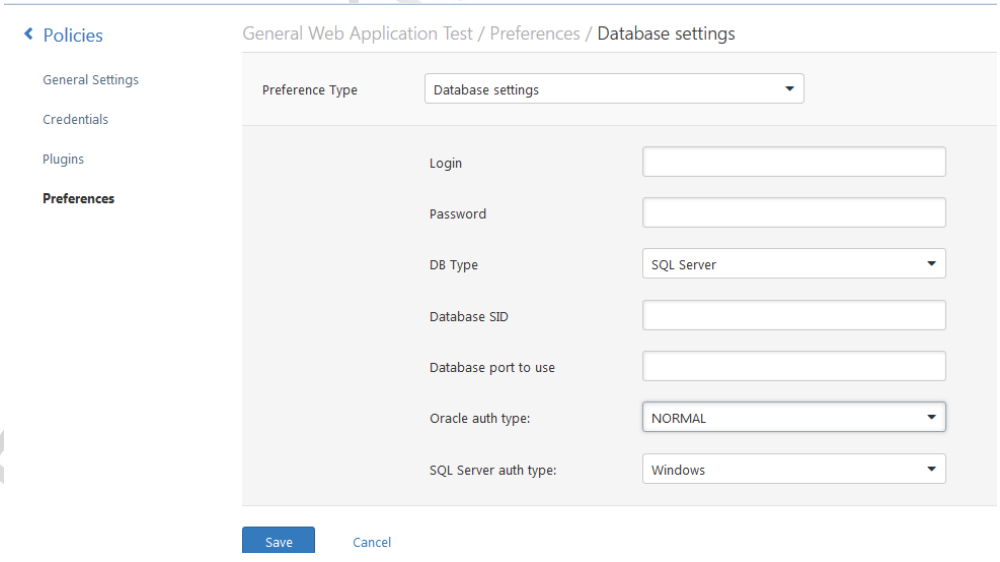


Εικόνα 47: Επιλογή Plugins πολιτικής σάρωσης

Διαμόρφωση καρτέλας Preferences

Οι τελευταίες ρυθμίσεις για να ολοκληρωθεί η διαδικασία θα γίνουν στην καρτέλα Preferences. Αρχικά από την επιλογή Preference Type θα επιλεχθεί η καρτέλα Database settings και θα γίνουν οι εξής ρυθμίσεις:

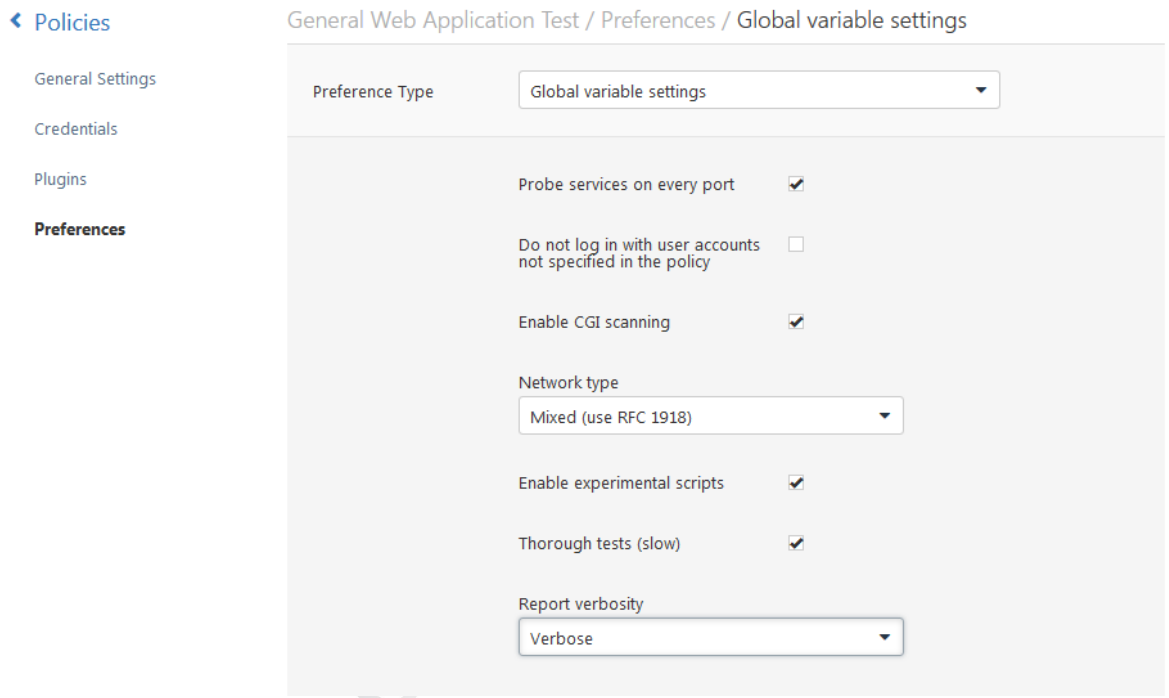
- DB Type = SQL Server
- SQL Server auth type = Windows



Εικόνα 48. Διαμόρφωση Preferences\Database Settings πολιτικής σάρωσης

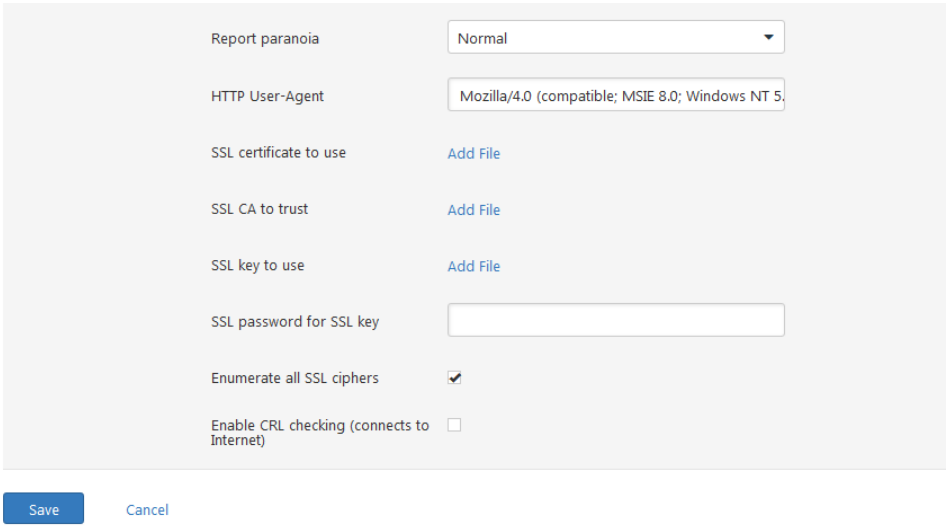
Στην καρτέλα Global Variable Settings οι ρυθμίσεις είναι οι παρακάτω :

- Probe Services on every port
- Enable CGI scanning
- Network Type (Mixed use (RFC 1918) από το μενού αναδίπλωσης)
- Enable experimental scripts
- Thorough tests (slow)
- Report verbosity (verbose από το μενού αναδίπλωσης)



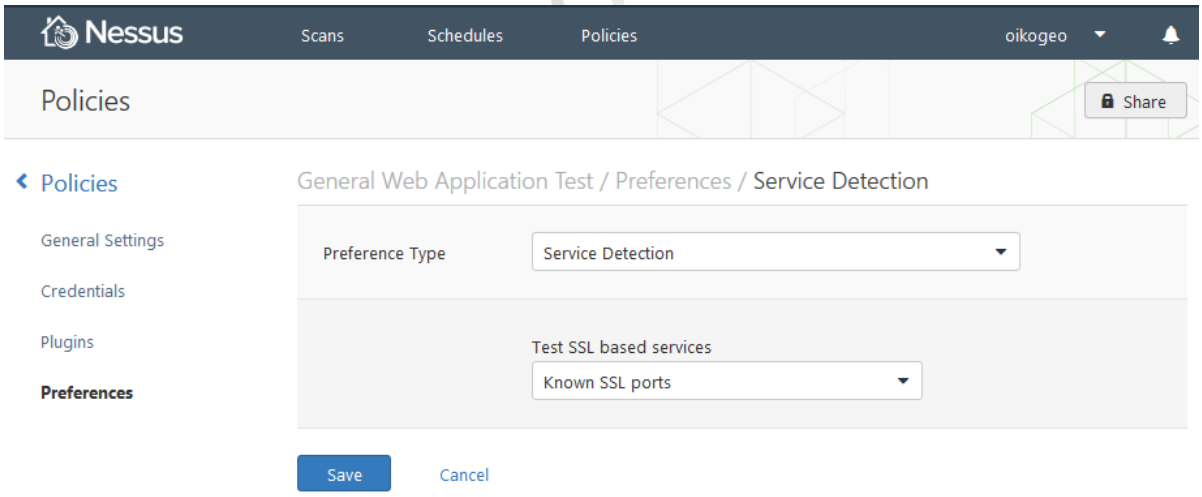
Εικόνα 49. Διαμόρφωση Preferences\Global Variable settings πολιτικής σάρωσης

- Report paranoia (normal από το μενού αναδίπλωσης)
- HTTP User-Agent Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0)
- Enumerate all SSL ciphers



Εικόνα 50. Διαμόρφωση Preferences\Database Settings πολιτικής σάρωσης

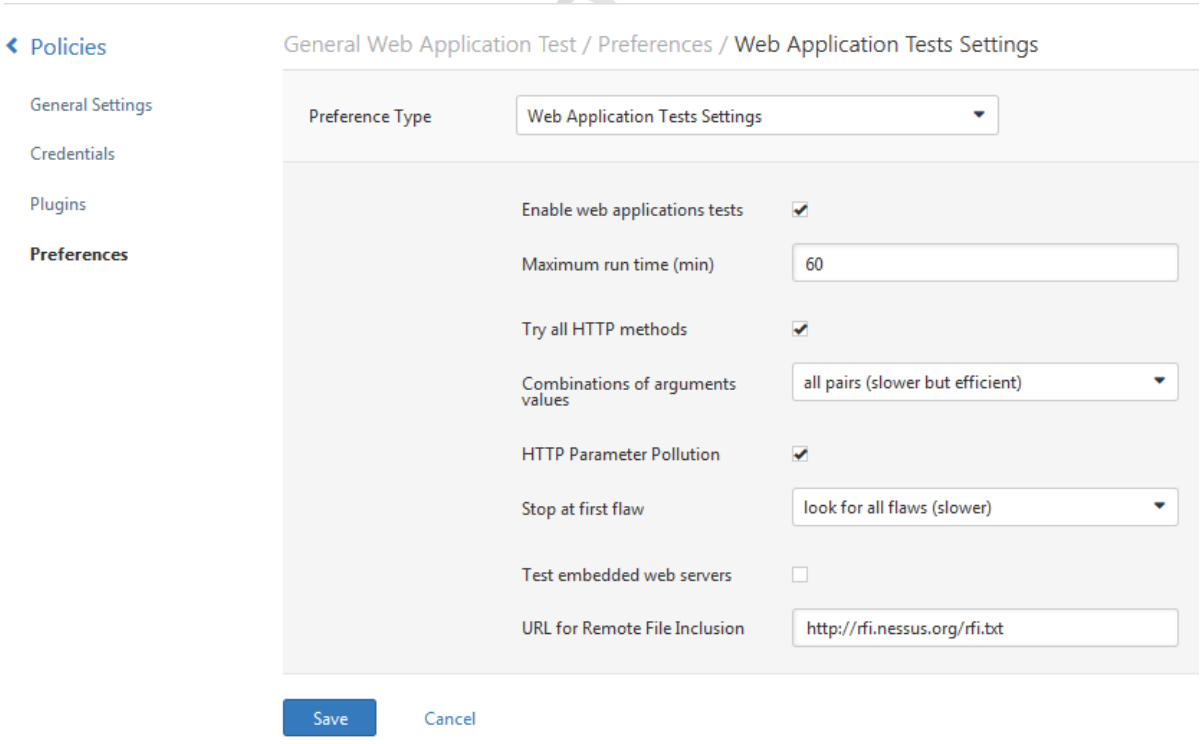
Στην καρτέλα Service Detection για την επιλογή Test SSL based services θα επιλεγεί το Known SSL ports από το μενού αναδίπλωσης.



Εικόνα 51. Διαμόρφωση Preferences\Service Detection πολιτικής σάρωσης

Στην καρτέλα Web Application Tests Settings οι ρυθμίσεις [38] είναι οι παρακάτω:

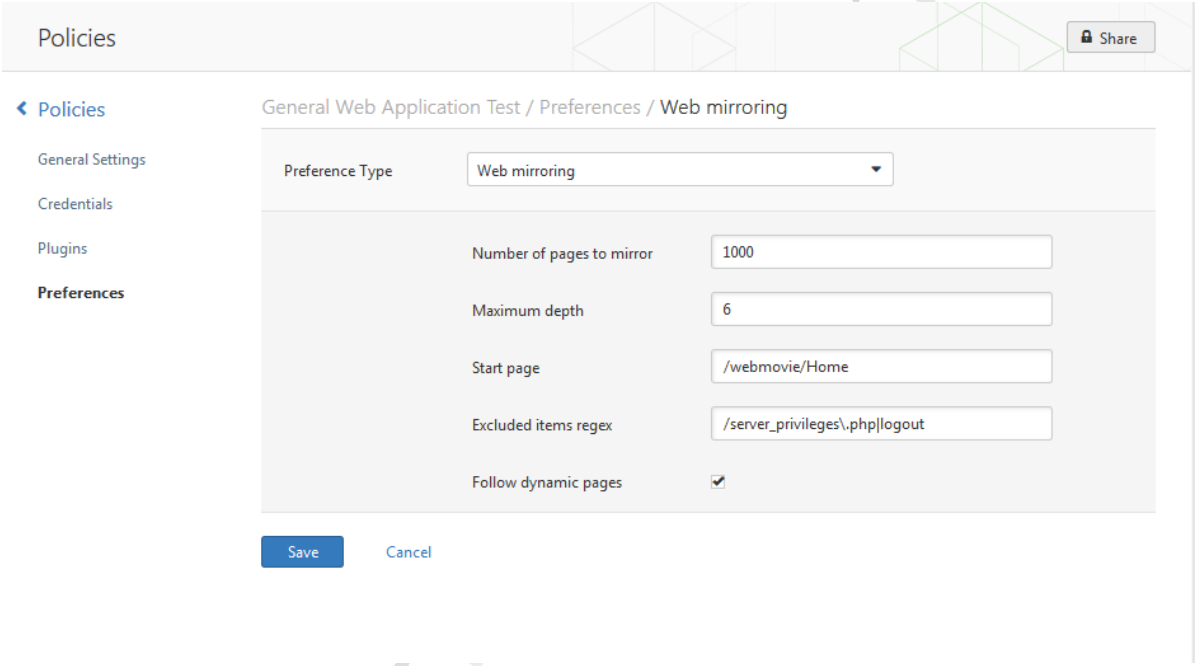
- Enable web applications tests – Με αυτή την επιλογή ενεργοποιείται ο έλεγχος με όλα τα plug-ins που αναφέρθηκαν στο Κεφάλαιο 3.5.4
- Maximum run time (min) 60 – Ορίζει το μέγιστο χρόνο εκτέλεσης για κάθε plug-in που στην συγκεκριμένη περίπτωση είναι 60 λεπτά.
- Try all HTTP methods – Εξασφαλίζει τον έλεγχο της εφαρμογής τόσο με την μέθοδο GET όσο και με την μέθοδο POST. Σε αντίθετη περίπτωση θα γινόταν μόνο ο έλεγχος GET.
- Combinations of arguments values (all pairs (slower but efficient) από το μενού αναδίπλωσης) – Γίνεται έλεγχος σε όλους τους συνδυασμούς αλφαριθμητικών επίθεσης και έγκυρων δεδομένων της εφαρμογής.
- HTTP Parameter Pollution – Διεξάγει ελέγχους στα δεδομένα κάνοντας επαύξηση αυτών με έγκυρα δεδομένα της εφαρμογής.
- Stop at first flaw (look for all flaws (slower) από το μενού αναδίπλωσης) – Διεξάγει ελέγχους για κάθε παράμετρο που θα ανακαλυφθεί.
- URL for Remote File Inclusion http://rfi.nessus.org/rfi.txt - Ορίζει το αρχείο που θα χρησιμοποιηθεί για τους ελέγχους κατά την διάρκεια του ελέγχου RFI.



