ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ ΣΤΗΝ ΚΑΤΕΥΘΥΝΣΗ

«ΔΙΚΤΥΟΚΕΝΤΡΙΚΑ ΣΥΣΤΗΜΑΤΑ»

Μεταπτυχιακή Διπλωματική Εργασία

Αξιολόγηση ευπαθειών δικτυακών εφαρμογών και εξυπηρετητών

Πάφιος Γιώργος

Α.Μ. ΜΕ09070

Επιβλέπων: Λαμπρινουδάκης Κωνσταντίνος, Επίκουρος Καθηγητής

Περιεχόμενα

Ευρετήριο εικόνων

Ευρετήριο Πινάκων

Πρόλογος

Η παρούσα μεταπτυχιακή διπλωματική εργασία πραγματοποιήθηκε στα πλαίσια του Προγράμματος Μεταπτυχιακών Σπουδών «Διδακτικής της Τεχνολογίας και Ψηφιακά Συστήματα» του τμήματος Ψηφιακών Συστημάτων του Πανεπιστημίου Πειραιώς. Στην εκπόνηση της εργασίας έπαιξε σημαντικό ρόλο το εργασιακό μου περιβάλλον, καθώς είμαι μέλος της ομάδας της Τεχνικής Υποστήριξης του τμήματος Ψηφιακών Συστημάτων του Πανεπιστημίου Πειραιώς και έτσι είχα την ευκαιρία να μελετήσω ένα θέμα το οποίο έχει απασχολήσει την ομάδα εργασίας μου καθώς και τον ίδιο, τον υπεύθυνο καθηγητή μου. Μου δόθηκε η δυνατότητα λόγω της πρότερης γνώσης που κατείχα στην αρχιτεκτονική των υποδομών, να χρησιμοποιήσω τις συγκεκριμένες πληροφορίες, καθώς και την περαιτέρω συνεργασία της ομάδα εργασίας μου, προς όφελος της διπλωματικής μου εργασίας και προς αποφυγή της πρόκλησης κάποιου προβλήματος στην ασφάλεια των ιστότοπων του τμήματος Ψηφιακών Συστημάτων του Πανεπιστημίου Πειραιώς και των δικτυακών εφαρμογών τους, που αποτελούν ένα σημαντικό μέρος της εικόνας και προβολής του τμήματος. Στα κριτήρια επιλογής του συγκεκριμένου θέματος συμπεριλαμβάνεται και το να αποτελέσει αυτή η διπλωματική εργασία ένα χρήσιμο εργαλείο μελέτης και βοήθειας για το τμήμα Ψηφιακών Συστημάτων και τους εργαζόμενους στον τομέα των πληροφοριών και τεχνολογίας (μηχανικούς ανάπτυξης λογισμικού, διαχειριστές και τεχνική υποστήριξη) του τμήματος.

Στο σημείο αυτό, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Λαμπρινουδάκη Κωνσταντίνο, ο οποίος μου έδωσε την ευκαιρία να ασχοληθώ με το συγκεκριμένο θέμα, καθώς και για την πολύτιμη καθοδήγησή του και το ενδιαφέρον που έδειξε καθ’ όλη τη διάρκεια των φάσεων της εκπόνησης της διπλωματικής μου εργασίας. Επιπλέον θέλω να ευχαριστήσω τους γονείς μου για την στήριξη που μου παρείχαν, με κάθε τρόπο, για την περάτωση των σπουδών μου και τους φίλους μου για την βοήθεια τους με τον ένα ή τον άλλο τρόπο στην προσπάθεια αυτή.

Περίληψη

Η ασφάλεια των δικτυακών εφαρμογών (Web Applications) αποτελεί ένα πρόβλημα το οποίο αφορά τόσο τους οργανισμούς και τις επιχειρήσεις, όσο και τον απλό χρήστη, αφού η διείσδυση τους στην καθημερινότητα μας είναι πλέον αναμφίβολη. Έχει διαπιστωθεί, από πλήθος διεθνών οργανισμών καθώς και από τη βιβλιογραφία, ότι με την εξέλιξη των διαδικτυακών εφαρμογών, προκύπτουν και νέα ζητήματα ασφάλειας τα οποία μάλιστα πολλές φορές παρουσιάζουν υψηλό επίπεδο δυσκολίας. Με την τάση των εταιρειών να προσφέρουν με πολύ μεγάλη συχνότητα καινούριες εφαρμογές στην αγορά, ο χρόνος ελέγχου της ασφάλειας των εφαρμογών στον κύκλο ζωής τους μειώνεται δραματικά ή ακόμη χειρότερα εξαλείφεται πλήρως. Για το λόγο αυτό, παρόλο που τα συμβατικά συστήματα ασφάλειας αξιοποιούνται στην πλέον εξελιγμένη τους έκδοση, ο κάθε οργανισμός είναι τελικά ευάλωτος σε επιθέσεις μέσω των εφαρμογών ιστού. Οι πιθανές αδυναμίες που υπάρχουν σε αυτές μπορεί να επιτρέψουν σε επίδοξους εισβολείς να παραβιάσουν την ασφάλεια ολόκληρου του πληροφοριακού συστήματος ενός οργανισμού. Οι μηχανικοί λογισμικού αλλά και οι υπεύθυνοι ασφάλειας, θα πρέπει να είναι σε θέση να ανιχνεύουν και να διαβαθμίζουν τις αδυναμίες έτσι ώστε να μπορούν να προβαίνουν στα κατάλληλα διορθωτικά μέτρα.

Η παρούσα διπλωματική εργασία, πραγματεύεται: α) τη μελέτη των κυριότερων ευπαθειών που μπορεί να εντοπιστούν σε δικτυακές εφαρμογές, β) την πειραματική εφαρμογή εργαλείων και μεθόδων για τον έλεγχο της ασφάλειας, και γ) την αποτύπωση προτάσεων - οδηγιών για την αυτόματη αξιολόγηση της ασφάλειας των εφαρμογών ιστού. Παρουσιάζονται αποτελέσματα ελέγχων που επιτρέπουν στους μηχανικούς ανάπτυξης δικτυακών εφαρμογών αλλά και στους υπεύθυνους ασφάλειας, να συλλέξουν πολύτιμες πληροφορίες και ενδείξεις σχετικά με τις αδυναμίες που εμφανίζονται σε τέτοιου είδους εφαρμογές. Η μεθοδολογία της έρευνας περιλαμβάνει μεταξύ άλλων την εξοικείωση με τις σχετικές έννοιες και τη μελέτη των οδηγιών και προτύπων ασφάλειας των εφαρμογών δικτύου, όπως έχουν οριστεί από αναγνωρισμένους οργανισμούς, όπως είναι ο OWASP [1] και ο CERT[2].

Συγκεκριμένα, μελετώντας τα προβλήματα και τις απειλές των δικτυακών εφαρμογών, έγινε προσπάθεια να εντοπιστούν και να αναδειχθούν οι αδυναμίες, καθώς και να παρασχεθούν μια σειρά κατευθυντήριων γραμμών (guidelines) που σκοπό έχουν τη

[1] OWASP, https://www.owasp.org/
[2] CERT, http://www.cert.org/

μεθόδευση της διαδικασίας εντοπισμού αδυναμιών. Παράλληλα εξετάστηκαν και σχετικά υποστηρικτικά εργαλεία που είναι διαθέσιμα, μέσω των οποίων πραγματοποιήθηκαν μετρήσεις αξιολόγησης ασφάλειας εφαρμογών ιστού. Συγκεκριμένα έγινε χρήση, κυρίως, του εργαλείου ελέγχου ασφάλειας δικτύων Nessus της TENABLE Network Security®[3], με το οποίο υπάρχει η δυνατότητα αυτοματοποίησης των σταδίων ελέγχου ασφάλειας των εφαρμογών ιστού με τη μέθοδο της δοκιμής διείσδυσης.

Στόχος μεταξύ άλλων είναι η υποστήριξη των μηχανικών ανάπτυξης λογισμικού και των υπεύθυνων ασφάλειας στη χρήση τέτοιων εργαλείων και στην αναγνώριση της χρησιμότητας τους για την πραγματοποίηση συχνών έλεγχων των εφαρμογών. Στα πλαίσια επίδειξης και δημιουργίας ενός οδηγού δόμησης και εκτέλεσης ελέγχων ασφάλειας εφαρμογών ιστού, χρησιμοποιήθηκε η δικτυακή εφαρμογή Open eClass [4], διαχείρισης ηλεκτρονικών μαθημάτων. Επιπλέον χρησιμοποιήθηκε το εργαλείο Wireshark [5] για τον εντοπισμό, συλλογή και ανάλυση πληροφοριών σχετικά με τα δικτυακά πακέτα που ανταλλάζονταν μεταξύ του εξυπηρετητή και του πελάτη, καθώς επίσης και το Oracle VitualBox [6] για τη δημιουργία μιας υποδομής ελεγχόμενου περιβάλλοντος. Να σημειωθεί ότι επιπλέον του παραπάνω ελέγχου αξιολογήθηκε και η ασφάλεια του δικτυακού τόπου (website) του Τμήματος Ψηφιακών Συστημάτων του Πανεπιστημίου Πειραιώς (www.ds.unipi.gr) και όλων των εφαρμογών που περιέχονται σε αυτό. Δεδομένης της ευαισθησίας των αποτελεσμάτων, αυτά συμπεριλήφθηκαν σε εμπιστευτικό παράρτημα της παρούσας διπλωματικής εργασίας.

Από την διπλωματική αυτή εξάγονται τα ακόλουθα συμπεράσματα:

- Η μέθοδος της δοκιμής διείσδυσης με την τεχνική Black Box, αποτελεί την αποτελεσματικότερη μέθοδο για την αξιολόγηση της ασφάλειας των εφαρμογών ιστού, αφού εκτιμά το επίπεδο ασφάλειας ακολουθώντας τεχνικές και βήματα πανομοιότυπα με αυτά που χρησιμοποιούν οι εισβολείς για την πραγματοποίηση των επιθέσεών τους.

- Η χρήση εργαλείων για την πραγματοποίηση των ελέγχων είναι αναγκαία, είτε πρόκειται για αυτοματοποιημένους, είτε όχι, λόγω της απαίτησης για συνεχείς και επαναλαμβανόμενους ελέγχους.

[3] Tenable Inc., http://www.tenable.com/
[4] Open eClass, http://www.openeclass.org/
[5] Wireshark, http://www.wireshark.org/
[6] Oracle VirtualBox, https://www.virtualbox.org/

- Η χειροκίνητη συλλογή πληροφοριών και λεπτομερειών σχετικά με τις εφαρμογές αποτελεί το κλειδί για την βελτιστοποίηση της διεξαγωγής των ελέγχων από τα εργαλεία.

Από την όλη προσπάθεια συμπεραίνουμε ότι τα εργαλεία αυτά πραγματικά κάνουν ευκολότερη τη διαδικασία του ελέγχου των εφαρμογών ιστού, αλλά δεν πρέπει σε καμία περίπτωση να θεωρούνται πανάκεια για την επίλυση στον εντοπισμό των προβλημάτων ασφάλειας των εφαρμογών αυτών. Αυτό είναι ίσως και το πιο σημαντικό στοιχείο, που πρέπει ο υπεύθυνος ασφάλειας να έχει κατά νου κατά τη διάρκεια των ελέγχων.

Η δομή της εργασίας έχει ως εξής: Στο **Κεφάλαιο 1 (Θεωρητικό Υπόβαθρο)** γίνεται μια αναφορά σε βασικούς ορισμούς ασφάλειας και στις σημαντικότερες ευπάθειες των εφαρμογών ιστού. Στο **Κεφάλαιο 2 (Παρουσίαση του Nessus)**, γίνεται μια εκτενέστερη παρουσίαση των λειτουργιών και δυνατοτήτων του εργαλείου που χρησιμοποιείται για τους ελέγχους. Στο **Κεφάλαιο 3 (Παραμετροποίηση και Ανάλυση Στοιχείων)** παρουσιάζεται η χρήση του εργαλείου για τη μέτρηση ασφάλειας εφαρμογών, παραθέτοντας ένα οδηγό δημιουργίας πολιτικών ελέγχου (διαμόρφωσης ρυθμίσεων ελέγχου). Η λογική που ακολουθείται εδώ είναι η παρουσίαση και δημιουργία μιας γενικευμένης πολιτικής που μπορεί να χρησιμοποιηθεί ως η αρχική "πρώτου επιπέδου" σάρωση σε κάποια εφαρμογή, αλλά και που λειτουργεί επίσης, ως βάση για τη δημιουργία εξειδικευμένων πολιτικών αργότερα, για συγκεκριμένες εφαρμογές, καθώς επίσης και η δημιουργία, τελικά, μιας εξειδικευμένης πολιτικής με τα βήματα που την απαρτίζουν. Στο **Κεφάλαιο 4 (Αποτελέσματα)** γίνεται η ανασκόπηση των αποτελεσμάτων που παράγει το Nessus μαζί με την αντιστοίχιση αυτών, με τη λίστα ευπαθειών του έργου OWASP Top 10 και παρουσιάζονται τα επιπλέον στοιχεία που προέκυψαν από τη διαδικασία των ελέγχων με το μεγαλύτερο ενδιαφέρον. Στο **Κεφάλαιο 5 (Οδηγίες κατάρτισης πολιτικών)** παρουσιάζονται οι κατευθυντήριες γραμμές που πρέπει να ακολουθηθούν για τη δημιουργία βέλτιστων πολιτικών σάρωσης ώστε να παραχθούν τα καλύτερα δυνατά αποτελέσματα. Τέλος στο **Κεφάλαιο 6 (Σύνοψη και Συμπεράσματα)** περιγράφονται αναλυτικά τα συμπεράσματα που εξήχθησαν από την εκπόνηση της συγκεκριμένης διπλωματικής εργασίας.

Κεφάλαιο 1 (Θεωρητικό Υπόβαθρο)

1.1. Εισαγωγή

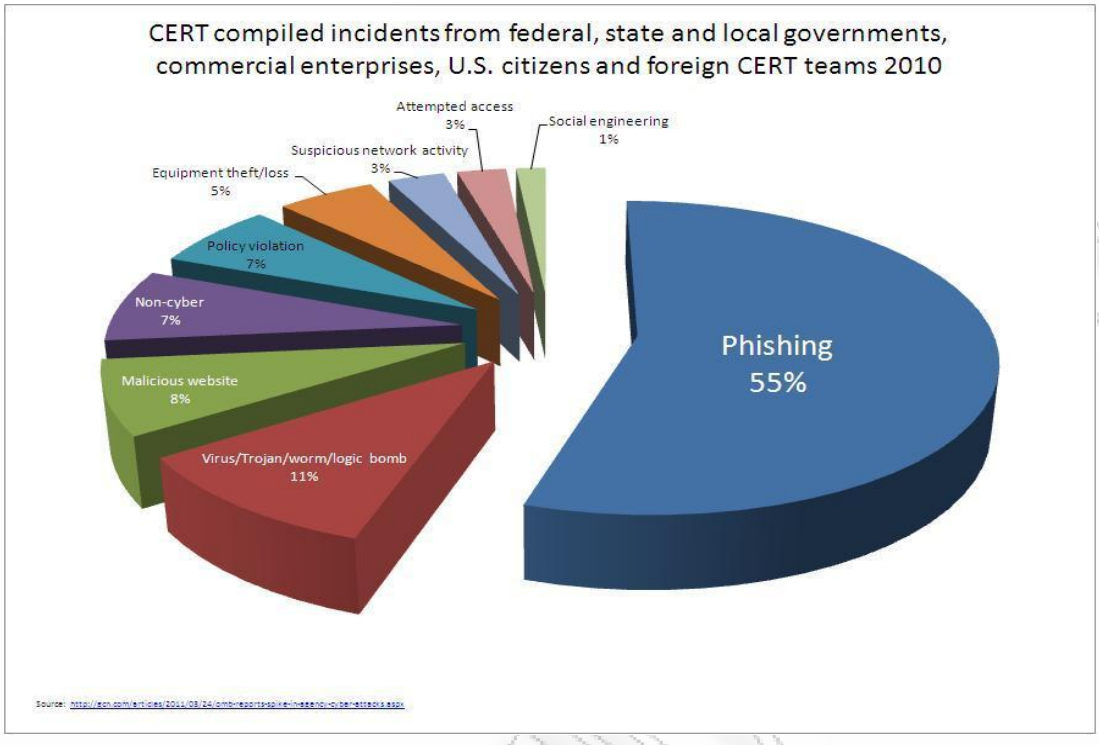
Καθώς το Internet αναπτύσσεται και εξελίσσεται συνεχώς, όλο και περισσότεροι οργανισμοί και επιχειρήσεις αντικαθιστούν τους απλούς ιστότοπους με κρίσιμες εφαρμογές σχεδιασμένες με τέτοιο τρόπο, που να επιτυγχάνουν την ενοποίηση/ενσωμάτωση των ήδη υπαρχόντων συστημάτων τους και την παροχή καλύτερων υπηρεσιών προς τους πελάτες τους. Έτσι λοιπόν σήμερα υπάρχει πληθώρα εμπορικών ιστοσελίδων για αγορά προϊόντων και παροχή κάθε είδους υπηρεσιών, στις οποίες λειτουργούν εφαρμογές ιστού και υπηρεσίες, οι οποίες είναι οι βασικοί συντελεστές του Internet επόμενης γενιάς ή αλλιώς Internet2. Το Internet2 είναι μια νέα σειρά τεχνολογιών υποδομής και εφαρμογών, που έχουν ως σκοπό την ανάπτυξη εξελιγμένων εφαρμογών, που θα καθιστούν ικανή τη συνεργασία μεταξύ ανθρώπων και θα παρέχουν αλληλεπιδραστική πρόσβαση σε πληροφορίες και πηγές με τέτοιο τρόπο, ο οποίος δεν είναι εφικτός με την τωρινή μορφή του Internet.

Το Internet αρχικά σχεδιάστηκε με κύριο στόχο να είναι «ανοικτό». Αυτό σήμαινε ότι ο καθένας θα μπορούσε να έχει δυνατότητα πρόσβασης σε πληροφορίες, που δημοσιεύονται στο internet και για αυτό το λόγο δεν θα υπήρχε η ανάγκη προστασίας τους. Η φύση όμως των δεδομένων και καθώς και το περιεχόμενό τους πολλές φορές καθιστούν απαραίτητη την προστασία τους όταν αυτά μπορούν να γίνουν εύκολα προσπελάσιμα μέσω του παγκόσμιου ιστού. Γι’ αυτό το λόγο οι περισσότεροι οργανισμοί και επιχειρήσεις, που έχουν εγκαταστήσει και λειτουργούν εφαρμογές ιστού, ανακαλύπτουν συνεχώς ότι είναι εκτεθειμένοι σε νέους κινδύνους που αφορούν τη σωστή λειτουργία των συστημάτων τους και κυρίως την ασφάλεια αυτών, με συνέπεια τις επιπτώσεις σχετικά με την ιδιωτικότητα των χρηστών και την προστασία των αγαθών.

Αυτός ο φόβος είναι δικαιολογημένος, αν εξετάσει κάποιος τα στατιστικά στοιχεία του Συντονιστικού Κέντρου CERT - Computer Emergency Response Team Coordination Center, ενός οργανισμού που χρηματοδοτείται από την κυβέρνηση των Ηνωμένων Πολιτειών, και ο οποίος έχει ως σκοπό την πληροφόρηση επί θεμάτων ασφάλειας στην κοινωνία του διαδικτύου (internet) και την ενίσχυση της άμεσης και αποτελεσματικής συνεργασίας μεταξύ ειδικών σε καταστάσεις επείγουσας ανάγκης - σύμφωνα με τα οποία οι κυβερνοεπιθέσεις στις υπηρεσίες και τους οργανισμούς αυξάνονται όσο η ετοιμότητα τους σε θέματα ασφάλειας υστερεί.

Η έκθεση του Γραφείου Διαχείρισης και Προϋπολογισμού αναφέρει ότι τα περιστατικά στον κυβερνοχώρο αυξήθηκαν 39 τοις εκατό κατά το τελευταίο οικονομικό έτος χρήσης. Οι κρατικές υπηρεσίες παρατήρησαν μια απότομη αύξηση κατά το τελευταίο οικονομικό έτος χρήσης στα περιστατικά ασφάλειας που συμβαίνουν στον κυβερνοχώρο , τα οποία αυξήθηκαν κατά 39 τοις εκατό το 2009, σύμφωνα με την ετήσια έκθεση από το Γραφείο Διαχείρισης και Προϋπολογισμού. Τριάντα τοις εκατό αυτών των περιστατικών ήταν κακόβουλες επιθέσεις κώδικα. Στην ετήσια έκθεση του Γραφείου Διαχείρισης και Προϋπολογισμού σχετικά με την εφαρμογή του ομοσπονδιακού Νόμου του 2002 περί της Διαχείρισης Ασφάλειας Πληροφοριών αναφέρθηκαν 41.776 ομοσπονδιακά περιστατικά σε 24 γραφεία το 2010, σε σύγκριση με 30.000 περιστατικά το 2009. "Κακόβουλος κώδικας με πολλαπλά μέσα (π.χ. phishing, ιοί, λογική βόμβα) εξακολουθεί να αποτελεί την πλέον διαδεδομένη προσέγγιση επίθεσης" όπως αναφέρει η έκθεση.