Εικόνα 52. Διαμόρφωση Preferences\Web Application Tests Settings πολιτικής σάρωσης

Στην καρτέλα Web mirroring οι ρυθμίσεις είναι οι παρακάτω :

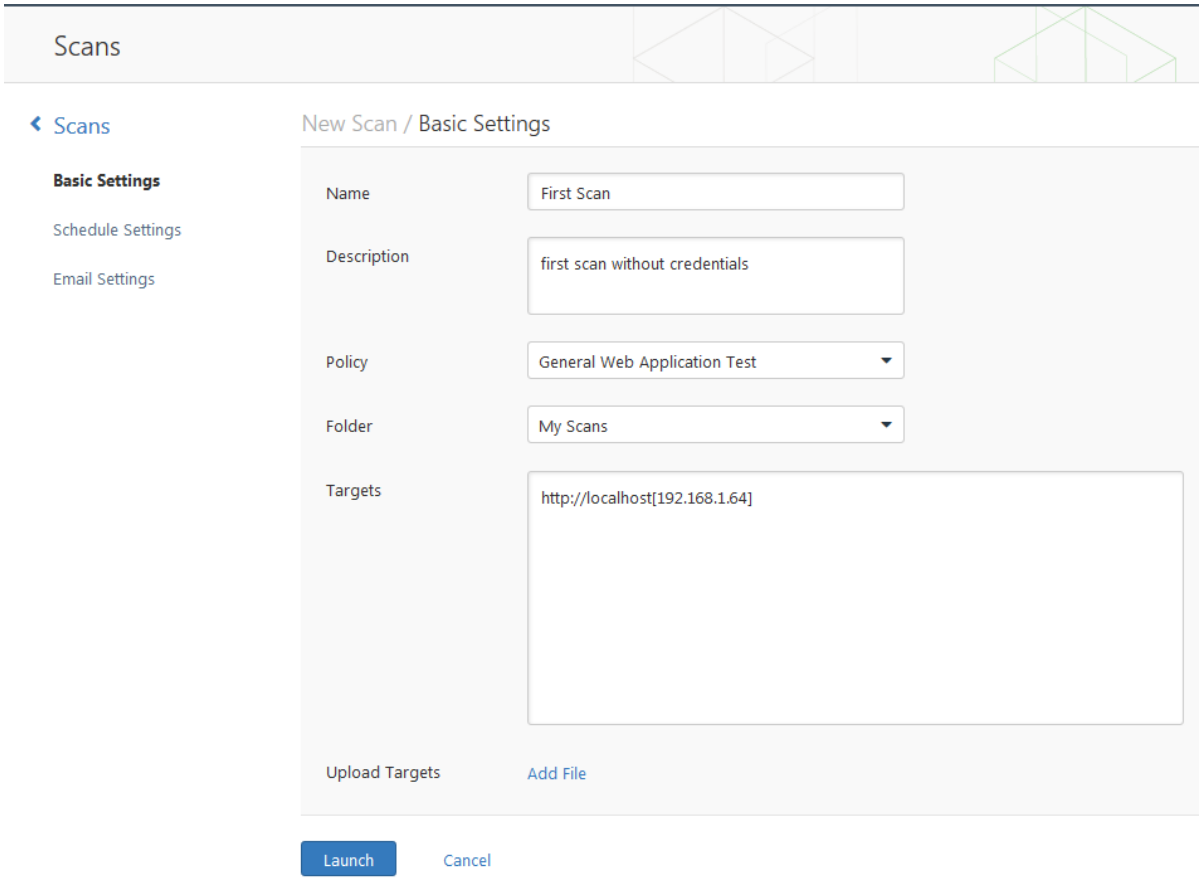
- Number of pages to mirror (1000)
- Maximum depth (6)
- Start page /webmovie/Home
- Excluded items regex /server_privileges\.php|logout
- Follow dynamic pages



Εικόνα 53. Διαμόρφωση Preferences\Web mirroring πολιτικής σάρωσης

Η πολιτική πια έχει διαμορφωθεί κατάλληλα για να διεξαχθεί ο πρώτος έλεγχος στην εφαρμογή. Επομένως στην καρτέλα Scans γίνεται η επιλογή New Scan.

Δίνεται η ονομασία First Scan και στην περιγραφή η ονομασία first scan without credentials. Γίνεται η επιλογή της πολιτικής που μόλις δημιουργήθηκε (General Web Application Test), γίνεται η εισαγωγή στο στόχο (Targets) του υπολογιστή που τρέχει σαν server την εφαρμογή με την ip διεύθυνση του (http://localhost/webmovie[192.168.1.64]).

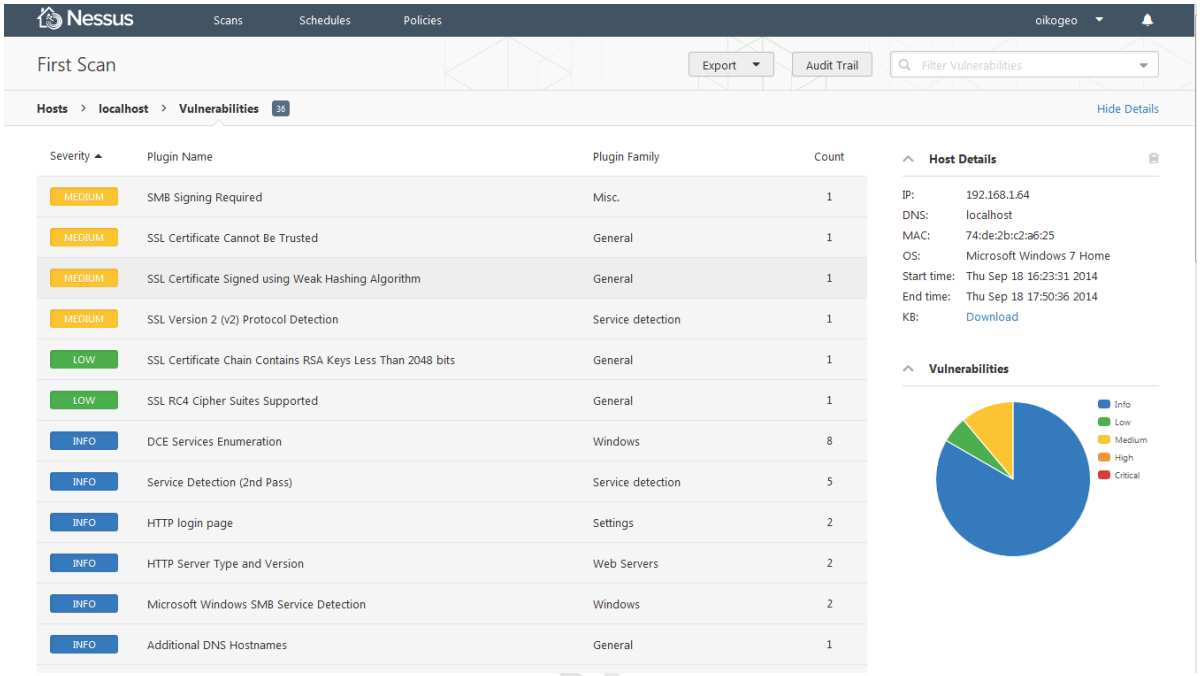


Εικόνα 54. Διαμόρφωση καρτέλας Scans πολιτικής σάρωσης

Γίνεται η επιλογή «Launch» και ξεκινάει ο έλεγχος.

Εικόνα 55. Έναρξη ελέγχου εφαρμογής

Ο έλεγχος μετά τον απαιτούμενο ολοκληρώθηκε και παρουσιάζεται η αναφορά παρακάτω.



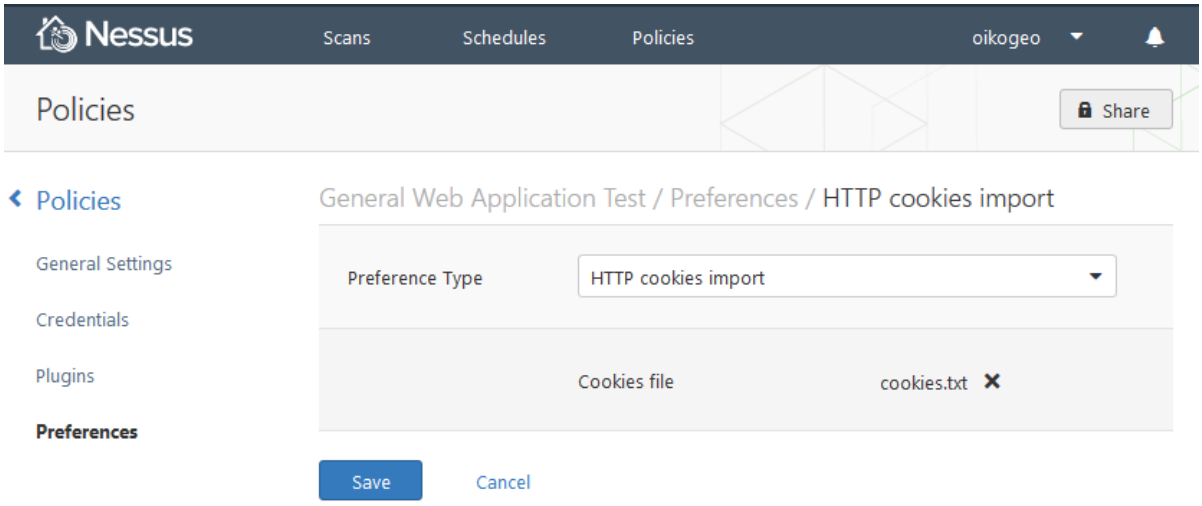
Εικόνα 56. Αναφορά ελέγχου εφαρμογής με τη χρήση πολιτικής σάρωσης χωρίς αυθεντικοποίηση

4.4.2 Διαμόρφωση πολιτικής σάρωσης με αυθεντικοποίηση

Στη δεύτερη φάση του ελέγχου της εφαρμογής θα γίνει αναδιαμόρφωση της πολιτικής ελέγχου που δημιουργήθηκε προηγουμένως ώστε το Nessus να μπορεί να αυθεντικοποιηθεί σαν χρήστης στην εφαρμογή και να την σαρώσει εξ ολοκλήρου. Όπως παρουσιάζεται και παρακάτω όλες οι απαιτούμενες ρυθμίσεις θα γίνουν στην καρτέλα Preferences.

Διαμόρφωση καρτέλας Preferences

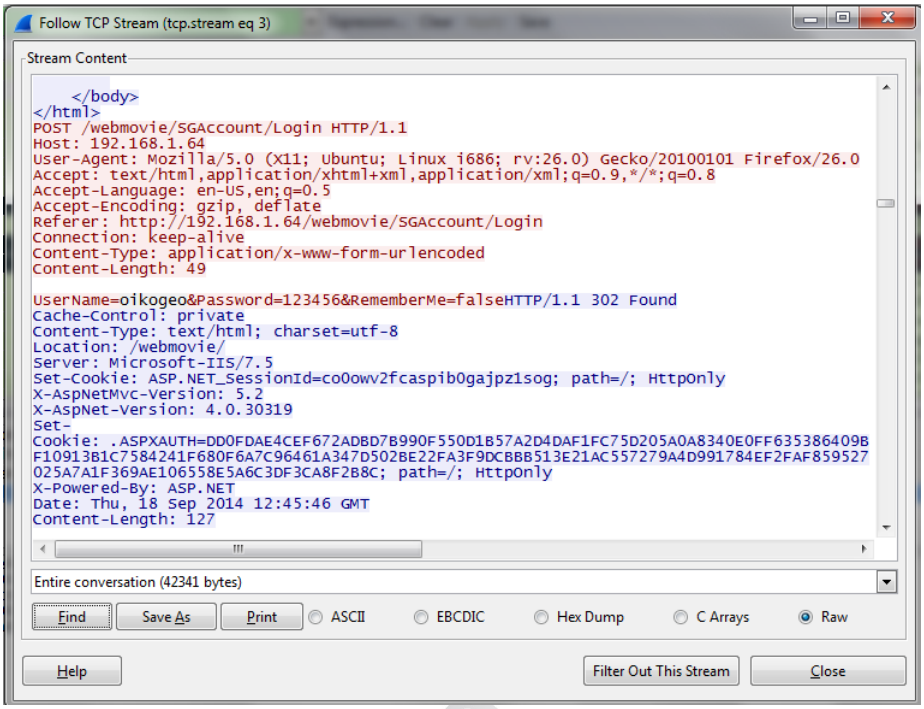
Αρχικά γίνεται επιλογή της πολιτικής General Web Application Test και της καρτέλας Preferences. Από την επιλογή Preference Type επιλέγεται η καρτέλα HTTP cookies import και κάνοντας add file ορίζεται το αρχείο cookies.txt να είναι το αρχείο που θα βρίσκει το Nessus τα cookies που χρειάζεται κατά την διάρκεια της αυθεντικοποίησης του. Το αρχείο cookies.txt δημιουργήθηκε από την εξαγωγή των cookies από τον browser αφού πρώτα είχε γίνει χρήση της εφαρμογής σαν αυθεντικοποιημένοι χρήστες.



Εικόνα 57. Διαμόρφωση Preferences\HTTP cookies import πολιτικής σάρωσης με αυθεντικοποίηση

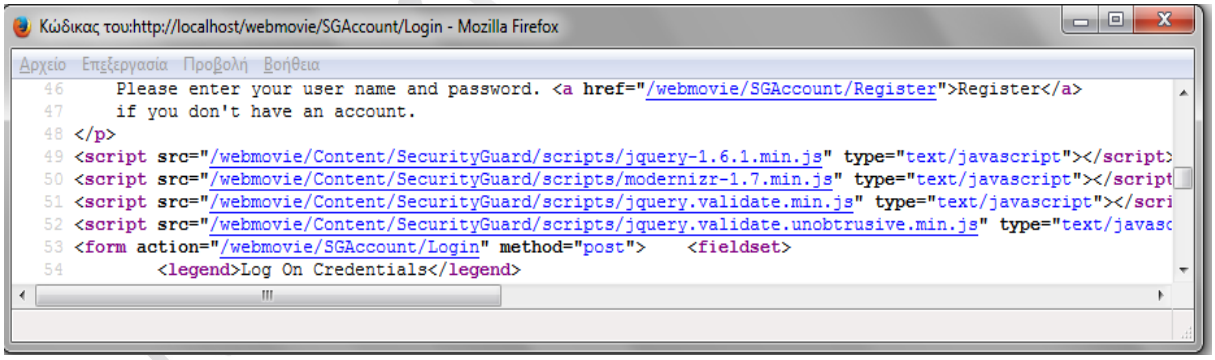
Στο επόμενο βήμα επιλέγεται η καρτέλα HTTP login page και γίνονται οι παρακάτω ρυθμίσεις:

- Login page – εισαγωγή της σελίδας που γίνεται η αυθεντικοποίηση των χρηστών στην εφαρμογή (/webmovie/SGAccount/Login).
- Login form – εισαγωγή της σελίδας που βρίσκεται η φόρμα αυθεντικοποίησης (/webmovie/SGAccount/Login).
- Login form fields – εισαγωγή της συμβολοσειράς που χρησιμοποιεί η εφαρμογή όταν ο χρήστης έχει συμπληρώσει τα στοιχεία του (username,password), η οποία εντοπίστηκε με την χρήση του εργαλείου Wireshark [39]
- (UserName=%USER%&Password=%PASS%&RememberMe=false).



Εικόνα 58. Συμβολοσειρά αυθεντικοποίησης από το στιγμιότυπο του Wireshark

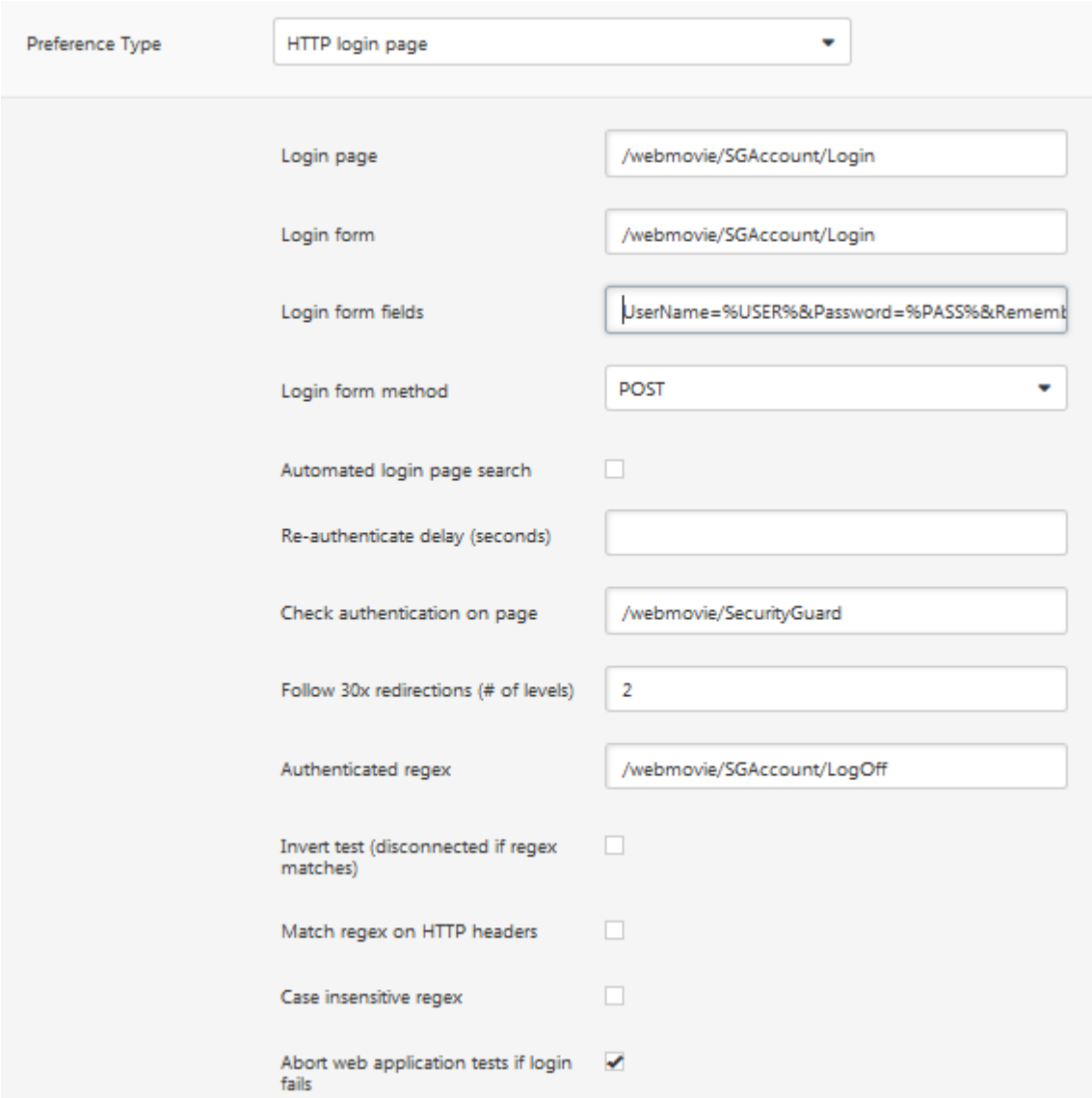
- Login form method- γίνεται επιλογή του POST για τη μέθοδο αυθεντικοποίησης το οποίο βρέθηκε από τον έλεγχο του πηγαίου κώδικα.



Εικόνα 59. Έυρεση μεθόδου Post στον πηγαίο κώδικα

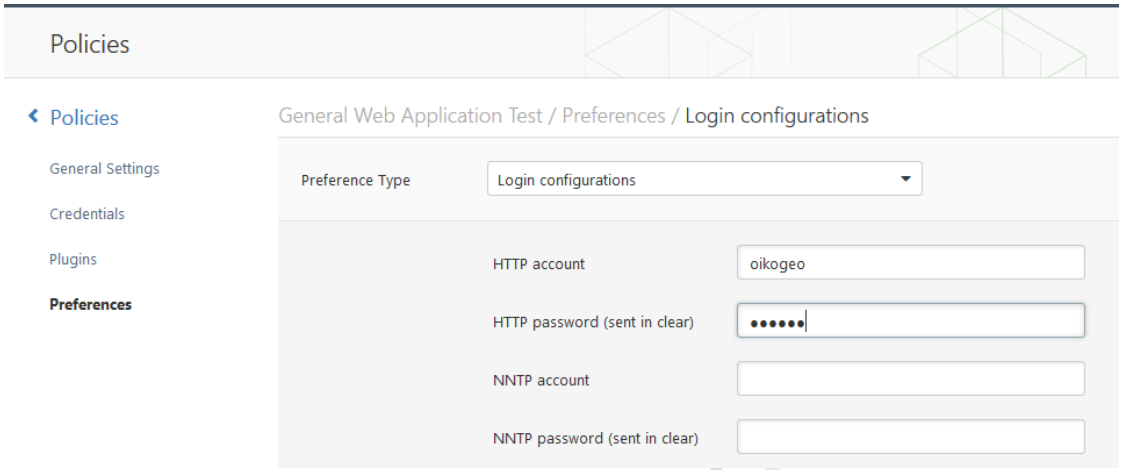
- Check authentication on page - εισαγωγή της σελίδας που ελέγχει το Nessus αν εξακολουθεί να είναι αυθεντικοποιημένο (/webmovie/SecurityGuard)
- Authenticated regex – εισαγωγή ενός χαρακτηριστικού που θα αναζητήσει το Nessus μετά την αυθεντικοποιήση για επιβεβαίωση (/webmovie/SGAccount/LogOff)

- Abort web application tests if login fails – επιλογή να σταματήσει τον έλεγχο της εφαρμογής αν αποτύχει η αυθεντικοποίηση.



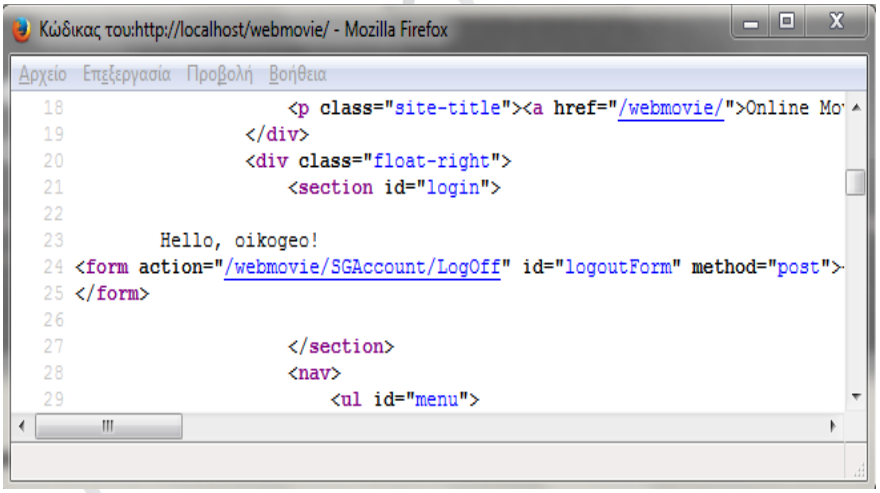
Εικόνα 60. Διαμόρφωση Preferences\HTTP login page πολιτικής σάρωσης με αυθεντικοποίηση

Στην καρτέλα Login configurations θα γίνει εισαγωγή στο HTTP account το username που χρησιμοποιείται για την εφαρμογή (oikogeo) και στο HTTP password (sent in clear) το password που χρησιμοποιείται (123456), τα οποία θα αντικαταστήσουν αντίστοιχα το %USER% και το %PASS% που εμπεριέχονται στην συμβολοσειρά αυθεντικοποίησης της προηγούμενης καρτέλας.

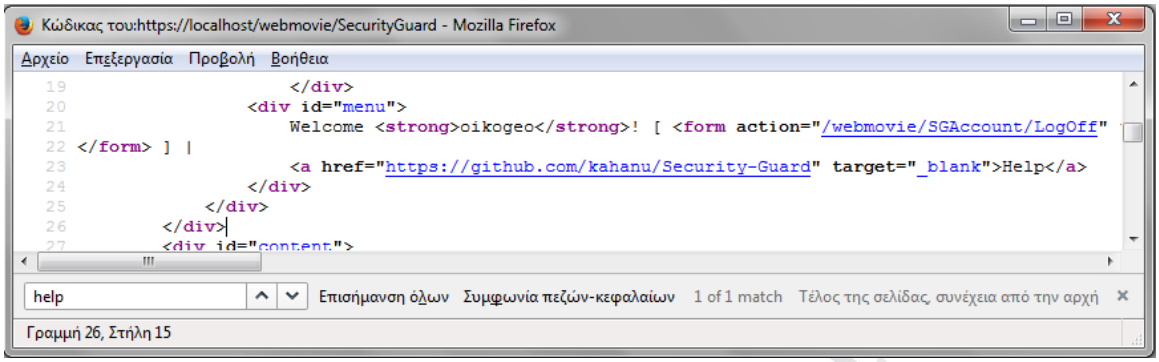


Εικόνα 61. Διαμόρφωση Preferences\Login configurations πολιτικής σάρωσης με αυθεντικοποίηση

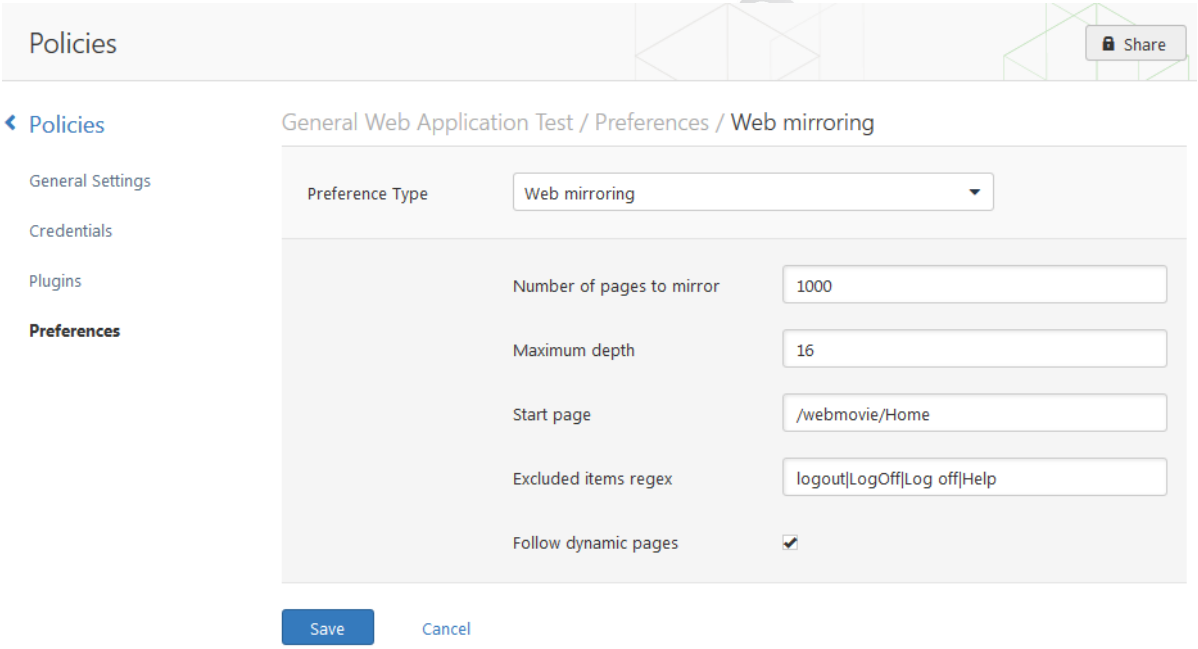
Στην καρτέλα Web mirroring , στο πεδίο Excluded items regex θα προστεθούν οι λέξεις LogOff και Help. Με αυτό τον τρόπο πληροφορείται το Nessus ότι όταν συναντήσει κατά την διάρκεια εξερεύνησης των συνδέσμων της εφαρμογής να μην τις επιλέξει γιατί υπάρχει κίνδυνος εξόδου από την εφαρμογή. Αυτοί οι σύνδεσμοι εντοπίστηκαν μετά από αναζήτηση στον πηγαίο κώδικα της εφαρμογής.