Εκτός από κακόβουλο κώδικα, σχεδόν το 14 τοις εκατό που εμπλέκεται με μη εξουσιοδοτημένη πρόσβαση, το 18 τοις εκατό με ανάρμοστη χρήση και το 27 τοις εκατό παρατίθεται ως υπό έρευνα ή κάτι άλλο. Έντεκα τοις εκατό που περιλαμβάνει σαρώσεις, ανιχνεύσεις και προσπάθεια πρόσβασης και 0,1 τοις εκατό ήταν denial of service επιθέσεις. Η Ομάδα Πληροφορικής Επείγουσας Ετοιμότητας των ΗΠΑ(US-CERT) κατάρτισε περιστατικά από ομοσπονδιακές, πολιτειακές και τοπικές κυβερνήσεις, εμπορικές επιχειρήσεις, πολίτες των ΗΠΑ και ξένες ομάδες CERT. Το 2010, ο οργανισμός έλαβε συνολικά 107.439 αναφορές και 108.710 το 2009. Περίπου το 53 τοις εκατό των αναφερθέντων περιστατικών το 2010 ήταν phishing επιθέσεις. Η κατανομή των εκθέσεων του US-CERT φαίνεται στην εικόνα παρακάτω:



Εικόνα 1: Η κατανομή των εκθέσεων με βάση τα αποτελέσματα από το US-CERT

- Phishing: 56,579 περιστατικά (52.7 %)
- Ιοί/Δούρειος Ίππος (Trojan)/Σκουλήκια (worms)/λογικές βόμβες (logic bombs): 11,001 περιστατικά (10.2%).
- Κακόβουλη Ιστοσελίδα: 7,971 περιστατικά (7.4%)
- Μη-κυβερνοχώρου (Non-cyber): 7,741 περιστατικά (7.2%)
- Παραβίαση Πολιτικής Ασφάλειας: 6,888 περιστατικά (6.4%)
- Κλοπή Εξοπλισμού/Απώλεια: 5,395 περιστατικά (5%)
- Ύποπτη δικτυακή δραστηριότητα: 3,121 περιστατικά (2.9%)
- Προσπάθειες ανάκτησης πρόσβασης: 2,712 περιστατικά (2.5%)
- Social engineering: 1,571 περιστατικά (1.5 %)

Παρά το γεγονός ότι μόνο το 1,5 τοις εκατό των συνολικών αναφερθεισών επιθέσεων περιελάμβαναν social engineering, η έκθεση του Γραφείου Διαχείρισης και Προϋπολογισμού σημειώνει ότι οι επιθέσεις αυτές είναι που προκαλούν ιδιαίτερη ανησυχία, καθώς εκμεταλλεύονται κώδικες που συχνά γίνονται διαθέσιμοι στο κοινό. "Υπήρξαν επανειλημμένες επιθέσεις σε zero-day ευπάθειες (άγνωστες ευπάθειες) μέσω του social engineering. Αυτές οι επιθέσεις συνήθως απαιτούν social engineering για να ξεγελάσουν τους χρήστες να επισκεφτούν επικίνδυνες ιστοσελίδες που φιλοξενούν κακόβουλα προγράμματα ή να ανοίξουν ένα κακόβουλο επισυνημμένο αρχείο για την

εκτέλεση του κακόβουλου λογισμικού στο δικό τους περιβάλλον”, αναφέρει η έκθεση. Επιπρόσθετα, η έκθεση διαπίστωσε «φτωχή» ομοσπονδιακή συμμόρφωση με τις κατευθυντήριες γραμμές της ασφάλειας πληροφοριών. "Μόνο μία υπηρεσία έλαβε βαθμολογία συμμόρφωσης 100 τοις εκατό για το πρόγραμμα ασφάλειας των πληροφοριών της, η οποία, με βάση την αναθεώρηση του IG της, συνάντησε όλα τα 62 χαρακτηριστικά", αναφέρει η έκθεση. “Οι υπόλοιπες υπηρεσίες είχαν τουλάχιστον μία περιοχή που χρειάζεται βελτίωση. Τρεις οργανισμοί δεν είχαν ούτε ένα πρόγραμμα ασφάλειας του κυβερνοχώρου σε ισχύ για έναν τομέα της ασφάλειας και ένα γραφείο δεν είχε έστω ένα πρόγραμμα σε ισχύ για δύο τομείς της ασφάλειας".

Το Γραφείο Διαχείρισης και Προϋπολογισμού χρησιμοποίησε τους τομείς της ανεπάρκειας για να υπολογίσει τις βαθμολογίες των οργανισμών. Έξι οργανισμοί πέτυχαν βαθμολογία πάνω από 90 τοις εκατό της συμμόρφωσης, οκτώ στο 65 με 90 τοις εκατό, και οι υπόλοιποι εννέα πέτυχαν βαθμολογία μικρότερη του 65 τοις εκατό, σύμφωνα με την έκθεση. Η έκθεση διαπίστωσε επίσης ότι πολλοί οργανισμοί δεν εκπαιδεύουν επαρκώς το προσωπικό τους στην ασφάλεια στον κυβερνοχώρο. Εξειδικευμένη εκπαίδευση ασφάλειας στον κυβερνοχώρο για τους χρήστες μιας υπηρεσίας με σημαντικές ευθύνες για την ασφάλεια των πληροφοριών υπολογίστηκε κατά μέσο όρο στο 88 τοις εκατό. Μία υπηρεσία παρείχε μόνο στο 2 τοις εκατό των χρηστών της τέτοιου είδους εκπαίδευση. Για τους νέους εργαζομένους, κατά μέσο όρο 73 τοις εκατό δόθηκε κατάρτιση για την ευαισθητοποίηση σε θέματα ασφάλειας πριν τους επιτραπεί η πρόσβαση στο δίκτυο. Δύο οργανισμοί δεν εκπαίδευσαν κανέναν από τους νέους υπαλλήλους τους, δύο εκπαίδευσαν μεταξύ πέντε και έξι τοις εκατό των υπαλλήλων τους και μία υπηρεσία εκπαίδευσε το ήμισυ περίπου του εισερχόμενου προσωπικού της κατά 55 τοις εκατό. Η έκθεση του Γραφείου Διαχείρισης και Προϋπολογισμού απηχεί σε παλαιότερες εκθέσεις και αναφορές για την κατάσταση της ασφάλειας του κυβερνοχώρου στην κυβέρνηση. Ο Gregory Wilshusen, διευθυντής του Γραφείου Κυβερνητικής Ευθύνης επί των θεμάτων ασφάλειας των πληροφοριών, δήλωσε πρόσφατα σε υποεπιτροπή εσωτερικής ασφάλειας ότι οι Ηνωμένες Πολιτείες δεν είναι έτοιμες για «εν δυνάμει καταστροφικές» επιθέσεις στον κυβερνοχώρο.

Ο Διευθυντής συστημάτων πληροφορικής του τμήματος Υποθέσεων Απόστρατων Roger Baker πρότεινε τη βελτίωση της ασφάλειας με τη συγκέντρωση της διοίκησης, δηλώνοντας ότι η κυβέρνηση υστερεί σε σχέση με τον ιδιωτικό τομέα σε αυτό το θέμα εξαιτίας της οργάνωσής της. Ο Baker μίλησε σε μια συνάντηση κορυφής για την ασφάλεια του κυβερνοχώρου στην Ουάσιγκτον που φιλοξενήθηκε από το FedScoop. Και τον Ιανουάριο το Εθνικό Ινστιτούτο Ασφάλειας Κυβερνοχώρου απέδωσε τους βαθμούς στην

κυβέρνηση Ομπάμα οι οποίοι κυμαίνονταν από Β έως D για τις πολιτικές ασφάλειας του κυβερνοχώρου της. Τα τμήματα που καλύπτονται στην έκθεση περιλαμβάνουν αυτά της Γεωργίας, του Εμπορίου, της Εθνικής Άμυνας, της Παιδείας, της Υγείας και Ανθρωπίνων Υπηρεσιών, της Εσωτερικής Ασφάλειας, του Οικισμού και Αστικής Ανάπτυξης, των Εσωτερικών, της Δικαιοσύνης, της Εργασίας, του Δημοσίου, του Υπουργείου Οικονομικών, των Μεταφορών, των Υποθέσεων Απόστρατων, της Υπηρεσίας Προστασίας του Περιβάλλοντος, της Γενικής Διοίκησης Υπηρεσιών, της Εθνικής Υπηρεσία Αεροναυτικής και Διαστήματος, του Εθνικού Ιδρύματος Επιστημών, της Πυρηνικής Ρυθμιστικής Επιτροπής, του Γραφείου διαχείρισης Προσωπικού, της Διοίκησης Μικρών Επιχειρήσεων, της Διοίκησης Κοινωνικής Ασφάλισης και της Υπηρεσίας Διεθνούς Ανάπτυξης των Ηνωμένων Πολιτειών.

Οι εφαρμογές ιστού, που περιέχουν ευπάθειες (vulnerabilities), είναι στις περισσότερες περιπτώσεις ο αδύναμος κρίκος στην αλυσίδα ασφάλειας των υπολογιστικών συστημάτων επιτρέποντας με αυτόν τον τρόπο να συμβαίνουν παραβιάσεις από κακόβουλους χρήστες (hackers). Οι ευπάθειες αυτές συνήθως υπάρχουν τόσο στην αρχιτεκτονική όσο και στη διαμόρφωση των συστημάτων, καθώς και στο σχεδιασμό των εφαρμογών, στη διαμόρφωση εγκατάστασης και στη διαχείριση των εφαρμογών. Οι κίνδυνοι από την ύπαρξη αυτών των ευπαθειών μπορεί να είναι πολύ μεγάλοι. Για αυτό το λόγο, οι υπεύθυνοι ανάπτυξης των εφαρμογών, αλλά και οι υπεύθυνοι ασφαλείας των οργανισμών-επιχειρήσεων που τις χρησιμοποιούν, οφείλουν να είναι ικανοί στο να ανιχνεύουν την ύπαρξη αλλά και τη σοβαρότητα των ευπαθειών, καθώς και να προτείνουν τα κατάλληλα αντίμετρα για την προστασία των εφαρμογών τους. Αυτό βέβαια προϋποθέτει ότι έχουν το κατάλληλο γνωστικό υπόβαθρο σε θέματα ασφάλειας, ώστε να μπορούν να ανιχνεύουν τις ευπάθειες, αλλά και τα κατάλληλα εργαλεία για να μπορούν να κάνουν τους απαραίτητους ελέγχους γρήγορα και σωστά, χωρίς να επηρεάζεται η εύρυθμη λειτουργία των οργανισμών-επιχειρήσεων τους. Στόχος, λοιπόν αυτού του κεφαλαίου είναι η εξοικείωση με τις σχετικές έννοιες και το κανονιστικό πλαίσιο σε θέματα ασφάλειας, που έχει ήδη ορισθεί από αναγνωρισμένους οργανισμούς, όπως είναι ο CERT και ο OWASP (Open Web Application Security Project – ένας παγκόσμιος οργανισμός που έχει ως σκοπό την βελτίωση της ασφάλειας του λογισμικού εφαρμογών) κ.α., καθώς και η μελέτη των οδηγιών και προτύπων ασφάλειας για την ανάπτυξη και έλεγχο των εφαρμογών ιστού.