Εικόνα 62. Έυρεση συνδέσμου εξόδου του χρήστη από την εφαρμογή (LogOff) στον πηγαίο κώδικα



Εικόνα 63. Εύρεση συνδέσμου εξόδου του χρήστη από την εφαρμογή (Help) στον πηγαίο κώδικα



Εικόνα 64. Διαμόρφωση Preferences\Web mirroring πολιτικής σάρωσης με αυθεντικοποίηση

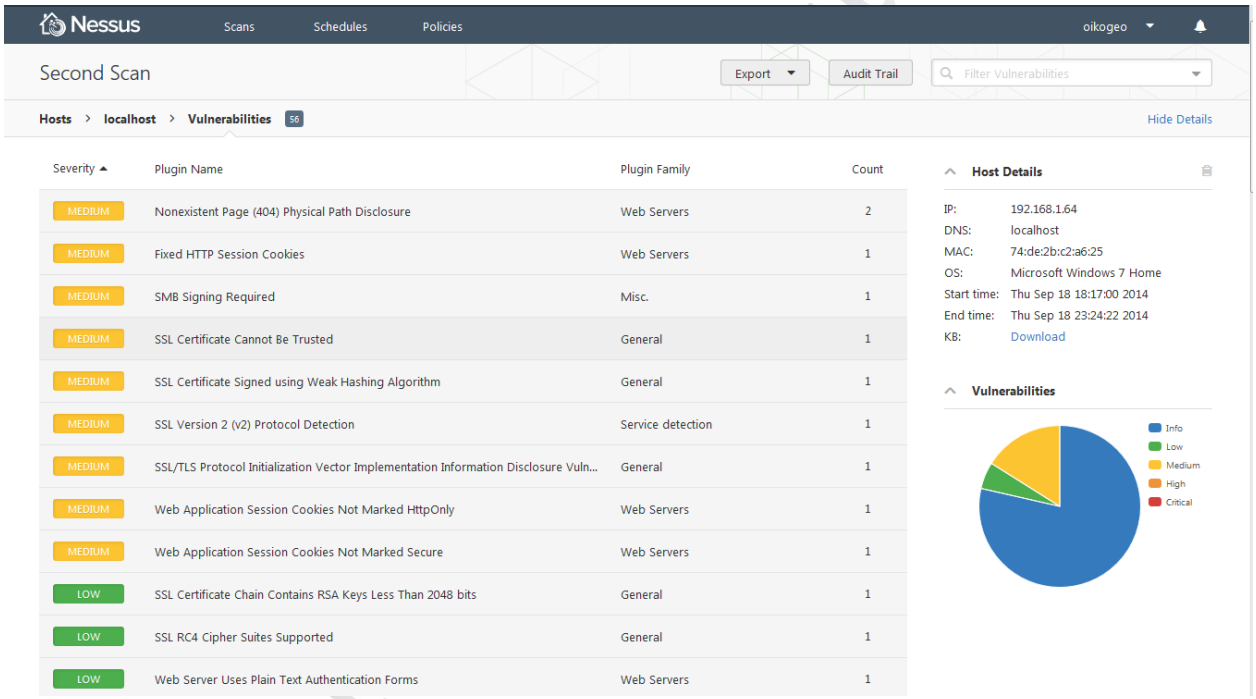
Εδώ έχει ολοκληρωθεί η διαμόρφωση των ρυθμίσεων της πολιτικής με αυθεντικοποίηση και γίνεται έλεγχος πάλι στην εφαρμογή. Μετά το τέλος του ελέγχου πρώτα διαπιστώνεται ότι πράγματι το Nessus κατάφερε να συνδεθεί στην εφαρμογή ως χρήστης όπως φαίνεται παρακάτω από την αναφορά στην Http login page (Εικόνα 65).

```
HTTP login succeeded with the following request :

GET /webmovie/SecurityGuard HTTP/1.1
Host: localhost
Accept-Charset: iso-8859-1,utf-8;q=0.9,*;q=0.1
Accept-Language: en
Connection: Keep-Alive
```

Εικόνα 65. Περιεχόμενο αναφοράς Http login page επιτυχής σύνδεσης

Η συνολική αναφορά ελέγχου είναι αυτή που παρουσιάζεται παρακάτω.



Εικόνα 66. Αναφορά ελέγχου εφαρμογής με τη χρήση πολιτικής σάρωσης με αυθεντικοποίηση

Κεφάλαιο 5°

5 Ανάλυση Ευπαθειών της Διαδικτυακής Εφαρμογής

5.1 Εισαγωγή

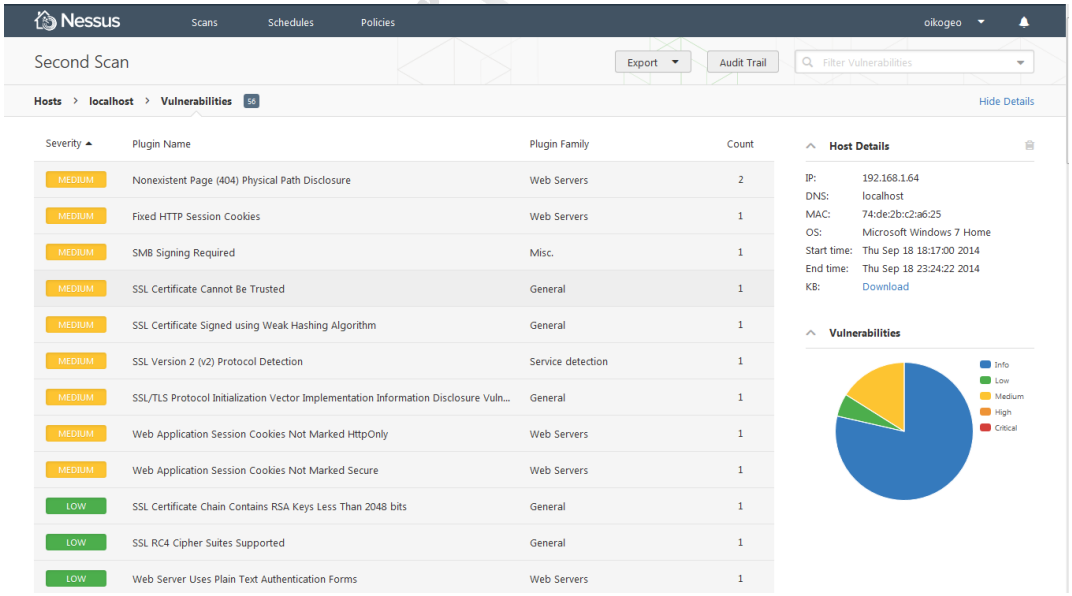
Εφόσον οι έλεγχοι της εφαρμογής ολοκληρώθηκαν κανονικά και δημιουργήθηκαν οι αναφορές από το Nessus, το επόμενο στάδιο είναι η ανάλυση των ευπαθειών της εφαρμογής. Οι αναφορές του Nessus είναι πολύ χρήσιμες για την ανάλυση αφού περιέχουν αναλυτική περιγραφή για κάθε μία ευπάθεια και ενδεικτικούς τρόπους αντιμετώπισης της.

Η σύγκριση των δυο αναφορών που εξήχθησαν από τους ελέγχους μας οδηγεί στη διαπίστωση ότι οι ευπάθειες που εντοπίστηκαν στον πρώτο έλεγχο, ο οποίος έγινε χωρίς αυθεντικοποίηση, συμπεριλαμβάνονται ως μέρος του συνόλου των ευπαθειών του δευτέρου ελέγχου. Αυτό οφείλεται στο γεγονός ότι το Nessus με την χρήση των διαπιστευτηρίων αυθεντικοποίησης, απέκτησε δυνατότητα πρόσβασης για έλεγχο σε όλες τις ιστοσελίδες της εφαρμογής και γενικά στο σύνολο της. Επομένως η ανάλυση των ευπαθειών θα αφορά αποκλειστικά τα ευρήματα της αναφοράς του δευτέρου ελέγχου.

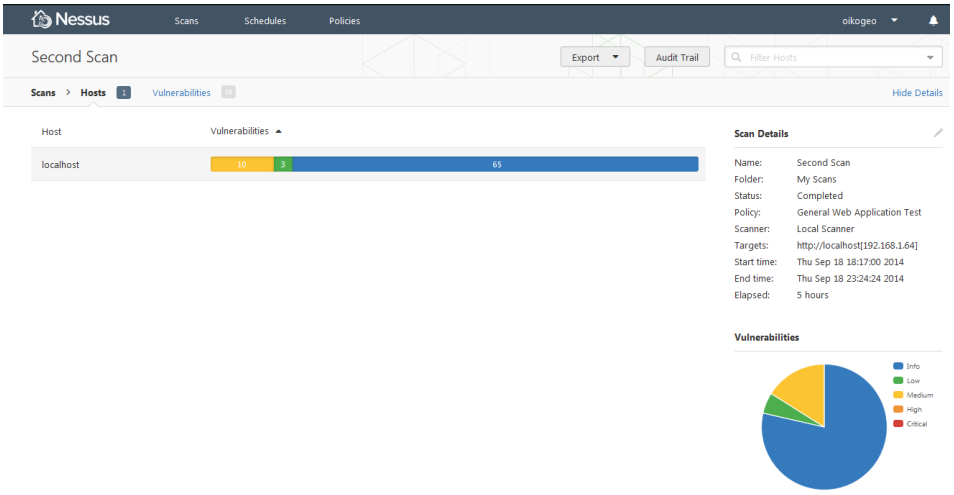
5.2 Παρουσίαση αναφοράς ελέγχου

Η αναφορά του ελέγχου που θα αναλυθεί, περιέχει 56 κατηγορίες ευπαθειών (vulnerabilities) εκ των οποίων οι 9 είναι μεσαίου επιπέδου κινδύνου, οι 3 χαμηλού και οι υπόλοιπες 44 είναι ενημερωτικού επιπέδου. Στο πρώτο επίπεδο αντιστοιχούν 10 ευπάθειες, στο δεύτερο 3 και στο τρίτο 65.

Αυτό φαίνεται στις παρακάτω εικόνες των αναφορών όπου στην πρώτη εικόνα (Εικ.67) φαίνεται ο αριθμός του συνόλου των 56 κατηγοριών και στην δεύτερη εικόνα (Εικ.68) φαίνεται το σύνολο των ευπαθειών ανά επίπεδο κινδύνου.



Εικόνα 67. Κατηγοριών ευπαθειών (Vulnerabilities 56)



Εικόνα 68. Συνολικός αριθμός ευπαθειών ανά επιπέδου κινδύνου

5.3 Ανάλυση και παρουσίαση ευπαθειών

Από τις συνολικά 78 ευπάθειες που περιέχονται στην αναφορά ελέγχου, στην παρούσα εργασία θα αναλυθούν μόνο αυτές που ανήκουν στις κατηγορίες μεσαίου και χαμηλού κινδύνου. Ομαδοποιήθηκαν οι ευπάθειες σε 4 κατηγορίες όπως παρουσιάζονται παρακάτω :

1. Αυθεντικοποίησης
2. Χρήσης SSL
3. Διαχείρισης Συνόδου
4. Στην πλευρά του εξυπηρετητή (server-side)

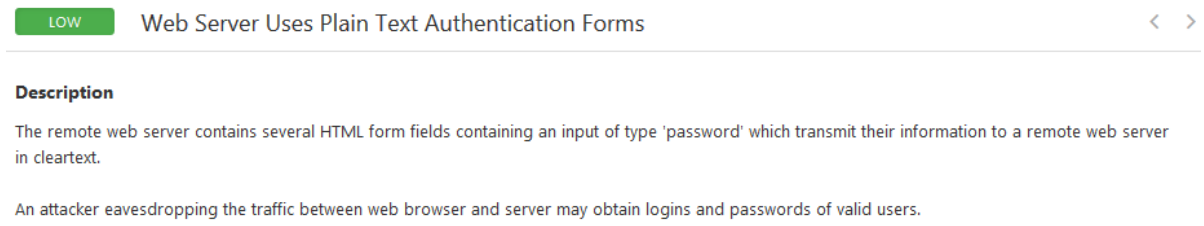
Η ανάλυση θα ξεκινήσει από τις ευπάθειες αυθεντικοποίησης και θα ολοκληρωθεί με αυτές στην πλευρά του εξυπηρετητή.

5.3.1 Ευπάθειες Αυθεντικοποίησης

Σε αυτές περιλαμβάνονται οι ευπάθειες 1) Web Server uses Plain Text Authentication Forms και 2) SMB Signing Required

1. Web Server uses Plain Text Authentication Forms

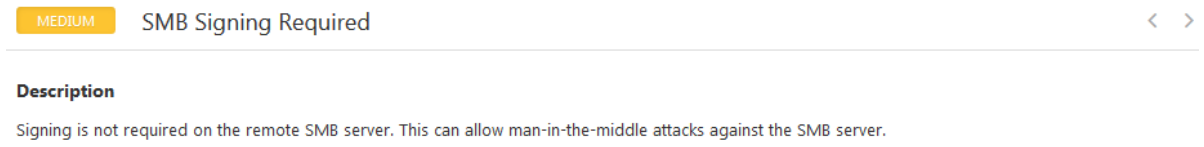
Η πρώτη ευπάθεια που αναλύεται ενημερώνει ότι ο server κατά την διαδικασία της αυθεντικοποίησης χρησιμοποιεί πεδία που περιέχουν καταχωρήσεις που αφορούν τον κωδικό πρόσβασης (password), τα οποία δεν είναι προστατευμένα και μπορεί αν κάποιος παρακολουθεί την κίνηση του δικτύου να τα υποκλέψει και να έχει την δυνατότητα πρόσβασης ως κανονικός χρήστης.



Εικόνα 69. Vulnerability low severity – Web Server Uses Plain Text Authentication Forms

2. SMB Signing Required

Η ευπάθεια αυτή ενημερώνει ότι ο SMB server δεν απαιτεί αυθεντικοποίηση κατά την χρήση του και αυτό μπορεί να επιτρέψει επιθέσεις man-in-the-middle εναντίον του.



Εικόνα 70. Vulnerability medium severity – SMB Signing Required

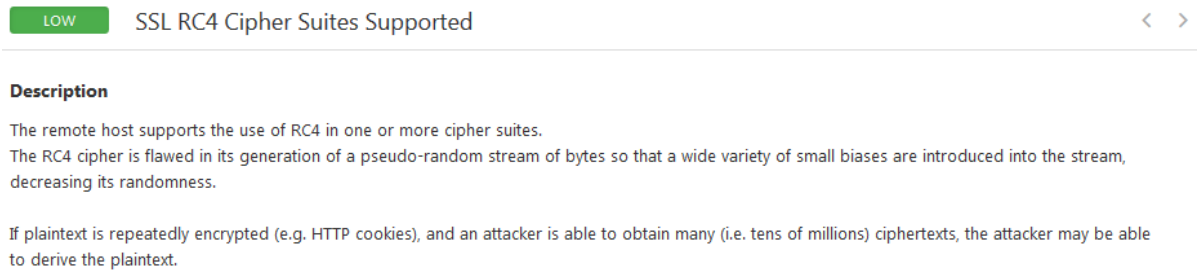
5.3.2 Ευπάθειες χρήσης SSL

Σε αυτή την κατηγορία περιλαμβάνονται οι ευπάθειες 1) SSL RC4 Cipher Suites Supported 2) SSL Certificate Chain Contains RSA Keys Less Than 2048 bits 3) SSL/TLS Protocol Initialization Vector Implementation Information Disclosure Vulnerability 4) SSL Version 2 (v2) Protocol Detection 5) SSL Certificate Signed using Weak Hashing Algorithm και 6) SSL Certificate Cannot Be Trusted.

1. SSL RC4 Cipher Suites Supported

Η συγκεκριμένη ευπάθεια ενημερώνει ότι η εφαρμογή που ελέγχθηκε χρησιμοποιεί σε κάποιες περιπτώσεις την κρυπτογράφηση RC4 για την λειτουργία του SSL. Αυτό αποτελεί κίνδυνο γιατί η κρυπτογράφηση RC4 έχει αποδειχτεί ελαττωματική κατά την δημιουργία της ψευδοτυχαίας ροής από bytes, έτσι ώστε να εισάγονται σε αυτή μια μεγάλη ποικιλία από μικρές διαστρεβλώσεις μειώνοντας έτσι την τυχαιότητά της.

Επομένως, αν το κείμενο έχει κρυπτογραφηθεί επανειλημμένα (π.χ Http cookies) και ο επιτιθέμενος μπορεί να αποκτήσει αρκετά τέτοια κρυπτογραφημένα κείμενα τότε υπάρχει η πιθανότητα να μπορεί να εξάγει το αρχικό κείμενο.

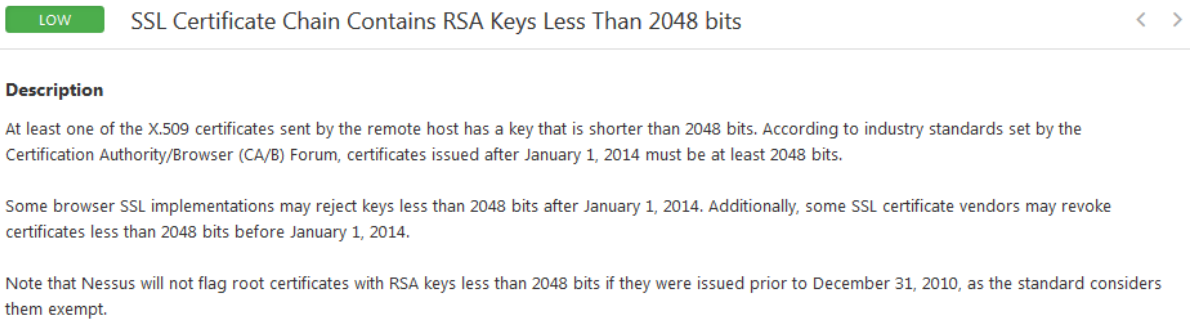


Εικόνα 71. Vulnerability low severity – SSL RC4 Cipher Suites Supported

2. SSL Certificate Chain Contains RSA Keys Less Than 2048 bits

Η ευπάθεια αυτή αναφέρεται στο πιστοποιητικό που δημιουργήσαμε για τον ιστότοπο και συγκεκριμένα στο κλειδί του που είναι μικρότερο από 2048 bits. Σύμφωνα με το Certification Authority/Browser φόρουμ τα κλειδιά των πιστοποιητικών που έχουν εκδοθεί από την 1η ιανουαρίου 2014 και μετά θα πρέπει να είναι τουλάχιστον 2048 bits.

Επίσης ενημερώνει ότι κάποιοι φυλλομετρητές στις συνδέσεις που χρησιμοποιούν SSL μπορεί να απορρίπτουν κλειδιά μικρότερα των 2048 bits μετά την 1η ιανουαρίου 2014. Επιπρόσθετα κάποιοι πάροχοι πιστοποιητικών SSL μπορεί και να αποσύρουν πιστοποιητικά με τέτοια κλειδιά.



Εικόνα 72. Vulnerability low severity – SSL Certificate Chain Contains RSA Keys Less Than 2048 bits

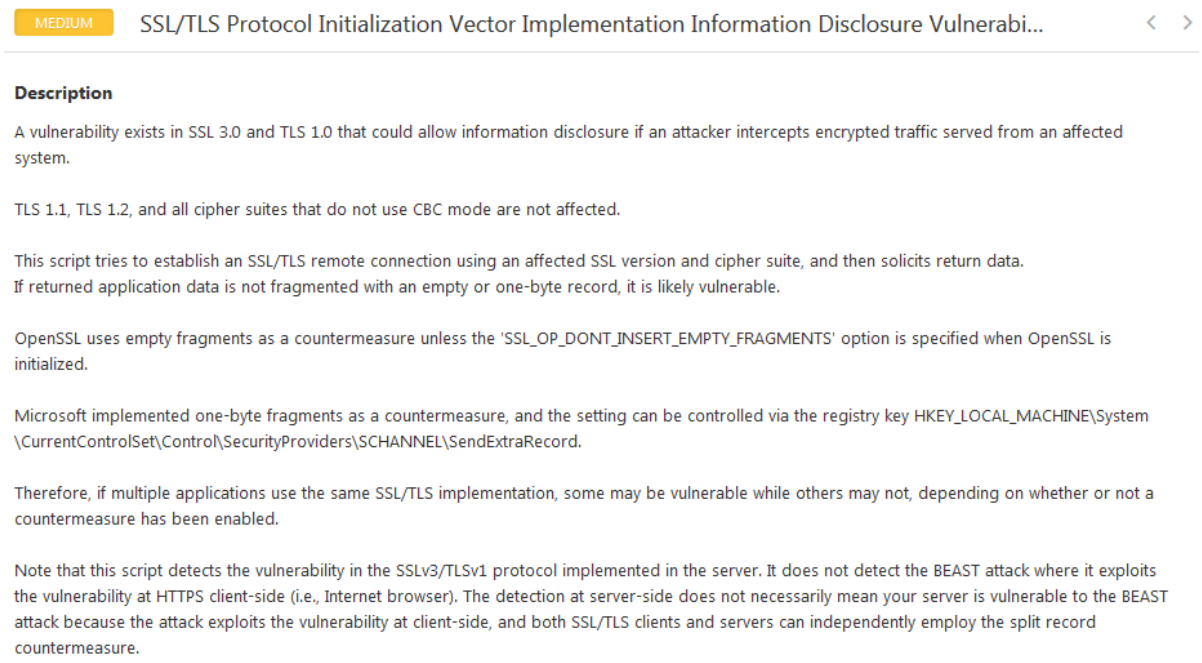
3. SSL/TLS Protocol Initialization Vector Implementation Information Disclosure Vulnerability

Η ευπάθεια αυτή υπάρχει στο πρωτόκολλο SSL 3.0 και TLS 1.0 η οποία μπορεί να επιτρέψει την αποκάλυψη πληροφορίας αν ο επιτιθέμενος υποκλέψει την κρυπτογραφημένη κυκλοφορία που παρέχεται από ένα σύστημα που έχει επηρεαστεί.

Τα TLS 1.0, TLS 1.2 και όλες οι κρυπτογραφήσεις που δεν χρησιμοποιούν την μορφή CBC δεν επηρεάζονται.

Αυτό το script προσπαθεί να δημιουργήσει μια απομακρυσμένη σύνδεση SSL/TLS χρησιμοποιώντας «πειραγμένη» την έκδοση SSL και το κρυπτογράφημα έτσι ώστε στη συνέχεια να «ψαρεύει» τα δεδομένα που επιστρέφονται. Αν τα δεδομένα της εφαρμογής που επιστρέφουν δεν είναι κατακερματισμένα με ένα άδειο αρχείο ή ένα αρχείο ενός byte, είναι πιθανό να είναι ευπαθή.

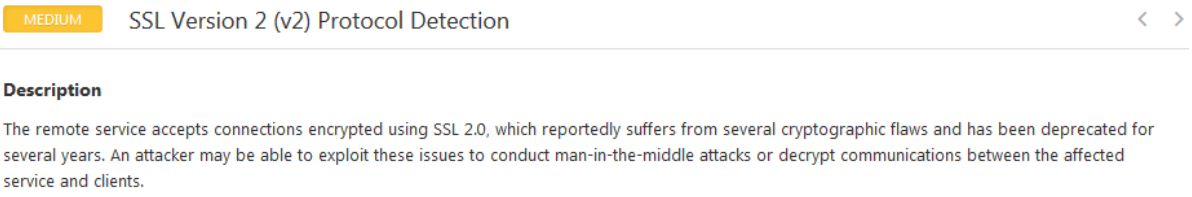
Το OpenSSL χρησιμοποιεί σαν αντίμετρο άδεια τμήματα (fragments) εκτός αν η επιλογή "SSL_OP_DONT_INSERT_EMPTY_FRAGMENTS" έχει καθοριστεί κατά την αρχικοποίηση του.



Εικόνα 73. Vulnerability medium severity – SSL/TLS Protocol Initialization Vector Implementation Information Disclosure Vulnerability

4. SSL Version 2 (v2) Protocol Detection

Η ευπάθεια εδώ ορίζεται στην αποδοχή κρυπτογραφημένων συνδέσεων που χρησιμοποιούν SSL 2.0, το οποίο κατά τα λεγόμενα υποφέρει από διάφορα κρυπτογραφικά ελαττώματα και έχει αποδοκιμαστεί για αρκετά χρόνια. Σε έναν επιτιθέμενο μπορεί να δοθεί η δυνατότητα να εκμεταλλευθεί αυτά τα ελαττώματα για να πραγματοποιήσει επιθέσεις man-in-the-middle ή να αποκωδικοποιήσει τις επικοινωνίες ανάμεσα στην επηρεασμένη υπηρεσία και στους πελάτες.

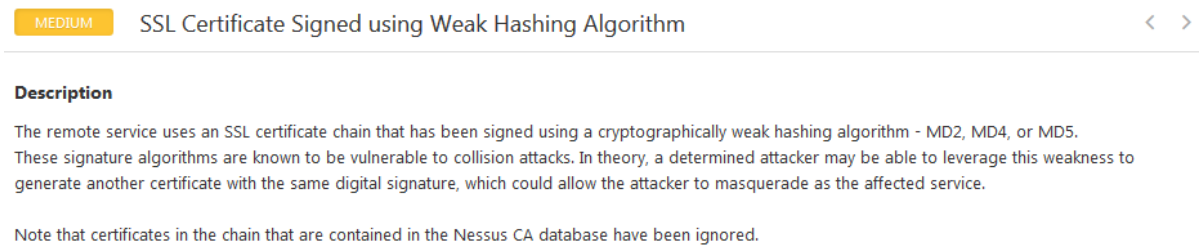


Εικόνα 74. Vulnerability medium severity – SSL Version 2 (v2) Protocol Detection

5. SSL Certificate Signed using Weak Hashing Algorithm

Ο κίνδυνος σε αυτή την περίπτωση προέρχεται πάλι από το πιστοποιητικό SSL που χρησιμοποιείται καθώς μας ενημερώνει ότι η «αλυσίδα» των πιστοποιητικών έχει υπογραφεί από έναν αλγόριθμο hash κρυπτογραφικά αδύναμο. Στην προκειμένη περίπτωση αναφέρεται στον αλγόριθμο MD5. Η υπογραφή με αυτόν τον αλγόριθμο είναι ευπαθής σε επιθέσεις σύγκρουσης (collision attacks).

Σύμφωνα με την θεωρία ο επιτιθέμενος μπορεί να αξιοποιήσει αυτή την αδυναμία ώστε να δημιουργήσει ένα άλλο πιστοποιητικό με την ίδια ψηφιακή υπογραφή, το οποίο θα του επιτρέπει να προσποιείται ότι είναι αυτός η παρεχομένη υπηρεσία.

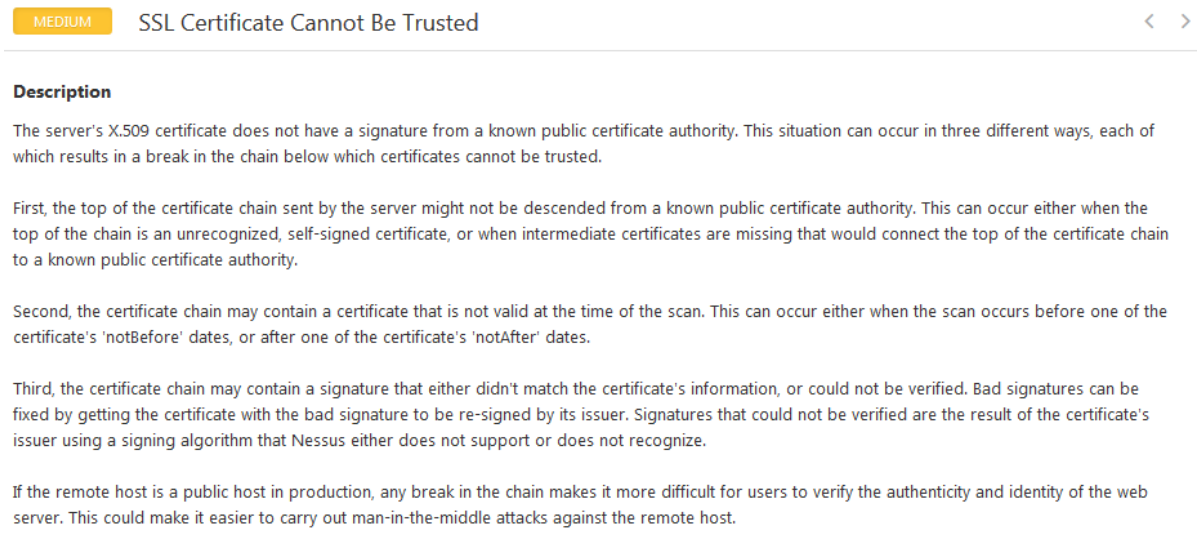


Εικόνα 75. Vulnerability medium severity – SSL Certificate Signed using Weak Hashing Algorithm

6. SSL Certificate Cannot Be Trusted

Το πιστοποιητικό X.509 του server δεν έχει υπογραφή από μια αναγνωρισμένη αρχή πιστοποίησης. Αυτό σύμφωνα με τις πιθανές αιτίες που μπορεί να το προκαλέσουν, οφείλεται στο ότι το πιστοποιητικό που χρησιμοποιήθηκε είναι αυτό-υπογεγραμμένο και δεν χρησιμοποιήθηκε μια αναγνωρισμένη αρχή πιστοποίησης.

Ο κίνδυνος που διατρέχει η εφαρμογή από αυτή την ευπάθεια είναι ότι γίνεται ευκολότερη η διενέργεια επιθέσεων man-in-the-middle.



Εικόνα 76. Vulnerability medium severity – SSL Certificate Cannot Be Trusted

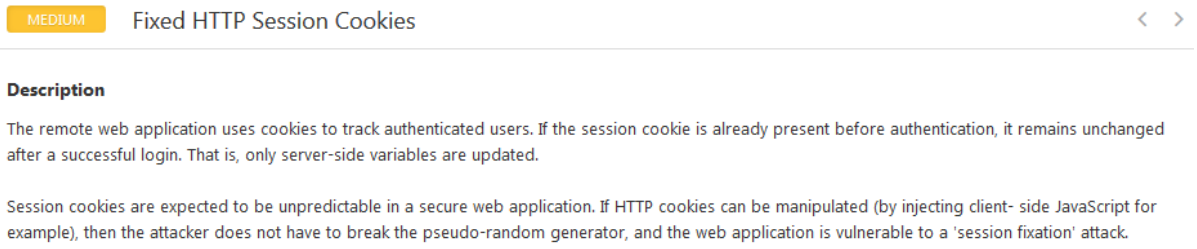
5.3.3 Ευπάθειες Διαχείρισης Συνόδου

Σε αυτές περιλαμβάνονται οι ευπάθειες 1) Fixed HTTP Session Cookies 2) Web Application Session Cookies Not Marked Secure και 3) Web application Session Cookies Not Marked Http Only

1. Fixed HTTP Session Cookies

Η εφαρμογή χρησιμοποιεί cookies για να εντοπίζει τους αυθεντικοποιημένους χρήστες. Αν η διαδικασία με την χρήση των cookies υπάρχει πριν την αυθεντικοποίηση, τότε αυτή δεν αλλάζει μετά από μια επιτυχημένη σύνδεση του χρήστη. Επομένως ανανεώνονται μόνο οι μεταβλητές στην πλευρά του server.