1.2. Ορισμοί

1.2.1. Επίθεση

Επιθέσεις (attacks) ονομάζονται οι τεχνικές, που χρησιμοποιούν οι επίδοξοι εισβολείς (intruders) για να εκμεταλλευθούν τις ευπάθειες των διαφόρων εφαρμογών. Οι επιθέσεις αυτές συχνά συγχέονται με τις ευπάθειες των εφαρμογών. Για το λόγο αυτό οφείλουμε να διευκρινίσουμε ότι επίθεση είναι μια πράξη, την οποία ο εισβολέας κάνει σε μια εφαρμογή και δεν είναι μια αδυναμία αυτής.

1.2.2. Απειλή

Απειλή (threat) είναι η ένδειξη του ότι επίκειται κάποιος κίνδυνος ή κάποιο κακό για την εφαρμογή ή το σύστημα γενικότερα. Είναι ο πιθανός κίνδυνος μιας επικείμενης επίθεσης που μπορεί να βλάψει την εφαρμογή. Οποιαδήποτε περίπτωση ή γεγονός με δυνατότητα πρόκλησης ζημιάς σε ένα σύστημα υπό μορφή καταστροφής, κοινοποίησης, τροποποίησης των στοιχείων του, ή/και άρνησης της υπηρεσίας.

1.2.3. Ευπάθεια

Ευπάθεια (vulnerability) ονομάζεται μια “τρύπα” ή μια αδυναμία της εφαρμογής, η οποία μπορεί να οφείλεται σε ένα τρωτό σημείο στη σχεδίαση ή ένα σφάλμα υλοποίησης. Αυτή επιτρέπει σε έναν επιτιθέμενο να βλάψει τους ιδιοκτήτες και τους νόμιμους χρήστες της εφαρμογής, ή άλλες οντότητες, που βασίζονται στην εφαρμογή. Ο όρος «ευπάθεια» πολύ συχνά χρησιμοποιείται εσφαλμένα. Πρέπει να διακρίνεται από τους όρους threat (απειλή), attack (επίθεση) και countermeasures (αντίμετρα).

1.2.4. Αντίμετρα

Τα αντίμετρα (countermeasures) περιλαμβάνουν τεχνολογίες ή μοντέλα άμυνας, που χρησιμοποιούνται με σκοπό να ανιχνεύσουν ή/και να αποτρέψουν επιθέσεις. Τα αναγκαία αντίμετρα σε μια εφαρμογή πρέπει να αναγνωρισθούν με τη χρήση της ανάλυσης κινδύνου (Risk Assessment), έτσι ώστε να διασφαλιστεί ότι η εφαρμογή προστατεύεται από κοινούς τύπους επιθέσεων. Μια αδυναμία ή μια ρωγμή στη σχεδίαση ενός αντιμέτρου ή η έλλειψη του αναγκαίου αντιμέτρου, έχει ως αποτέλεσμα μια ευπάθεια, η οποία μπορεί να είναι σε θέση να καταστήσει την εφαρμογή ευάλωτη σε επιθέσεις.

1.2.5. Δοκιμή διείσδυσης

Δοκιμή διείσδυσης (penetration testing) ονομάζεται η μέθοδος εκτίμησης της ασφάλειας ενός υπολογιστικού συστήματος ή δικτύου, με τη μέθοδο της εξομοίωσης μιας επίθεσης. Η δοκιμή διείσδυσης σε μια εφαρμογή ιστού επικεντρώνεται μόνο στην εκτίμηση της ασφάλειας της εφαρμογής ιστού. Η διαδικασία περιλαμβάνει μια ενεργή ανάλυση της εφαρμογής για κάθε είδους αδυναμίες, τεχνικές ρωγμές ή ευπάθειες. Κάθε πρόβλημα ασφάλειας, που ανακαλύπτεται, πρέπει να παρουσιάζεται στον ιδιοκτήτη της εφαρμογής μαζί με τον καθορισμό του αντίκτυπού του και συνήθως με μια πρόταση για τον περιορισμό του ή μια τεχνική λύση αντιμετώπισής του

1.2.6. Τι είναι ο έλεγχος (testing)

Τι εννοούμε όταν μιλάμε για έλεγχο (testing) μιας εφαρμογής; Κατά τη διάρκεια ανάπτυξης του κύκλου ζωής μίας δικτυακής εφαρμογής (web application) πολλά πράγματα χρειάζεται να εξεταστούν. Το λεξικό του Meriam-Webster περιγράφει το testing σαν:

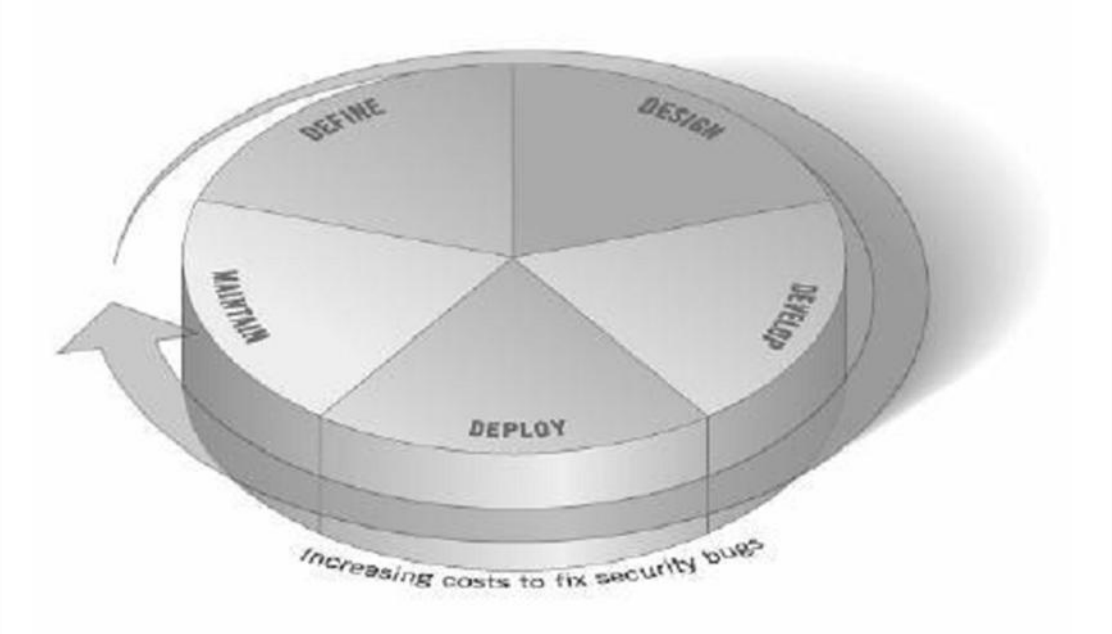
- Το να τίθεται κάτι σε δοκιμή ή απόδειξη
- Να υποβάλλεται σε δοκιμή
- Να διατίθεται διαρκώς ή για αξιολόγηση βασισμένη σε δοκιμές

Για τους σκοπούς αυτού του κειμένου, ο έλεγχος (testing) σημαίνει, μία διαδικασία σύγκρισης της κατάστασης ενός συστήματος/εφαρμογής απέναντι σε ένα σύνολο κριτηρίων. Στη βιομηχανία της ασφάλειας των δεδομένων, οι άνθρωποι συχνά “τεστάρουν” σε σχέση με ένα σύνολο διανοητικών κριτηρίων που δεν είναι ούτε καλά ορισμένα ούτε και ολοκληρωμένα. Γι’ αυτό το λόγο, αλλά και άλλους, πολλοί που είναι έξω από το χώρο της ασφάλειας θεωρούν τον έλεγχο της ασφάλειας κάτι σαν μαύρη μαγεία. Ο σκοπός αυτού του κειμένου είναι να αλλάξει αυτή η αντίληψη και να κάνει ευκολότερο για ανθρώπους χωρίς γνώση σε βάθος της ασφάλειας, να κατανοήσουν τις διαδικασίες ελέγχου.

1.2.7. Πότε πραγματοποιούμε έναν έλεγχο

Οι περισσότεροι προγραμματιστές στις μέρες μας δεν ελέγχουν το λογισμικό τους μέχρι να δημιουργηθεί και να βρίσκεται στην φάση της ανάπτυξης του κύκλου ζωής του (πχ ο κώδικας έχει δημιουργηθεί και αποτελεί μία διαδικτυακή εφαρμογή που βρίσκεται σε λειτουργία). Αυτή γενικά είναι μία πολύ αναποτελεσματική και με απαγορευτικό κόστος πρακτική. Μία από τις καλύτερες μεθόδους ώστε να αποτρέψουμε να εμφανιστούν σφάλματα κατά την παραγωγή-ανάπτυξη εφαρμογών είναι να βελτιώσουμε τον Κύκλο

Ζωής Ανάπτυξης του Λογισμικού (Software Development Life Cycle) συμπεριλαμβάνοντας και το κομμάτι της ασφάλειας σε κάθε μία από τις φάσεις του. Ο Κύκλος Ζωής Ανάπτυξης του Λογισμικού (SDLC) είναι μία δομή που επιβάλλεται κατά την ανάπτυξη των δημιουργημάτων του λογισμικού. Η εικόνα που ακολουθεί δείχνει ένα γενικό μοντέλο ενός Κύκλου Ζωής Ανάπτυξης του Λογισμικού καθώς και την (κατ' εκτίμηση) αύξηση του κόστους της διόρθωσης των σφαλμάτων ασφάλειας σε ένα τέτοιο μοντέλο. Οι εταιρείες θα πρέπει να ελέγχουν το συνολικό Κύκλο Ζωής Ανάπτυξης του Λογισμικού για να εξασφαλίσουν ότι η ασφάλεια αποτελεί αναπόσπαστο κομμάτι της διαδικασίας ανάπτυξης. Ο Κύκλος Ζωής Ανάπτυξης του Λογισμικού θα πρέπει να περιλαμβάνουν ελέγχους ασφάλειας που να εξασφαλίζουν ότι η ασφάλεια καλύπτεται επαρκώς και πως οι έλεγχοι είναι αποτελεσματικοί καθ' όλη τη διαδικασία ανάπτυξης



Εικόνα 2: Κύκλος ζωής ανάπτυξης λογισμικού

1.2.8. Τι πρέπει να ελέγξουμε

Θα ήταν χρήσιμο να σκεφτούμε τη διαδικασία ανάπτυξης λογισμικού σαν ένα συνδυασμό ανθρώπων, διαδικασιών και τεχνολογίας. Αν αυτοί είναι οι παράγοντες που δημιουργούν το λογισμικό, τότε είναι λογικό πως αυτοί είναι και οι παράγοντες που πρέπει να ελεγχθούν. Σήμερα οι περισσότεροι άνθρωποι γενικά ελέγχουν την τεχνολογία ή το λογισμικό αυτό καθ' αυτό. Ένας αποτελεσματικός έλεγχος προγράμματος θα πρέπει να περιέχει στοιχεία που ελέγχει **Ανθρώπους** - ώστε να διασφαλίσει ότι υπάρχει επαρκής εκπαίδευση και ευαισθητοποίηση – **Διαδικασίες** – για να διαβεβαιώσει πως υπάρχουν οι κατάλληλες πολιτικές και πρότυπα και οι άνθρωποι ξέρουν πώς να ακολουθήσουν αυτές τις

πολιτικές – **Τεχνολογία** – για να διασφαλίσει ότι η διαδικασία έχει υπάρξει αποτελεσματική στην υλοποίησή της. Εξαιρώντας την υιοθέτηση μιας ολιστικής προσέγγισης και ελέγχοντας μόνο την τεχνική υλοποίηση μιας εφαρμογής δεν θα αποκαλυφθούν διοικητικές ή λειτουργικές ευπάθειες που θα μπορούσαν να υπάρχουν. Ελέγχοντας τους ανθρώπους, τις πολιτικές και τις διαδικασίες ένας οργανισμός μπορεί να ανακαλύψει θέματα που θα εμφανιστούν αργότερα σαν ελαττώματα στην τεχνολογία όπως η εξάλειψη των σφαλμάτων εξ' αρχής και ο προσδιορισμός των βασικών αιτιών των ελαττωμάτων. Ομοίως ελέγχοντας μόνο κάποια από τα τεχνικά θέματα που περιέχονται σε ένα σύστημα θα έχει σαν αποτέλεσμα την ελλιπή και την εσφαλμένη εκτίμηση της κατάστασης της ασφάλειας. Ο Dennis Verdon επικεφαλής της ασφάλειας πληροφοριών του Fidelity National Financial [7] παρουσίασε μία εξαιρετική παρομοίωση για αυτή την παρανόηση στο συνέδριο AppSEc 2004 του OWASP [8] στη Νέα Υόρκη. *«Εάν τα αυτοκίνητα αναπτύσσονταν σαν εφαρμογές, οι έλεγχοι ασφάλειας θα περιελάμβαναν μόνο τη μετωπική σύγκρουση. Τα αυτοκίνητα δε θα ελέγχονταν ως προς την ολίσθησή τους ή για την σταθερότητά τους σε ελιγμούς έκτακτης ανάγκης, για την αποτελεσματικότητα των φρένων, την πλευρική πρόσκρουση και την αντίσταση στην κλοπή»*.

1.3. Προβλήματα ασφάλειας δικτυακών εφαρμογών

Ο OWASP (Open Web Application Security Project), ο οποίος αποτελεί έναν παγκόσμιο οργανισμό που έχει ως σκοπό την βελτίωση της ασφάλειας του λογισμικού των εφαρμογών ιστού και όχι μόνο, δημοσίευσε για πρώτη φορά τις κατευθυντήριες γραμμές ελέγχου των εφαρμογών ιστού (web applications) το 2004, οι οποίες στη συνέχεια ενημερώθηκαν εκ νέου για δεύτερη φορά το 2007 και ξανά για τρίτη φορά το 2010. Οι κατευθυντήριες γραμμές του OWASP χαρακτηρίζονται ως κίνδυνοι με αριθμό προτεραιότητας από το A1 έως A10 και με τη χαρακτηριστική ονομασία του συγκεκριμένου έργου ως "**OWASP Top 10**". Τα σημαντικότερα, λοιπόν, προβλήματα που αντιμετωπίζουν οι εφαρμογές ιστού όσον αφορά την ασφάλεια, σύμφωνα με τον οργανισμό OWASP, αναφέρονται συνοπτικά στους παρακάτω πίνακες που αφορούν και τις τρείς εκδόσεις του:

OWASP Top 10 – 2010		
A1	**Injection**	Αδυναμίες έγχυσης, όπως είναι τα SQL, OS και LDAP injections, συμβαίνουν όταν μη αξιόπιστα δεδομένα

[7] FNF, http://www.fnf.com/
[8] AppSec 2004 OWASP, https://www.owasp.org/index.php/OWASP_AppSec_NYC_2004

		αποστέλλεται για εκτέλεση ως μέρος μιας ήδη υπάρχουσας εντολής ή ερωτήματος. Τα δεδομένα που στέλνει ο εισβολέας μπορεί να ξεγελάσουν την διεργασία ώστε να εκτελέσει εντολές ή να παραχωρήσει πρόσβαση σε δεδομένα που χρειάζονται προηγουμένως εξουσιοδότηση.
A2	**Cross Site Scripting (XSS)**	Οι αδυναμίες XSS παρατηρούνται όταν γίνεται μια αίτηση με μη αξιόπιστα δεδομένα και η εφαρμογή τα αποστέλλει σε ένα web browser χωρίς την κατάλληλη επικύρωση και εισαγωγή χαρακτήρων διαφυγής. Το XSS επιτρέπει στους επιτιθέμενους να εκτελέσουν κώδικα, στον web browser του θύματος μέσω του οποίου μπορούν να κλέψουν συνεδρίες, να παραποιήσουν ιστοσελίδες, ή να ανακατευθύνουν το χρήστη σε κακόβουλες ιστοσελίδες
A3	**Broken Authentication and Session Management**	Αδυναμίες που οι λειτουργίες τους σχετίζονται με τη διαχείριση ταυτότητας και συνεδριών. Λόγω κακής υλοποίησης, επιτρέπουν στους επιτιθέμενους την υποκλοπή κωδικών πρόσβασης, κλειδιών, συνεδριών, και την εκμετάλλευση των αδυναμιών για να προσποιηθούν άλλους χρήστες.
A4	**Insecure Direct Object References**	Αδυναμία που παρουσιάζεται όταν γίνεται μια άμεση αναφορά προς ένα εσωτερικό αντικείμενο της εφαρμογής. Για παράδειγμα, αναφορά σε ένα αρχείο, κατάλογο, ή σε κάποια κλειδιά στη βάση δεδομένων, χωρίς τον κατάλληλο έλεγχο πρόσβασης προηγουμένως ή άλλες μεθόδους προστασίας. Ο επιτιθέμενος μπορεί με αυτό τον τρόπο να εκμεταλλευτεί αυτές τις αναφορές και να αποκτήσει μη πρόσβαση σε μη εξουσιοδοτημένα δεδομένα
A5	**Cross Site Request Forgery (CSRF)**	Μια επίθεση CSRF αναγκάζει το πρόγραμμα περιήγησης ενός αυθεντικοποιημένου χρήστη να στείλει ένα πλαστό HTTP αίτημα, το οποίο φυσικά θα συμπεριλαμβάνει, το

		cookie συνόδου του θύματος και οποιαδήποτε άλλη πληροφορία αυθεντικοποίησης είναι απαραίτητη, σε μια ευπαθή εφαρμογή ιστού. Αυτό επιτρέπει στον επιτιθέμενο να αναγκάσει το πρόγραμμα περιήγησης του θύματος να δημιουργήσει αιτήματα τα οποία η ευπαθής εφαρμογή θεωρεί ότι είναι νόμιμα αιτήματα από το θύμα
A6	**Security Misconfiguration**	Η καλή ασφάλεια απαιτεί τη χρήση μιας καλά ορισμένης και ασφαλούς διαμόρφωσης, για την εφαρμογή, για το πλαίσιο αυτής, για τον application server, για τον web server, για τον server της βάσης δεδομένων, και για την πλατφόρμα. Όλες αυτές οι ρυθμίσεις πρέπει να καθοριστούν, να υλοποιηθούν και να διατηρηθούν καθώς σχεδόν ποτέ δεν διατίθενται ως προεπιλογές. Αυτό περιλαμβάνει ακόμα και την ενημέρωση των λογισμικών, καθώς και όλων των βιβλιοθηκών που χρησιμοποιούνται από την εφαρμογή
A7	**Insecure Cryptographic Storage**	Πολλές δικτυακές εφαρμογές δεν προστατεύουν σωστά τα ευαίσθητα δεδομένα, όπως, τις πιστωτικές κάρτες, τα SSN και τα πιστοποιητικά αναγνώρισης, με την κατάλληλη κρυπτογράφηση ή κατακερματισμό. Οι επιτιθέμενοι μπορούν να κλέψουν ή να τροποποιούν τα δεδομένα αυτά για τη χρήση τους αργότερα σε απάτες με πιστωτικές κάρτες, κλοπές ταυτότητας ή άλλα εγκλήματα.
A8	**Failure to Restrict URL Access**	Πολλές δικτυακές εφαρμογές ελέγχουν τα δικαιώματα προσπέλασης σε κάποιο URL πριν από την αναπαραγωγή του περιεχομένου των διαφόρων συνδέσμων και κουμπιών. Ωστόσο και οι εφαρμογές πρέπει να πραγματοποιούν τέτοιους ελέγχους πρόσβασης, κάθε φορά που προσπαθούν να προσπελάσουν αυτές τις σελίδες, αλλιώς οι εισβολείς θα είναι σε θέση να παραποιήσουν τις διευθύνσεις URL ώστε να αποκτήσουν πρόσβαση σε αυτές τις σελίδες.

A9	**Insufficient Transport Layer Protection**	Πολύ συχνά οι εφαρμογές αποτυγχάνουν σε ότι έχει να κάνει με τη προστασία της εμπιστευτικότητας και ακεραιότητας των ευαίσθητων δεδομένων που κινούνται στο δίκτυο. Και καμιά φορά, όταν γίνεται η προσπάθεια προστασίας των δεδομένων αυτών, χρησιμοποιούνται αδύναμοι αλγόριθμοι, ληγμένα ή μη έμπιστα πιστοποιητικά ή και λάθος χρήση τους.
A10	**Unvalidated Redirects and Forwards**	Συχνά οι εφαρμογές ιστού, ανακατευθύνουν ή/και προωθούν τους χρήστες σε άλλες σελίδες και δικτυακούς τόπους χρησιμοποιώντας μη ασφαλή δεδομένα κατά τον καθορισμό του προορισμού. Χωρίς την κατάλληλη επικύρωση, οι επιτιθέμενοι μπορούν να ανακατευθύνουν τα θύματα σε ιστοσελίδες phishing ή malware, είτε να χρησιμοποιήσουν τη διαδικασία της προώθησης για την πρόσβαση σε μη εξουσιοδοτημένες σελίδες.