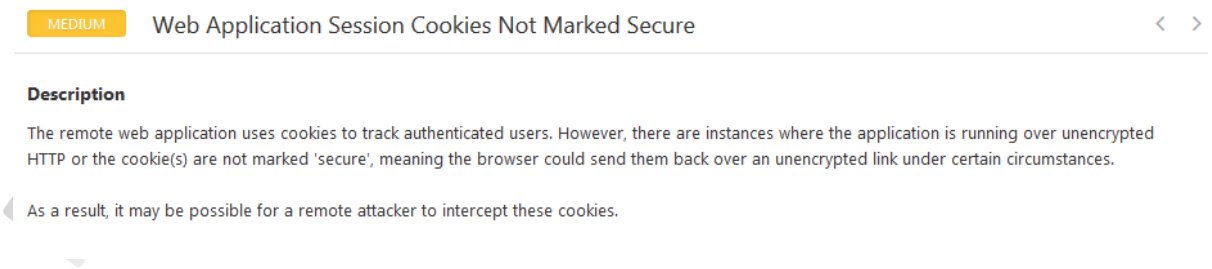
Η διαδικασία με την χρήση των cookies αναμένεται να μην είναι προβλέψιμη, οπότε αν τα HTTP cookies μπορούν να χειραγωγούνται (με την έγχυση ενός Javascript από την πλευρά του πελάτη) τότε ο επιτιθέμενος δεν χρειάζεται να σπάσει την ψευδο-τυχαία γεννήτρια και η εφαρμογή είναι ευπαθής σε επιθέσεις “session fixation”.



Εικόνα 77. Vulnerability medium severity – Fixed HTTP Session Cookies

2. Web Application Session Cookies Not Marked Secure

Σε αυτή την ευπάθεια η εφαρμογή χρησιμοποιεί cookies για να εντοπίζει τους αυθεντικοποιημένους χρήστες. Παρόλα αυτά υπάρχουν περιπτώσεις που η εφαρμογή δεν χρησιμοποιεί ασφαλείς συνδέσεις (π.χ Http) ή τα cookies δεν αναφέρονται ως ασφαλή, που σημαίνει ότι ο φυλλομετρητής μπορεί να τα στείλει πίσω από μια μη κρυπτογραφημένη και κατά συνέπεια μη ασφαλή σύνδεση. Αυτό εμφανίζει τον κίνδυνο ένας επιτιθέμενος να τα υποκλέψει.

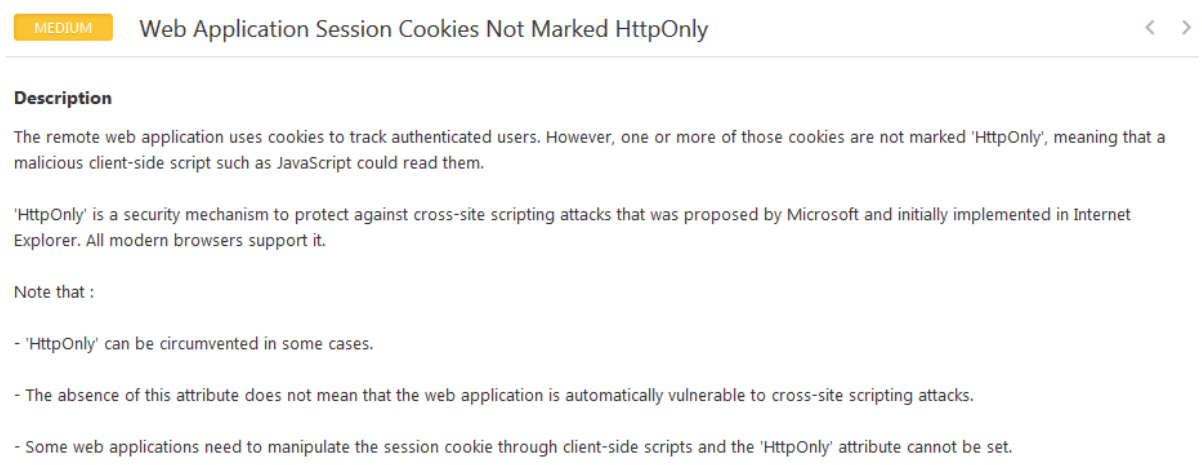


Εικόνα 78. Vulnerability medium severity – Web Application Session Cookies Not Marked

3. Web application Session Cookies Not Marked Http Only

Η συγκεκριμένη ευπάθεια έχει συνάφεια με την προηγουμένη αφού αφορά τα cookies. Ένα η περισσότερα από αυτά δεν είναι χαρακτηρισμένα σαν “HttpOnly” που σημαίνει ότι ένα κακόβουλο script όπως πχ. ένα Javascript από την πλευρά του πελάτη θα μπορούσε να τα διαβάσει.

Το “HttpOnly” είναι ένας μηχανισμός ασφαλείας για προστασία από επιθέσεις cross-site scripting το οποίο προτείνεται από την Microsoft και είχε αρχικά εφαρμοστεί στον Internet Explorer.



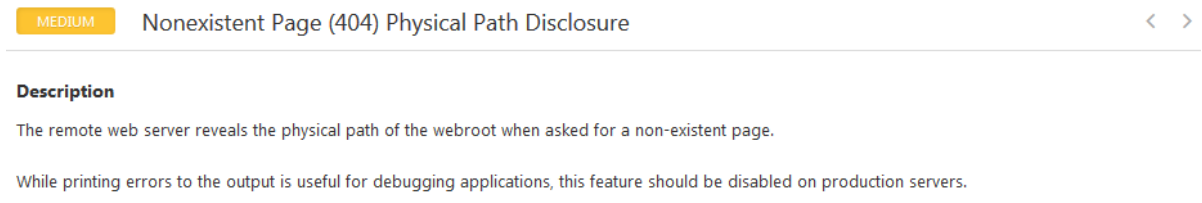
Εικόνα 79. Vulnerability medium severity – Web Application Session Cookies Not Marked Http Only

5.3.4 Ευπάθειες στη πλευρά του εξυπηρετητή

Σε αυτή την κατηγορία περιλαμβάνεται μόνο η ευπάθεια Nonexistent Page (404) Physical Path Disclosure

Nonexistent Page (404) Physical Path Disclosure

Ο web server που χρησιμοποιήθηκε αποκαλύπτει την τοποθεσία που βρίσκεται το site όταν ερωτηθεί για μια μη-υπαρκτή σελίδα (non-existent page). Έτσι προειδοποιεί ότι ενώ αυτό είναι χρήσιμο κατά την ανάπτυξη κάποιας εφαρμογής, σε περίπτωση χρήσης για υποστήριξη εμπορικής εφαρμογής θα πρέπει να απενεργοποιηθεί από τον server.



Εικόνα 80. Vulnerability medium severity – Nonexistent Page (404) Physical Path Disclosure

Κεφάλαιο 6°

6 Ανασχεδιασμός Ασφάλειας και Εφαρμογή Μέτρων Ασφάλειας

6.1 Εισαγωγή

Ο εντοπισμός των ευπαθειών σε μια διαδικτυακή εφαρμογή «απαιτεί» την ανάπτυξη των καταλλήλων μέτρων ώστε να προστατευθεί από τους δυνητικούς επιτιθέμενους. Σε αυτό το κεφάλαιο θα αναπτυχθούν και θα παρουσιαστούν τα μέτρα αυτά που αντιμετωπίζουν τις ευπάθειες χαμηλού και μεσαίου κινδύνου όλων των κατηγοριών που αναλύθηκαν στο προηγούμενο κεφάλαιο.

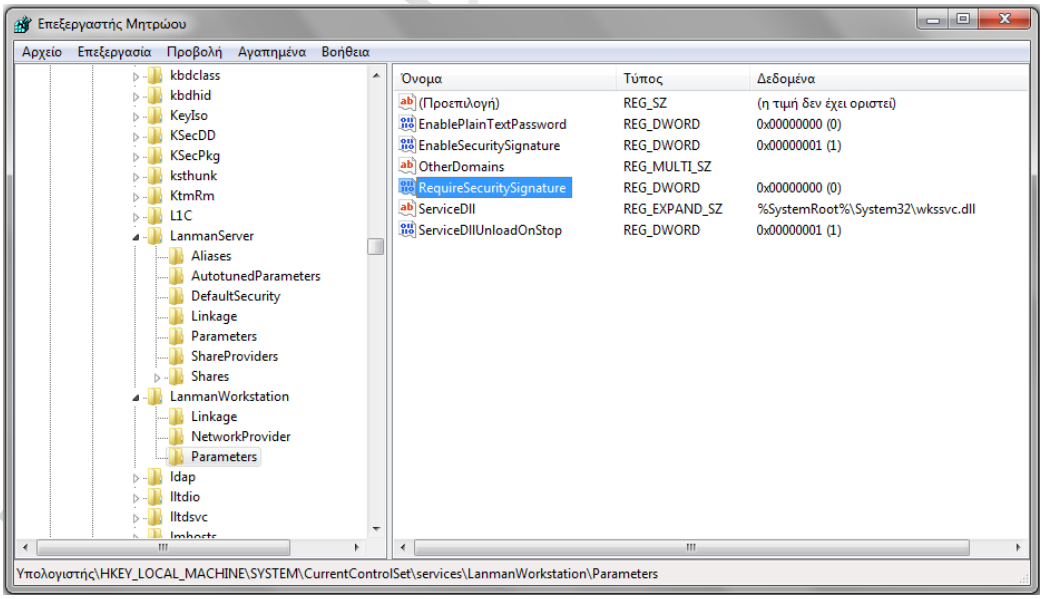
Τα μέτρα που θα ληφθούν σε αρκετές περιπτώσεις αντιμετωπίζουν περισσότερες από μια ευπάθειες. Αφορούν τόσο την διόρθωση του κώδικα της εφαρμογής (Security re-engineering) αλλά και ρυθμίσεις του τεχνολογικού περιβάλλοντος που φιλοξενεί την εφαρμογή.

6.2 Αντιμετώπιση ευπαθειών Αυθεντικοποίησης

6.2.1 Αντιμετώπιση ευπάθειας SMB Signing Required

Για την προστασία από την ευπάθεια «SMB Signing Required» πρέπει να ενεργοποιηθεί η ψηφιακά υπογεγραμμένη επικοινωνία με τον server [40]. Η ενεργοποίηση γίνεται από τον registry προορισμό

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\LanManWorkstation\ParametersValue\RequireSecuritySignature θέτοντας την τιμή στο (REG_DWORD) σε 00000001.



Εικόνα 81. Ενεργοποίηση RequireSecuritySignature

6.2.2 Αντιμετώπιση ευπάθειας Web Server Uses Plain Text Authentication Forms

Η προστασία από την ευπάθεια «Web Server Uses Plain Text Authentication Forms» θα γίνει με την ρύθμιση της εφαρμογής μας ώστε όλες οι σελίδες να απαιτούν χρήση του πρωτοκόλλου https. Αυτό θα γίνει με την προσθήκη του χαρακτηριστικού [RequireHttps] στους Controllers όλων των σελίδων.

```
using MvcMovie.Mailers.Models;
using SecurityGuard.Core;
using SecurityGuard.Interfaces;
using SecurityGuard.Services;
using SecurityGuard.ViewModels;
using viewModels = MvcMovie.Areas.SecurityGuard.ViewModels;
using Recaptcha.Web.Mvc;
using Recaptcha.Web;
using System.Net;

namespace MvcMovie.Controllers
{
    /// <summary>
    /// This class handles all the normal logon, logoff,
    /// register, change password, and forgot password operations
    /// that occur in the public part of your web application.
    /// </summary>
    [RequireHttps]
    public class SGAccountController : BaseController
    {

        #region ctors

        private IMembershipService membershipService;
        private IAuthenticationService authenticationService;
        private IFormsAuthenticationService formsAuthenticationService;
        private IPasswordResetMailer _mailer = new PasswordResetMailer();

        public SGAccountController()
        {
            this.membershipService = new MembershipService(Membership.Provider);
            this.authenticationService = new AuthenticationService(membershipService, new FormsAuthenticationService());
            this.formsAuthenticationService = new FormsAuthenticationService();
```

Εικόνα 82. MvcMovie Controllers – προσθήκη [RequireHttps]

Επίσης στα ViewModels όλων των σελίδων θα ενεργοποιηθεί το πρωτόκολλο https όπως φαίνεται στο παρακάτω παράδειγμα.

Αρχική μορφή

@Html.ActionLink("Register", "Register", "SGAccount", null, null, null, null, null)

Ενεργοποιημένο “https”

@Html.ActionLink("Register", "Register", "SGAccount", “https”, null, null, null, null)

Τέλος στο αρχείο web.config θα τεθεί τιμή στην μεταβλητή requireSSL="true" ενώ πριν η τιμή της ήταν false.

```
<sessionState mode="InProc" customProvider="DefaultSessionProvider">
  <providers>
    <add name="DefaultSessionProvider" type="System.Web.Providers.DefaultSessionStateProvider, System.Web.Providers
  </providers>
</sessionState>
<authentication mode="Forms">
  <forms loginUrl="~/SGAccount/Login" timeout="2880" cookieless="UseCookies" requireSSL="true" />
```

Εικόνα 83. Ρύθμιση αρχείου web.config

Έτσι εξασφαλίζεται ότι όλα τα ευαίσθητα δεδομένα που σχετίζονται με την αυθεντικοποίηση πια διακινούνται μέσω κρυπτογραφημένων συνδέσεων.

6.3 Αντιμετώπιση ευπαθειών Χρήσης SSL

6.3.1 Αντιμετώπιση ευπαθειών SSL Protocol

Προκείμενου να αποτραπούν οι παρακάτω δύο ευπάθειες [41], [42]

- «SSL Version 2 (v2) Protocol Detection»
- «SSL/TLS Protocol Initialization Vector Implementation Information Disclosure Vulnerability»

θα απενεργοποιηθεί το πρωτόκολλο SSL 2.0 έτσι ώστε οι υπηρεσίες IIS να μην επιχειρούν να διαπραγματευτούν χρησιμοποιώντας αυτό και θα ενεργοποιηθεί το πρωτόκολλο TLS 1.2. Επειδή για το πρωτόκολλο που χρειάζεται να ενεργοποιηθεί δεν υπάρχουν τα αντίστοιχα κλειδιά αναγκαστικά θα δημιουργηθούν.

Πρώτα θα δημιουργηθούν τα κλειδιά του TLS 1.2 για τον client και τον server. Από command line PowerShell θα εκτελεστούν οι ακόλουθες εντολές.

```
md “HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\
Protocols\TLS 1.2”

md “HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\
Protocols\TLS 1.2\Client”
```

Στη συνέχεια θα τα ενεργοποιήσουμε με τις ακόλουθες εντολές.

```
new-itemproperty-path “HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\
SCHANNEL\Protocols\TLS 1.2\Server” -name “Enabled” -value 1 -PropertyType “DWord”

new-itemproperty-path “HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\
SCHANNEL\Protocols\TLS 1.2\Server” -name “DisabledByDefault” -value 0 -
PropertyType “DWord”

new-itemproperty-path “HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\
SCHANNEL\Protocols\TLS 1.2\Client” -name “Enabled” -value 1 -PropertyType “DWord”

new-itemproperty-path “HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\
SCHANNEL\Protocols\TLS 1.2\Client” -name “DisabledByDefault” -value 0 -
PropertyType “DWord”
```

Τέλος θα απενεργοποιήσουμε το πρωτόκολλο SSL 2.0.

```
md “HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\
```

Protocols\SSL 2.0\Server”

new-itemproperty-path “HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\SSL 2.0\Server” -name Enabled -value 0 -PropertyType “DWord”

6.3.2 Αντιμετώπιση ευπαθειών SSL Certificate

Οι τρείς ευπάθειες που ακολουθούν αφορούν το πιστοποιητικό

- «SSL Certificate Cannot Be Trusted»
- «SSL Certificate Signed using Weak Hashing Algorithm»
- «SSL Certificate Chain Contains RSA Keys Less Than 2048 bits»

και για την αντιμετώπισή τους θα χρησιμοποιηθεί ένα πιστοποιητικό από μια αναγνωρισμένη Αρχή πιστοποιητικών (Certificate Authority) το οποίο θα τηρεί όλες τις σύγχρονες αρχές ασφαλείας και δεν θα είναι αυτό-υπογεγραμμένο όπως συμβαίνει στην παρούσα εργασία. Ούτε θα έχει χρησιμοποιηθεί για την υπογραφή του «αδύναμος» αλγόριθμος και τέλος τα κλειδιά που θα χρησιμοποιεί θα είναι μεγαλύτερα από 2048 bits.