Πίνακας 1: Κίνδυνοι ασφάλειας των εφαρμογών ιστού με βάση τον OWASP έκδοση 2010

Στον πίνακα που ακολουθεί περιγράφονται οι αλλαγές, στους υψηλού επιπέδου κινδύνους και ποιες από αυτές καλύπτονται στην λίστα μεταξύ του 2010 και του 2007 μαζί με τις αλλαγές που επήλθαν:

OWASP Top 10 – 2010		vs. OWASP Top 10 – 2007	
A1	Injection	**A2**	Injection Flaws
A2	Cross-Site Scripting (XSS)	**A1**	Cross-Site Scripting (XSS)
A3	Broken Authentication and Session Management	**A7**	Broken Authentication and Session Management
A4	Insecure Direct Object References	**A4**	Insecure Direct Object Reference
A5	Cross-Site Request Forgery (CSRF)	**A5**	Cross-Site Request Forgery (CSRF)
A6	Security Misconfiguration		Insecure Configuration Management
A7	Insecure Cryptographic Storage	**A8**	Insecure Cryptographic Storage

A8	Failure to Restrict URL Access	**A10**	Failure to Restrict URL Access
A9	Insufficient Transport Layer Protection	**A9**	Insecure Communications
A10	Invalidated Redirects and Forwards		Δεν συμπεριλαμβανόταν στην λίστα Top 10 του 2007
	Αφαιρέθηκε από τη λίστα Top 10 του 2010	**A3**	Malicious File Execution
	Αφαιρέθηκε από τη λίστα Top 10 του 2010	**A6**	Information Leakage and Improper Error Handling

Πίνακας 2: Σύγκριση των Top 10 ευπαθειών για το 2010 και το 2007 με βάση τον OWASP

Στη συνέχεια παρατίθεται επίσης και ο πίνακας σύγκρισης, με τις αλλαγές στους υψηλού επιπέδου κινδύνους και ποιες από αυτές καλύπτονται στη λίστα μεταξύ του 2007 και του 2004:

OWASP Top 10 – 2007		vs.	OWASP Top 10 – 2004
A1	Cross-Site Scripting (XSS)	**A4**	Cross Site scripting
A2	Injection Flaws	**A6**	Injection Flaws
A3	Malicious File Execution		Δεν συμπεριλαμβανόταν στην λίστα Top 10 του 2004
A4	Insecure Direct Object Reference	**A2**	Broken Access Control (διαχωρίστηκε στο Top 10 - 2007)
A5	Cross-Site Request Forgery (CSRF)		Δεν συμπεριλαμβανόταν στην λίστα Top 10 του 2004
A6	Information Leakage and Improper Error Handling	**A7**	Improper Error Handling
A7	Broken Authentication and Session Management	**A3**	Broken Authentication and Session Management
A8	Insecure Cryptographic Storage	**A8**	Insecure Storage
A9	Insecure Communications		Εκτός δεκάδας αλλά είχε εξεταστεί ως

			θέμα κάτω από το A10
A10	Failure to Restrict URL Access	**A2**	Broken Access Control (διαχωρίστηκε στο Top 10 - 2007)
	Αφαιρέθηκε από τη λίστα Top 10 του 2007	**A1**	Unvalidated Input
	Αφαιρέθηκε από τη λίστα Top 10 του 2007	**A5**	Buffer Overflows
	Αφαιρέθηκε από τη λίστα Top 10 του 2007	**A9**	Denial of Service
	Αφαιρέθηκε από τη λίστα Top 10 του 2007	**A10**	Insecure Configuration Management

Πίνακας 3: Σύγκριση των Top 10 ευπαθειών για το 2007 και το 2004 με βάση τον OWASP

1.4. Βασικές μέθοδοι ελέγχου ασφάλειας στις εφαρμογές ιστού.

Ο συνεχής έλεγχος και η μέτρηση της ασφάλειας των εφαρμογών ιστού είναι απαραίτητη προϋπόθεση για τη διατήρηση της ασφάλειας ενός συστήματος, το οποίο συνδέεται άμεσα στο διαδίκτυο. Όμως σε έναν ιστότοπο είναι δυνατό να φιλοξενείται ένα μεγάλο πλήθος εφαρμογών ιστού διαφορετικού τύπου με διαφορετικό λογισμικό. Οι εφαρμογές ιστού κατασκευάζονται σε επίπεδα, από προγράμματα και δεδομένα τα οποία φιλοξενούνται σε πολλαπλούς servers (web servers, application servers, database servers). Για το λόγο αυτό υπάρχουν διάφορες μέθοδοι ελέγχου ασφάλειας των εφαρμογών ιστού. Οι σημαντικότερες και ευρέως χρησιμοποιούμενες μέθοδοι είναι οι ακόλουθες:

- Επιθεώρηση Ασφάλειας (security audit):
  - Ένα σύστημα ελέγχεται με βάση ένα σύνολο από λίστες ελέγχου (checklists), οι οποίες διαμορφώνονται με βάση διεθνή πρότυπα σχετικά με την ασφάλεια, καθώς και κατάλληλες πολιτικές ασφάλειας του οργανισμού, που χρησιμοποιεί την εφαρμογή ιστού.
  - Οι ελεγκτές εκτελούν την εργασία τους μέσα από προσωπικές συνεντεύξεις, ανιχνεύσεις αδυναμιών, εξετάσεις των ρυθμίσεων, αναλύσεις των διαμοιρασμένων πόρων δικτύου και μελέτες των ιστορικών στοιχείων (log files).
- Αυτο-αξιολόγηση Ασφάλειας (security self-assessment):

- Εδώ δεν υπάρχουν συγκεκριμένα standards ως προς τα οποία θα μετρηθεί το σύστημα, αλλά ο στόχος προσδιορίζεται από την περιοχή, που χρειάζεται διερεύνηση και βελτίωση στη θωράκισή της.
- Ξεπερνά τους πίνακες ελέγχου (checklists) και επεκτείνεται σε ένα πιο λεπτομερή έλεγχο για εντοπισμό αδυναμιών, αλλά και σε συστάσεις για επιδιορθώσεις και βελτιώσεις.
- Πλεονέκτημά της είναι η δυνατότητα να οριστούν επίπεδα προτεραιότητας σε κάθε συστατικό που αξιολογείται, έτσι ώστε με την ολοκλήρωσή της να δοθεί μια κατάταξη προτεραιοτήτων στην επιδιόρθωση των ευπαθειών που ανιχνεύθηκαν.

- Δοκιμή Διείσδυσης (“penetration testing” ή "ethical hacking"):
  - Είναι η ελεγχόμενη προσομοίωση μιας επίθεσης προκειμένου να επιτευχθεί ένας προκαθορισμένος στόχος. Επίσης είναι γνωστή και ως εσωτερική επιθεώρηση ασφάλειας (internal security auditing).
  - Σκοπός της είναι να εντοπιστούν συγκεκριμένες πληροφορίες σχετικές με την ύπαρξη γνωστών ευπαθειών και να διερευνηθεί κατά πόσο είναι δυνατόν ένας ξένος, κάνοντας χρήση αυτών των πληροφοριών, να είναι σε θέση να δημιουργήσει προβλήματα στην εφαρμογή ιστού. Δεν έχει σκοπό να εντοπίσει όλες τις ευπάθειες, αλλά να αποδείξει ότι η ασφάλεια του συστήματος μπορεί να διακυβευτεί.
  - Η δοκιμή μπορεί να πραγματοποιηθεί στη βάση μηδενικής γνώσης (zero knowledge) ή με πλήρη γνώση (full knowledge) του συστήματος, που δοκιμάζεται. Χρησιμοποιείται για να καθορίσει την αξιοπιστία και τη δύναμη των μέτρων ασφάλειας, που παίρνουμε.
  - Οι “ethical hackers” προσπαθούν να υιοθετήσουν τις τεχνικές επιθέσεων των hackers, ώστε να μπορέσουν να μετρήσουν το επίπεδο ασφάλειας της εφαρμογής.

1.4.1. Σύγκριση των μεθόδων μέτρηση ασφάλειας

Κάνοντας μια σύγκριση των μεθόδων μέτρησης ασφάλειας των εφαρμογών ιστού μπορούμε να πούμε ότι:

- Για τον έλεγχο της ασφάλειας μιας εφαρμογής ιστού με τις μεθόδους της επιθεώρησης ασφάλειας και της αυτο-αξιολόγησης απαιτείται η μετακίνηση

μιας μεγάλης ομάδας ειδικών ασφαλείας στον τόπο που λειτουργεί ο οργανισμός, του οποίου ελέγχεται η ασφάλεια της εφαρμογής.

- Η ομάδα αυτή πρέπει να έχει στη διάθεσή της τα κατάλληλα checklists, να έχει υψηλή τεχνογνωσία και να είναι άρτια συντονισμένη.
- Για την ενέργεια των ελέγχων απαιτείται πολύς χρόνος, ώστε να ολοκληρωθούν οι συνεντεύξεις, οι επιθεωρήσεις, οι αξιολογήσεις και οι έρευνες στη διάρκεια των οποίων αποκαλύπτεται και διαταράσσεται η λειτουργία του οργανισμού.

Όλα τα παραπάνω σε συνάρτηση με την ανάγκη για συνεχείς και επαναλαμβανόμενους ελέγχους στις εφαρμογές ιστού, καθιστούν τη δοκιμή διείσδυσης, σχεδόν μονόδρομη λύση. Επιπλέον, η δοκιμή διείσδυσης έχει τα ακόλουθα πλεονεκτήματα:

- απαιτείται ελάχιστο προσωπικό και δεν είναι αναγκαία η μετακίνησή του
- παρέχει τη δυνατότητα αυτοματοποίησης
- διαρκεί μικρό χρονικό διάστημα και είναι εύκολα επαναλαμβανόμενη
- δεν απαιτεί τη γνώση σε βάθος της ελεγχόμενης εφαρμογής
- δεν διαταράσσει τη λειτουργία του οργανισμού
- είναι πολύ οικονομικότερη από τις δύο άλλες μεθόδους.

1.4.2. Τεχνικές δοκιμής διείσδυσης

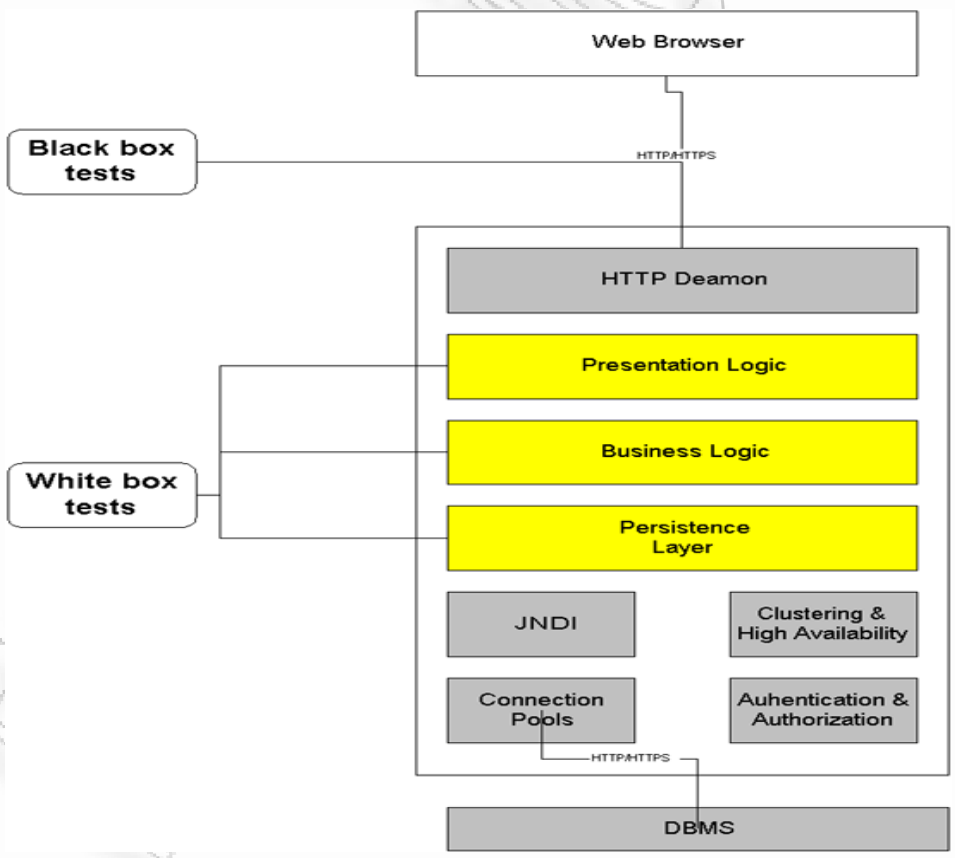
Υπάρχουν δύο κύριες προσεγγίσεις τεχνικών ελέγχου ασφάλειας των εφαρμογών ιστού με τη μέθοδο της δοκιμής διείσδυσης:

- **Χειροκίνητη (manual)**, στην οποία όλη η διαδικασία ελέγχου γίνεται βήμα- βήμα χωρίς την ύπαρξη αυτοματισμών επανάληψης παρόμοιων βημάτων.
- **Αυτοματοποιημένη (automated)**, στην οποία με τη βοήθεια χρήσης εργαλείων αυτοματοποιούνται μερικοί ή όλοι οι έλεγχοι και οι διαδικασίες ελέγχου. Η αυτοματοποιημένη διαδικασία ελέγχου διακρίνεται σε δύο τεχνικές:
  - **Black Box**: ονομάζεται η εφαρμογή δοκιμαστικών δεδομένων που έχουν προέλθει από καθορισμένες λειτουργικές απαιτήσεις χωρίς να λαμβάνουν υπόψη ή να έχουν πρότερη γνώση για τη δομή της εφαρμογής, στην οποία

εφαρμόζονται. Η εφαρμογή εξετάζεται χρησιμοποιώντας την εξωτερική της διεπαφή, ίδια με αυτή, που χρησιμοποιούν οι απλοί χρήστες. Μιμούνται την ακολουθία αλληλεπιδράσεων χρήστη-εφαρμογής και με κάθε αποτυχία να δείχνει ότι ο χρήστης έλαβε ανεπαρκή υπηρεσία.

- **White Box**: ονομάζεται η τεχνική στην οποία εξετάζεται η δομή της εφαρμογής και στη βάση αυτής καθορίζονται τα δεδομένα της δοκιμής. Λαμβάνεται υπόψη και υπάρχει πρότερη γνώση της εσωτερικής δομή της εφαρμογής χρησιμοποιώντας τη διεπαφή προγραμματισμού εφαρμογών (Application Programming Interface), η οποία αποτελεί το μέσο επικοινωνίας των εφαρμογών με τον πυρήνα του λειτουργικού συστήματος ή με βιβλιοθήκες τρίτων κατασκευαστών.

Στην εικόνα 3 παρουσιάζεται η διαφορά στη διεπαφή, που χρησιμοποιεί ο ελεγκτής ασφάλειας για την πραγματοποίηση των ελέγχων του με τις δύο αυτές τεχνικές:



Εικόνα 3: Σχηματική παρουσίαση των Black και White box τεχνικών δοκιμής διείσδυσης σε μια εφαρμογή ιστού

Όπως παρατηρείται από την παραπάνω ανάλυση η μέθοδος της δοκιμής διείσδυσης με την τεχνική Black Box, αποτελεί την αποτελεσματικότερη μέθοδο για την μέτρηση της ασφάλειας των εφαρμογών ιστού, αφού αξιολογεί την ασφάλεια ακολουθώντας τεχνικές

και βήματα πανομοιότυπα με αυτά που χρησιμοποιούν οι επίδοξοι εισβολείς για την πραγματοποίηση των επιθέσεών τους. Για το λόγο αυτό στη συνέχεια θα αναλυθεί και παρουσιαστεί το εργαλείο σάρωσης ασφάλειας δικτύων Nessus, το οποίο αξιοποιεί πλήρως τη συγκεκριμένη μέθοδο και θα αποτελέσει στη συνέχεια το αντικείμενο μελέτης περίπτωσης για την πραγματοποίηση των αυτοματοποιημένων ελέγχων σάρωσης και ανακάλυψης αδυναμιών στις εφαρμογές δικτύου της παρούσας διπλωματικής εργασίας.

Κεφάλαιο 2 (Παρουσίαση του NESSUS)

2.1. Εισαγωγή

Το Nessus 5.0 Vulnerability scanner είναι ιδιοκτησία της Tenable Network Security®. Το εργαλείο Nessus παρέχει υποστήριξη για πληθώρα λειτουργικά συστήματα στα οποία μπορεί να εγκατασταθεί και λειτουργήσει, όπως λειτουργικά τύπου *NIX, Windows και Mac OS. Βασίζεται στην αρχιτεκτονική πελάτη – εξυπηρετητή, η οποία έχει ως προτερήματα τη διαχείριση καθώς και τη λειτουργία του λογισμικού να πραγματοποιείται μέσω της web διεπαφής που παρέχεται και η οποία είναι κοινή για όλα τα λειτουργικά συστήματα αφού η χρήση της υποστηρίζεται σχεδόν από κάθε γνωστό φυλλομετρητή ιστού με ενεργοποιημένη την τεχνολογία flash [9].

Το Nessus είναι ένα ισχυρό και εύκολο στη χρήση εργαλείο σάρωσης, για την ασφάλεια δικτύων. Περιέχει μια ευρεία βάση δεδομένων από plugins που ενημερώνεται καθημερινά. Σήμερα καταλέγεται μεταξύ των κορυφαίων προϊόντων αυτού του τύπου σε όλη τη βιομηχανία της ασφάλειας και έχει εγκριθεί από επαγγελματικές οργανώσεις της ασφάλειας πληροφοριών, όπως το Ινστιτούτο SANS [10]. Επιτρέπει τον έλεγχο εξ' αποστάσεως ενός δεδομένου δικτύου και μπορεί να διαπιστώσει κατά πόσο το συγκεκριμένο δίκτυο έχει παραβιαστεί με κάποιο τρόπο. Το Nessus επίσης παρέχει τη δυνατότητα τοπικού ελέγχου σε συγκεκριμένα μηχανήματα-στόχους, είτε για τρωτά σημεία που ίσως διαθέτουν, είτε για την διαπίστωση τήρησης προδιαγραφών, είτε για παραβιάσεις της πολιτικής περιεχόμενου, και πολλά άλλα.

Τα βασικά δομικά και λειτουργικά χαρακτηριστικά του Nessus περιγράφονται παρακάτω:

- **Intelligent Scanning (Έξυπνη Σάρωση):** Σε αντίθεση με πολλούς άλλους σαρωτές ασφαλείας, το Nessus δεν θεωρεί τίποτε δεδομένο. Δηλαδή, δεν θα υποθέσει ότι μια συγκεκριμένη υπηρεσία λειτουργεί πάντοτε σε μια σταθερή και συγκεκριμένη πόρτα. Αυτό σημαίνει ότι αν κάποιος web server τρέχει στην πόρτα 1234, το Nessus θα τον ανιχνεύσει και θα προχωρήσει στις κατάλληλες δοκιμές ασφάλειας. Θα επιχειρήσει να επαληθεύσει μια ευπάθεια με την εκμετάλλευση κάποιου exploit όταν αυτό είναι δυνατόν. Στις περιπτώσεις όπου ο έλεγχος με κάποιο διαθέσιμο exploit, δεν θα είναι αξιόπιστος ή ενδέχεται να επηρεάσει αρνητικά το

[9] Flash Player, http://en.wikipedia.org/wiki/Adobe_Flash
[10] SANS, http://www.sans.org/

στόχο, το Nessus δεν θα τον πραγματοποιήσει και θα βασιστεί σε πληροφορίες που θα περιέχονται σε ένα server banner για να καθορίσει την παρουσία της ευπάθεια. Σε τέτοιες περιπτώσεις, γίνεται σαφές στην τελική αναφορά το ποια μέθοδος χρησιμοποιήθηκε.

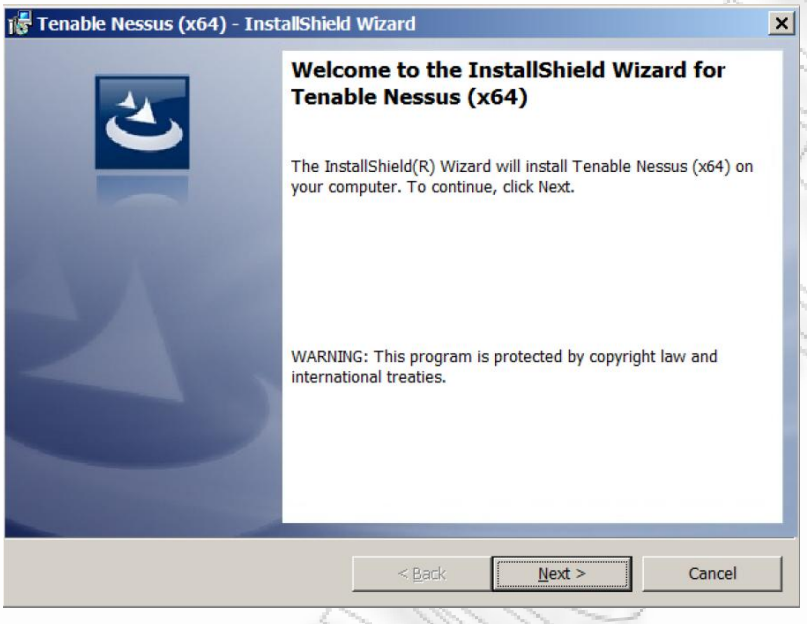
- **Modular Architecture (Η Αρχιτεκτονική):** Η αρχιτεκτονική πελάτη/εξυπηρετητή παρέχει την ευελιξία της εγκατάστασης του σαρωτή (εξυπηρετητής) σε ένα σημείο και τη σύνδεση του με τη γραφική διεπαφή (πελάτης), από οποιοδήποτε μηχάνημα που διαθέτει απλά ένα web browser, με αποτέλεσμα τη μείωση του κόστους διαχείρισης (ένας εξυπηρετητής μπορεί να προσπελαστεί από πολλούς πελάτες).
- **CVE Compatible (Συμβατότητα CVE):** Τα περισσότερα plugins παρέχουν συνδέσμους CVE τους διαχειριστές ώστε να είναι δυνατή η ανάκτηση περισσότερων πληροφοριών σχετικά με δημοσιευμένες ευπάθειες. Επίσης συχνά περιλαμβάνονται και αναφορές σε Bugtraq (BID), OSVDB, καθώς και σε ειδοποιήσεις ασφαλείας από διάφορους κατασκευαστές.
- **Plugin Architecture (Αρχιτεκτονική Plugins):** Κάθε έλεγχος ασφάλειας είναι γραμμένος ως ένα εξωτερικό plugin και όλα τα plugins ομαδοποιούνται σε 42 οικογένειες. Με αυτό τον τρόπο, μπορεί εύκολα να δημιουργηθούν προσαρμοσμένοι έλεγχοι, να επιλέγουν ειδικά plugins, ή να επιλέγει μια ολόκληρη οικογένεια, χωρίς να χρειάζεται να ξανά-συνταχθεί ο κώδικας της διεργασίας του Nessus server, **`nessusd`**.
- **NASL:** Nessus Attack Scripting Language, είναι η γλώσσα που σχεδιάστηκε για το εργαλείο Nessus και χρησιμοποιείται συγκεκριμένα για τη δημιουργία των plugins ελέγχου ασφάλειας.
- **Up-to-date Security Vulnerability Database (Ενημερωμένη Βάση Δεδομένων Ευπαθειών)**: Η ενημέρωση της βάσης δεδομένων με νέους ελέγχους ασφαλείας από την Tenable Inc. είναι καθημερινή.
- **Test Multiple Hosts Simultaneously (Έλεγχος πολλαπλών στόχων παράλληλα):** Ανάλογα με την διαμόρφωση του σαρωτή, μπορεί να γίνει έλεγχος σε ένα μεγάλο αριθμό από στόχους ταυτόχρονα.
- **Smart Service Recognition (Έξυπνη αναγνώριση υπηρεσιών):** Το Nessus δεν περιμένει οι υπηρεσίες να σέβονται απόλυτα τους κανόνες της IANA σχετικά με τις πόρτες που έχουν καθοριστεί για αυτές να "τρέχουν". Οπότε, για παράδειγμα, θα αναγνωρίσει έναν Web Server που τρέχει στην πόρτα 8080 ή στην 1234 αντί για την προκαθορισμένη 80.