Βέβαια η αγορά και η χρήση ενός τέτοιου πιστοποιητικού για την αντιμετώπιση των παραπάνω ευπαθειών θα υλοποιηθεί στην περίπτωση που η διαδικτυακή εφαρμογή έχει σκοπό να χρησιμοποιηθεί για εμπορική χρήση. Στα πλαίσια της μεταπτυχιακής εργασίας δεν κρίνεται απαραίτητο.

6.3.3 Αντιμετώπιση ευπάθειας SSL RC4 Cipher Suites Supported

Τέλος η ευπάθεια «SSL RC4 Cipher Suites Supported» θα αντιμετωπιστεί με την απενεργοποίηση του RC4 Cipher και την ενεργοποίηση του AES-GCM [43]. Ο συνδυασμός της χρήσης του TLS 1.2 με το AES-GCM προτείνεται από την Microsoft για την χρήση κρυπτογραφικών συνδέσεων. Η απενεργοποίηση του RC4 Cipher γίνεται με τον εξής τρόπο.

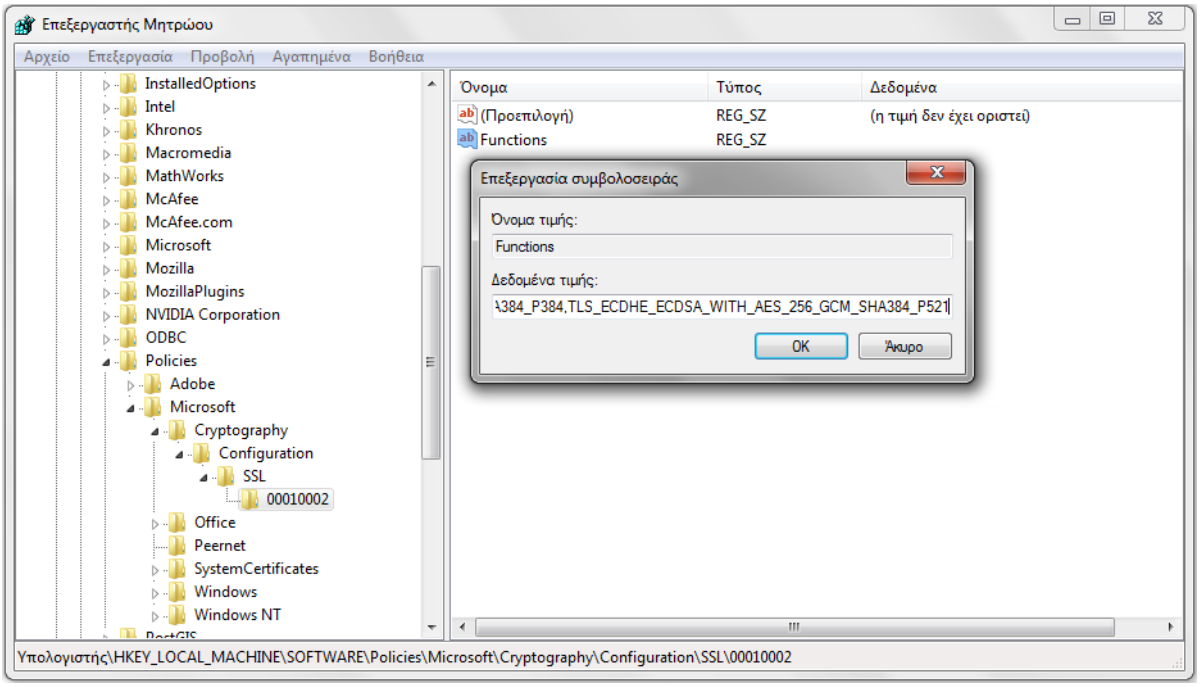
Στους παρακάτω registry προορισμούς στο (REG_DWORD) μετατρέπεται η τιμή σε 00000000.

- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\RC4 128/128
- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\RC4 40/128
- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\RC4 56/128

Για την ενεργοποίηση των AES-GCM Cipher Suites αντίστοιχα στον προορισμό registry HKLM\SOFTWARE\Policies\Microsoft\Cryptography\Configuration\SSL\00010002

δημιουργείται μια νέα τιμή συμβολοσειράς με το όνομα Functions όπου προστίθεται η ακόλουθη συμβολοσειρά όπου έχει κατά σειρά προτεραιότητας τα AES-GCM που χρειάζεται να χρησιμοποιούνται.

TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256_P256,TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256_P384,TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256_P521,TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384_P384,TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384_P521



Εικόνα 84. Ενεργοποίηση AES-GCM Cipher Suites

6.4 Αντιμετώπιση ευπαθειών Διαχείρισης Συνόδου

6.4.1 Αντιμετώπιση ευπάθειας Fixed HTTP Session Cookies

Η ευπάθεια «Fixed HTTP Session Cookies» θα αντιμετωπιστεί με την εισαγωγή του χαρακτηριστικού [ValidateAntiForgeryToken] που διατίθεται από το .NET Framework για την αντιμετώπιση επιθέσεων Cross-Site Request Forgery (CSRF). Η λειτουργία του είναι ότι κάθε φορά που δημιουργείται μια σελίδα, δημιουργεί ένα κρυμμένο πεδίο με μια νέα τιμή (token). Αυτό μαζί με το ήδη υπάρχον cookie αποτελούν ένα νέο, μοναδικό κάθε φορά συνδυασμό. Οπότε μετά από μια επιτυχημένη αυθεντικοποίηση ενός χρήστη ενώ το cookie παραμένει ίδιο, ο συνδυασμός των δύο είναι κάθε φορά διαφορετικός.

Αυτό θα γίνει με την εισαγωγή του χαρακτηριστικού [ValidateAntiForgeryToken] στις μεθόδους post, στους Controllers όλων των σελίδων.

```
[HttpPost]
[ValidateAntiForgeryToken]
public virtual ActionResult Login(LogOnViewModel model, string returnUrl)
{
    if (ModelState.IsValid)
    {
        if (authenticationService.LogOn(model.UserName, model.Password, model.RememberMe))
        {
            if (Url.IsLocalUrl(returnUrl) && returnUrl.Length > 1 && returnUrl.StartsWith("/")
                && !returnUrl.StartsWith("//") && !returnUrl.StartsWith("/\\"))
            {
                return Redirect(returnUrl);
```

Εικόνα 85. Ρύθμιση Controllers – προσθήκη [RequireHttps]

Επίσης στα Views όλων των σελίδων θα προστεθεί το AntiforgeryToken με την μέθοδο HtmlHelper @Html.AntiForgeryToken() όπως φαίνεται στο παρακάτω παράδειγμα.

```
<section id="loginForm">
<h2>Use a local account to log in.</h2>
@using (Html.BeginForm(new { ReturnUrl = ViewBag.ReturnUrl })) {
    @Html.AntiForgeryToken()
    @Html.ValidationSummary(true)
```

Εικόνα 86. Ρύθμιση Views – προσθήκη @Html.AntiforgeryToken()

6.4.2 Αντιμετώπιση ευπαθειών Cookies

Οι δύο ευπάθειες που αφορούν τα cookies

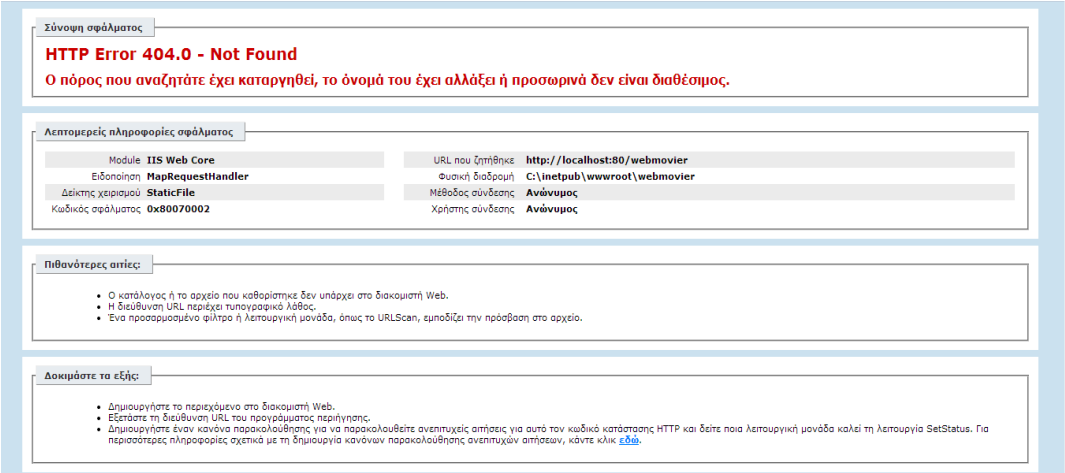
- «Web Application Session Cookies Not Marked HttpOnly»
- «Web Application Session Cookies Not Marked Secure»

θα αντιμετωπιστούν με την ρύθμιση της εφαρμογής μας ώστε όλες οι σελίδες να απαιτούν χρήση του πρωτοκόλλου https όπως παρουσιάστηκε στο κεφάλαιο 6.2.2. Οπότε όσον αφορά αυτές τις δύο ευπάθειες, τα cookies θα διακινούνται πλέον μέσω συνδέσεων με κρυπτογράφηση και δεν τίθεται θέμα ασφάλειας τους.

6.5 Αντιμετώπιση ευπαθειών στην πλευρά του εξυπηρετητή (server-side)

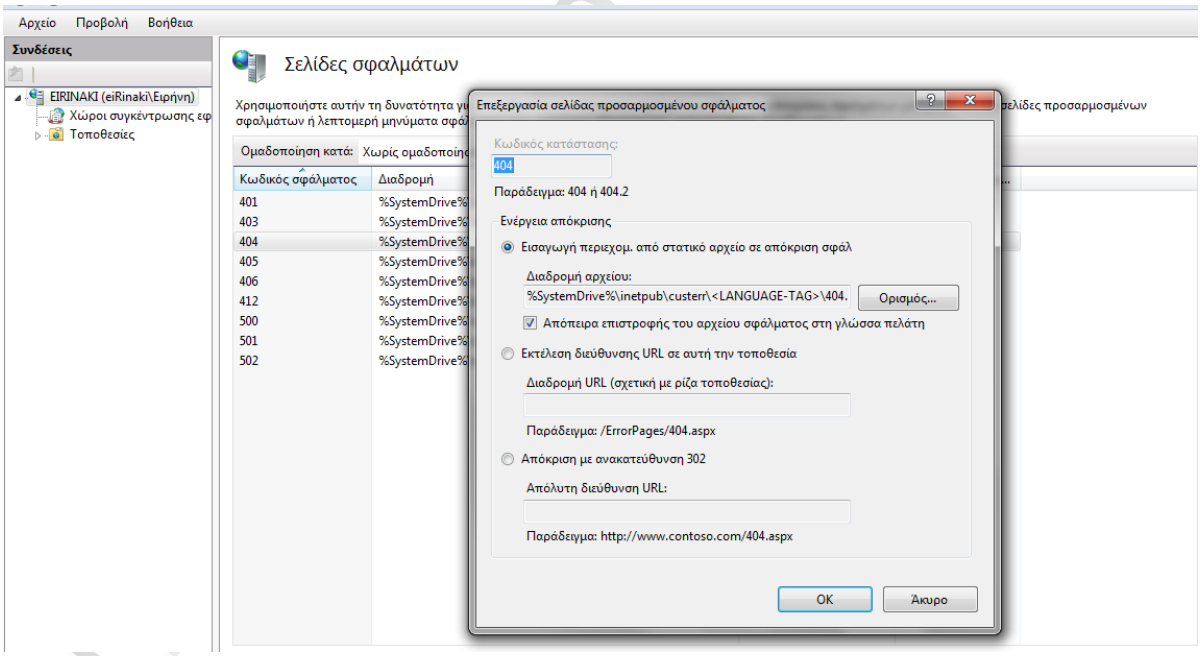
6.5.1 Αντιμετώπιση ευπάθειας Nonexistent Page (404) Physical Path Disclosure

Η ευπάθεια «Nonexistent Page (404) Physical Path Disclosure» που πληροφορεί ότι ο server αποκαλύπτει την φυσική τοποθεσία που βρίσκεται το site μας μπορεί να αντιμετωπιστεί με την ρύθμιση του ώστε κάθε φορά που ερωτάται να μην εμφανίζει την αναλυτική σελίδα που αποκαλύπτει την φυσική τοποθεσία.



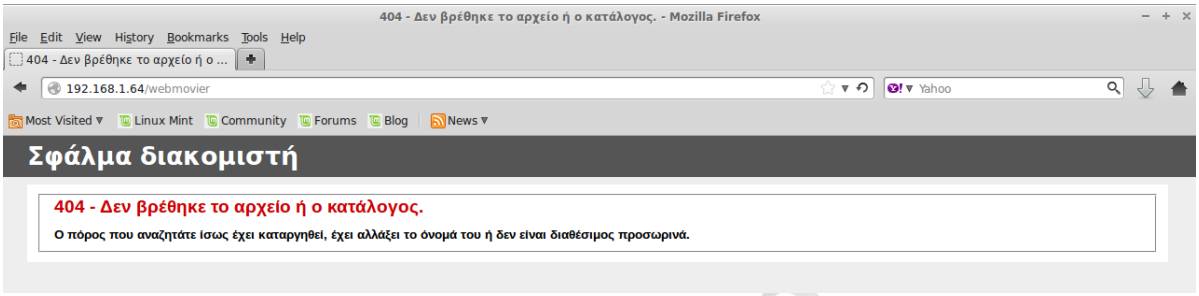
Εικόνα 87. Σελίδα λάθους 404 από ερώτηση τοπικού δικτύου στον server

Αυτό μπορεί να γίνει από τις ρυθμίσεις του IIS στις Σελίδες Σφαλμάτων (Εικόνα 88) όπου δίνεται η δυνατότητα να οριστεί να δείχνει κάποια άλλη σελίδα όταν ο server ερωτάται για κάποια σελίδα που δεν υφίσταται.



Εικόνα 88. Ρύθμιση σελίδας σφαλμάτων IIS

Παρόλα αυτά, στη συγκεκριμένη περίπτωση δεν χρειάζεται να γίνει κάποια αλλαγή στον server μας γιατί είναι ρυθμισμένος να αποκαλύπτει την φυσική τοποθεσία μόνο όταν ερωτάται τοπικά, το οποίο είναι χρήσιμο για την επίλυση προβλημάτων. Ενώ όταν ερωτάται από κάποιον εξωτερικό πελάτη, τότε η απάντηση είναι ότι δεν βρέθηκε το αρχείο ή ο κατάλογος.



Εικόνα 89. Σελίδα λάθους 404 από ερώτηση εκτός τοπικού δικτύου στον server

Κεφάλαιο 7ο

7 Συμπεράσματα

7.1 Γενικά συμπεράσματα

Η ασφάλεια των διαδικτυακών εφαρμογών είναι ένα σύνθετο ζήτημα που δεν μπορεί να αντιμετωπιστεί μεμονωμένα και πρόσκαιρα και στη συνέχεια να παραμεληθεί ή να αγνοηθεί. Οι εφαρμογές εμφανίζουν στην πλειοψηφία τους κενά ασφαλείας που ενσωματώθηκαν σε αυτές κατά την διάρκεια του σχεδιασμού τους αλλά και στην υλοποίησή τους. Επειδή το περιβάλλον στο οποίο λειτουργεί η εφαρμογή επηρεάζει την ασφαλή της λειτουργία αλλά και επειδή οι απαιτήσεις ασφάλειας καθημερινά αυξάνονται, οι έλεγχοι ασφάλειας σε τακτά χρονικά διαστήματα κρίνονται πλέον απαραίτητοι για τον εντοπισμό νέων κενών ασφάλειας.

Οι υπεύθυνοι ασφάλειας θα πρέπει να αξιολογούν την αναγκαιότητα ή όχι της διόρθωσης κάθε αδυναμίας χωριστά. Για όσες αδυναμίες πρέπει να διορθωθούν, θα πρέπει να γίνεται συντήρηση της εφαρμογής αλλά και ενδεχομένως αλλαγή ρυθμίσεων του τεχνολογικού περιβάλλοντος στο οποίο έχει εγκατασταθεί η εφαρμογή.

Τα τελευταία χρόνια, η εκμετάλλευση των αδυναμιών ασφάλειας έχει ανεβεί στην πρώτη θέση των κινδύνων. Η αιτία γι' αυτό το γεγονός είναι ότι οι αδυναμίες των διαδικτυακών εφαρμογών αποτελούν τον πιο αδύναμο κρίκο στην αλυσίδα των μέτρων πρόληψης της ασφάλειας των πληροφοριακών συστημάτων. Οι επίδοξοι hackers το γνωρίζουν πολύ καλά αυτό και έτσι εκμεταλλεύονται τις αδυναμίες αυτές προκειμένου να αποκτήσουν τις πληροφορίες που θέλουν.

Δυστυχώς οι διαχειριστές εφαρμογών και δικτύων συνήθως δεν έχουν την απαραίτητη εκπαίδευση και γνώση για να αξιολογήσουν τον κίνδυνο υποκλοπής πληροφοριών μέσω των αδυναμιών των εφαρμογών και έτσι δεν ενσωματώνουν στην πολιτική ασφάλειας τα απαραίτητα προληπτικά μέτρα. Επίσης δεν έχουν την απαραίτητη εκπαίδευση και γνώση για τον προσδιορισμό απαιτήσεων ασφάλειας των διαδικτυακών εφαρμογών. Οι απαιτήσεις αυτές θα πρέπει να συμπεριλαμβάνονται στον σχεδιασμό και την υλοποίηση των εφαρμογών, εφόσον οι υπεύθυνοι ασφάλειας το κρίνουν αναγκαίο.

Ο OWASP από το 2003 έχει κάνει μια σημαντική προσπάθεια κατηγοριοποίησης των 10 συνηθέστερων και σημαντικότερων αδυναμιών ασφαλείας. Αυτό όμως σε καμία περίπτωση δεν σημαίνει ότι οι εκατοντάδες άλλες γνωστές αδυναμίες εξακολουθούν να μην είναι στις σημαντικότερες και παράλληλα να αποτελούν σημείο ιδιαίτερου ενδιαφέροντος για την ασφάλεια καθώς εμφανίζονται στο προσκήνιο ανά τακτά χρονικά διαστήματα.

Τα δεδομένα του OWASP βασίζονται σε πραγματικά στοιχεία περίπου 500,000 αδυναμιών λογισμικού. Αυτά αντλήθηκαν από εκατοντάδες οργανισμούς και χιλιάδες εφαρμογές δίνοντάς μας έτσι μια πραγματική εικόνα του πώς είναι το πεδίο. Επίσης με το έργο του συμβάλλει στη διαρκή ενημέρωση του ενδιαφερόμενου κοινού τόσο για την ύπαρξη των κινδύνων όσο και για τους τρόπους αντιμετώπισης τους με την παροχή λογισμικού υψηλής ποιότητας σε προγραμματιστές και αναλυτές ασφαλείας.

Ο έλεγχος μιας διαδικτυακής εφαρμογής εξαρτάται από την συχνότητα υλοποίησής του και από την επιλογή του εργαλείου ελέγχου. Σήμερα διατίθεται προς χρήση ένας μεγάλος αριθμός αξιόπιστων εργαλείων για εντοπισμό ευπαθειών σε διαδικτυακές εφαρμογές. Αυτά τα εργαλεία διατηρούν βάσεις δεδομένων με τις ευπάθειες που έχουν παρουσιαστεί μέχρι τώρα και βασιζόμενες σε αυτές κάνουν τους ελέγχους τους. Είναι πολύ σημαντικό να ενημερώνουν συνεχώς τις βάσεις τους αλλιώς οι

έλεγχοι δε θα εντοπίζουν τις νέες ευπάθειες που εμφανίζονται και έτσι δεν θα έχουν και καμία χρησιμότητα να επαναλαμβάνονται.

Το Nessus συγκαταλέγεται ανάμεσα στα κορυφαία της κατηγορίας του με μια βάση ευπαθειών που ανανεώνεται συνεχώς. Παρέχει στον χρήστη δυνατότητες διαμόρφωσης της κατάλληλης πολιτικής ελέγχου ώστε να είναι στοχευμένος ο έλεγχος και να εντοπίζει-αποκαλύπτει τις ευπάθειες που φέρει μια εφαρμογή. Επίσης παρέχει αναφορές με αναλυτικές πληροφορίες για τις ευπάθειες και ενδεικτικές προτάσεις για την αντιμετώπιση τους.

7.2 Ειδικά συμπεράσματα

Για την ανάδειξη της σημασίας της ασφάλειας στις διαδικτυακές εφαρμογές και παρουσίασης της σε πρακτικό επίπεδο χρησιμοποιήθηκε η εφαρμογή (Online Movies) που αναπτύχθηκε σε περιβάλλον .NET. Όπως παρουσιάστηκε και στα προηγούμενα κεφάλαια το περιβάλλον αυτό δίνει τα μέσα για την ανάπτυξη αξιόπιστων εφαρμογών. Στην πρώτη φάση του ελέγχου διαπιστώθηκε ότι η σάρωση της εφαρμογής με την χρήση των διαπιστευτηρίων ενός αυθεντικοποιημένου χρήστη είναι ενδελεχής και εντοπίζει όλες τις αδυναμίες σε σχέση με μια σάρωση χωρίς την χρήση αυτών.

Στον έλεγχο αυτό εντοπίστηκαν ευπάθειες οι οποίες ήταν μεσαίου και χαμηλού επιπέδου κινδύνου. Έχει μεγάλη σημασία που δεν εντοπίστηκε ευπάθεια υψηλού κινδύνου. Οι ευπάθειες αυτές αφορούσαν τον κώδικα της εφαρμογής αλλά και το τεχνολογικό περιβάλλον που την φιλοξενεί. Βέβαια το πρώτο στάδιο ανάπτυξης της εφαρμογής είναι λογικό να έχει κάποια κενά ασφαλείας, τα οποία ονομάζονται «ευπάθειες στην ημέρα μηδέν» (zero-day vulnerabilities) αφού το πρώτο μέλημα ενός προγραμματιστή είναι η λειτουργικότητα της εφαρμογής και η όσο το δυνατόν καλύτερη εξυπηρέτηση των χρηστών που απευθύνεται.

Στη συνέχεια όμως μετά τον έλεγχο της εφαρμογής και τον εντοπισμό των ευπαθειών το σημαντικό είναι ότι το .NET Framework διαθέτει όλα τα απαραίτητα εργαλεία και λειτουργίες για την δημιουργία μέτρων αντιμετώπισης των ευπαθειών και εν τέλει θωράκισης της εφαρμογής. Έτσι με την χρήση των δυνατοτήτων που παρέχει, πραγματοποιήθηκαν και παρουσιάστηκαν όλες οι αλλαγές που είναι απαραίτητες για την εφαρμογή (Online Movies) καθιστώντας την ασφαλή για χρήση.

Τέλος το γενικό συμπέρασμα αυτής της εργασίας είναι ότι η ασφάλεια των διαδικτυακών εφαρμογών είναι ένας συνεχής αγώνας που απαιτεί την αδιάλειπτη προσπάθεια και ενημέρωση των προγραμματιστών και των υπευθύνων ασφαλείας. Συνεχώς εμφανίζονται νέες ευπάθειες που πρέπει να αντιμετωπίζονται γρήγορα και αποτελεσματικά.

Κεφάλαιο 8ο

8 Βιβλιογραφικές Πηγές

Βιβλιογραφία

[1] Σ. Κάτσικας, Ασφάλεια Δικτύων, Πάτρα: Ελληνικό Ανοιχτό Πανεπιστήμιο, 2001.

[2] J. Meier, A. Mackman, M. Dunner, S. Vasireddy, R. Escamilla και A. Murukan, Improving Web Application Security: Threats and Countermeasures, Microsoft Corporation, 2003.

[3] J. Meier. [Ηλεκτρονικό]. Available: http://blogs.msdn.com/b/jmeier/archive/2008/04/07/security-principles.aspx.

[4] Π. Κοτζανικολάου και Χ. Δουληγέρης, Ασφάλεια Δικτύων.

[5] Σ. Κάτσικας, Ασφάλεια Υπολογιστών, Πάτρα: Ελληνικό Ανοικτό Πανεπιστήμιο, 2011.

[6] Cisco, [Ηλεκτρονικό]. Available: http://www.ciscopress.com/articles/article.asp?p=1998559.

[7] Microsoft, [Ηλεκτρονικό]. Available: http://technet.microsoft.com/en-us/library/cc751383.aspx.

[8] The Owasp Foundation, [Ηλεκτρονικό]. Available: http://www.owasp.org.

[9] TechTarget, [Ηλεκτρονικό]. Available: http://searchsoftwarequality.techtarget.com/definition/OWASP.

[10] The Owasp Foundation1, [Ηλεκτρονικό]. Available: https://www.zero.gr/images/OWASP-T10-2010_greek_brochure.pdf.

[11] OWASP Greece, [Ηλεκτρονικό]. Available: http://owasp.wordpress.com/2010/04/20/owasp-top10-2010-released/.

[12] The Owasp Foundation, [Ηλεκτρονικό]. Available: https://www.owasp.org/index.php/ESAPI.

[13] The Owasp Foundation, [Ηλεκτρονικό]. Available: https://www.owasp.org/index.php/ASVS.

[14] The Owasp Foundation, [Ηλεκτρονικό]. Available: https://www.owasp.org/images/5/58/OWASP_ASVS_Version_2.pdf.

[15] Μ. Αλεξανδροπούλου, Μέθοδοι Αξιολόγησης Ασφάλειας Λογισμικού, Μεταπτυχιακή Διατριβή, Τρίπολη: Πανεπιστήμιο Πελοποννήσου, Σχολή Θετικών Επιστημών και Τεχνολογίας, Τμήμα Επιστήμης και Τεχνολογίας Υπολογιστών, 2011.

[16] T. Hunt, OWASP Top 10 for .NET developers, 2011.

[17] J. Scambray, V. Liu και C. Sima, Hacking Exposed Web Applications 3rd Edition, McGraw-Hill Companies.

[18] The Owasp Foundation, [Ηλεκτρονικό]. Available: https://www.owasp.org/images/5/56/OWASP_Testing_Guide_v3.pdf.

[19] OpenVas, [Ηλεκτρονικό]. Available: http://www.openvas.org/.

[20] PORTSWIGGER, [Ηλεκτρονικό]. Available: http://portswigger.net/burp/.

[21] beyondtrust, [Ηλεκτρονικό]. Available: http://www.beyondtrust.com/Products/RetinaNetworkSecurityScanner/.

[22] Rapid7, [Ηλεκτρονικό]. Available: http://www.rapid7.com/products/nexpose/index.jsp.

[23] Microsoft, [Ηλεκτρονικό]. Available: http://technet.microsoft.com/en-us/security/cc184923.

[24] CORE SECURITY, [Ηλεκτρονικό]. Available: http://www.coresecurity.com/core-impact-pro.

[25] Teenable Network Security, Inc, [Ηλεκτρονικό]. Available: http://www.tenable.com/products/nessus.

[26] Nmap, [Ηλεκτρονικό]. Available: http://nmap.org/.

[27] Qualys, [Ηλεκτρονικό]. Available: https://www.qualys.com/.

[28] GFI, [Ηλεκτρονικό]. Available: http://www.gfi.com/products-and-solutions/network-security-solutions/gfi-languard.

[29] Sans, [Ηλεκτρονικό]. Available: http://www.sans.org/.

[30] Teenable Network Security,Inc, «Nessus 5.2 Installation and Configuration Guide,» 10 June 2014. [Ηλεκτρονικό].

[31] H. Shah. [Ηλεκτρονικό]. Available: http://www.infosecwriters.com/text_resources/pdf/NASL_HShah.pdf.

[32] N. Rathaus. [Ηλεκτρονικό]. Available: http://blogs.securiteam.com/index.php/archives/author/noam.

[33] Teenable Network Security,Inc, «Nessus 5.2 Enterprise User Guide,» 8 June 2014. [Ηλεκτρονικό].

[34] J. Chadwick, T. Snyder και H. Panda, Programming ASP.NET MVC4, United States of America: O'REILLY Media, Inc, 2012.

[35] B. Lakshmiraghavan, Pro ASP.NET Web API Security, Apress, 2012.

[36] M. CENTRAL. [Ηλεκτρονικό]. Available: http://www.mvccentral.net/Story/Details/tools/kahanu/securityguard-nuget-package-for-asp-net-membership.

[37] Teenable Network Security, Inc, [Ηλεκτρονικό]. Available: http://static.tenable.com/documentation/Tenable_Products_Plugin_Families.pdf.

[38] Γ. Πάφιος, Αξιολόγηση Ευπαθειών Δικτυακών Εφαμογών και εξυπηρετητών, Μεταπτυχιακή Διατριβή, Πειραιάς: Πανεπιστήμιο Πειραιώς,Τμήμα Πληροφορικής.

[39] Wireshark. [Ηλεκτρονικό]. Available: https://www.wireshark.org/docs/wsug_html_chunked/.

[40] Microsoft, [Ηλεκτρονικό]. Available: http://support2.microsoft.com/kb/982860.

[41] C. Bill. [Ηλεκτρονικό]. Available: http://forums.iis.net/t/1173883.aspx.

[42] S. Derek. [Ηλεκτρονικό]. Available: http://www.derekseaman.com/2010/06/enable-tls-12-aes-256-and-sha-256-in.html.

[43] Microsoft, [Ηλεκτρονικό]. Available: http://blogs.technet.com/b/srd/archive/2013/11/12/security-

advisory-2868725-recommendation-to-disable-rc4.aspx.

[44] Π. Γιαλούρης, Μέθοδοι και εργαλεία ανάλυσης ευπαθειών δικτύων και εφαρμογών, Μεταπτυχιακή Διατριβή, Πειραιάς: Πανεπιστήμιο Πειραιώς, Τμήμα Πληροφορικής, 2011.

[45] B. Walther και B. Hope, Web Security Testing Cookbook:Systematic Techniques to Find Problems Fast, O'Reilly, 2008.