- **Multiple Services (Πολλαπλές Υπηρεσίες):** Εάν υπάρχουν δύο Web Server σε έναν εξυπηρετητή που λειτουργούν παράλληλα, όπου ο ένας "ακούει" στην πόρτα 80 και ο άλλος στην πόρτα 8080 τότε το Nessus θα αναγνωρίσει και τους δύο.
- **Plugin Cooperation (Συνεργασία μεταξύ των plugin):** Τα plugins του Nessus συνεργάζονται κατά τη διάρκεια της πραγματοποίησης των ελέγχων ασφαλείας, με αποτέλεσμα, οι μη απαραίτητοι έλεγχοι να μην πραγματοποιούνται. Για παράδειγμα αν ένας FTP Server δεν παρέχει ανώνυμη αυθεντικοποίηση τότε οι έλεγχοι που αφορούν την ανώνυμη αυθεντικοποίηση δεν θα πραγματοποιηθούν.
- **Complete Reports (Πλήρης αναφορές):** Το Nessus δεν θα εμφανίσει στην τελική αναφορά μόνο την ύπαρξη κάποιας ευπάθειας μαζί με την αξιολόγηση της (*Info, Low, Medium, High και Critical*), αλλά θα προτείνει και μεθόδους για την άμβλυνση τους.
- **Full SSL Support (Πλήρης υποστήριξη SSL):** Το Nessus παρέχει τη δυνατότητα ελέγχου υπηρεσιών που προσφέρονται μέσω του πρωτοκόλλου SSL όπως για παράδειγμα τα HTTPS, SMTPS, IMAPS κα.
- **Smart Plugins (Έξυπνα plugins – προαιρετικό):** Το Nessus παρέχει την επιλογή για "βελτιστοποιημένη" χρήση των plugins κατά τη διάρκεια των ελέγχων ώστε να μπορεί να καθορίσει κατά πόσο κάποιος έλεγχος πρέπει ή όχι να εκτελεστεί. Για παράδειγμα δεν θα πραγματοποιηθεί έλεγχος για αδυναμίες του **`sendmail`** εάν ανιχνευτεί η χρήση της μεθόδου `postfix` στον εξυπηρετητή.
- **Non Destructive (Μη καταστρεπτικοί έλεγχοι - προαιρετικό):** Ορισμένοι έλεγχοι μπορεί να είναι επιζήμιοι ως προς κάποιες υπηρεσίες. Ενεργοποιώντας την επιλογή "Safe Checks" το Nessus θα βασίζεται κυρίως σε μηνύματα που συλλέγει μέσω των server banners για να καθοριστεί εάν μια ευπάθεια υπάρχει ή όχι. Δεν θα πραγματοποιήσει τους ελέγχους με τη χρήση των διαθέσιμων exploits που πιθανό να θέσουν σε κίνδυνο τις υπηρεσίες δικτύου που ελέγχονται.

2.2. Εγκατάσταση

Η τελευταία έκδοση του εργαλείου Nessus είναι διαθέσιμη για μεταφόρτωση από το σύνδεσμο http://www.nessus.org/products/nessus/nessus-download-agreement . Η έκδοση 5 του εργαλείου είναι διαθέσιμη για λειτουργικά συστήματα Windows XP, Server 2003, Server 2008, Vista και Windows 7, καθώς επίσης και για πληθώρα εκδόσεων LINUX και Mac OS. Η εγκατάσταση πρέπει να γίνει από λογαριασμό όπου διαθέτει προνόμια διαχειριστή αλλιώς ενδέχεται να προκληθούν σφάλματα του τύπου "Access Denied" κατά την εγκατάσταση.

Σημείωση: Ορισμένα αντιικά προγράμματα ενδέχεται να χαρακτηρίσουν το Nessus ως κάποιου τύπου malware ή ιό, λόγω του μεγάλου αριθμού των TCP συνδέσεων που πραγματοποιεί κατά τη διάρκεια μιας σάρωσης.

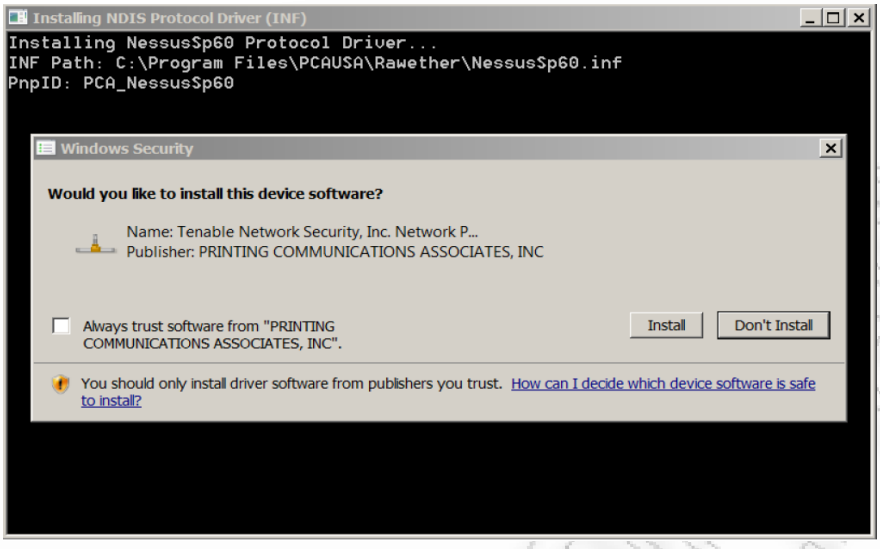


Εικόνα 4: Εγκατάσταση της εφαρμογής Nessus 1

Κατά τη διάρκεια της εγκατάστασης ο χρήστης θα πρέπει να παρέχει κάποιες βασικές πληροφορίες στο παράθυρο αλληλεπίδρασης και να αποδεχτεί την άδεια χρήσης ώστε να ξεκινήσει η εγκατάσταση

Εικόνα 5: Άδεια χρήσης του Nessus

Μετά την αποδοχή την εγκατάσταση το Nessus θα προσπαθήσει να εγκαταστήσει ένα οδηγό ο οποίος χρησιμοποιείται για την επικοινωνία του λογισμικού με την κάρτα Ethernet.



Εικόνα 6: Εγκατάσταση οδηγού επικοινωνίας κάρτας δικτύου

Μετά το τέλος της εγκατάστασης πιέστε “Finish”

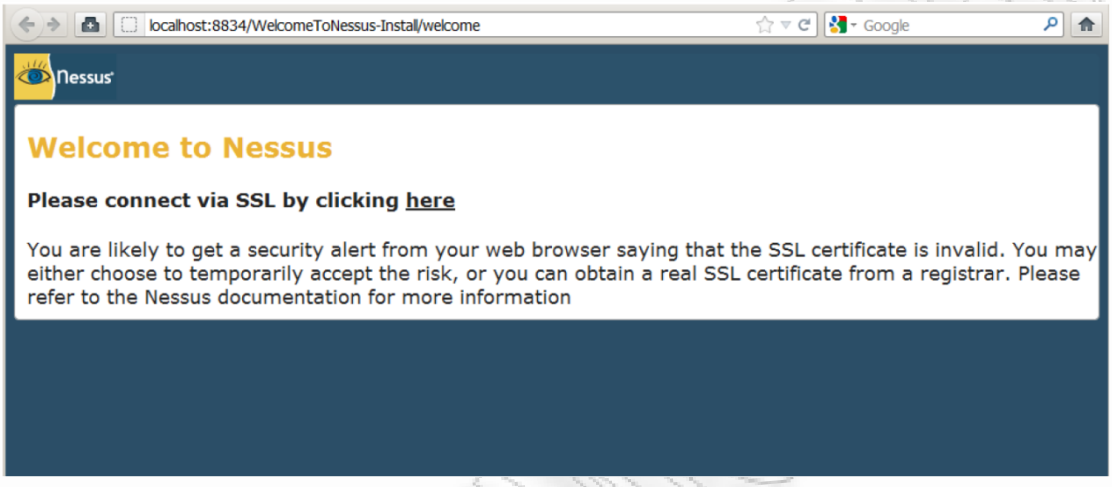
Εικόνα 7: Τέλος εγκατάστασης του Nessus

Σε αυτό το σημείο το Nessus τελειώνει την εγκατάσταση και θα φορτώσει τον προκαθορισμένο πρόγραμμα περιήγησης ιστού, ώστε να συνεχίσει η εγκατάσταση με τις αρχικές επιλογές διαμόρφωσης μέσω της web-based διεπαφής.

Όπως ήδη έχει αναφερθεί στα χαρακτηριστικά του Nessus, όλοι οι έλεγχοι εκτελούνται μέσω των plugin και η εγκατάσταση απλά του εργαλείου δεν φέρει κανένα από αυτά μαζί της. Αμέσως μετά την εγκατάσταση ορίζεται ένα παράθυρο έξι ωρών, για λόγους ασφάλειας, για την ολοκλήρωση της διαδικασίας της εγγραφής και την πραγματοποίηση

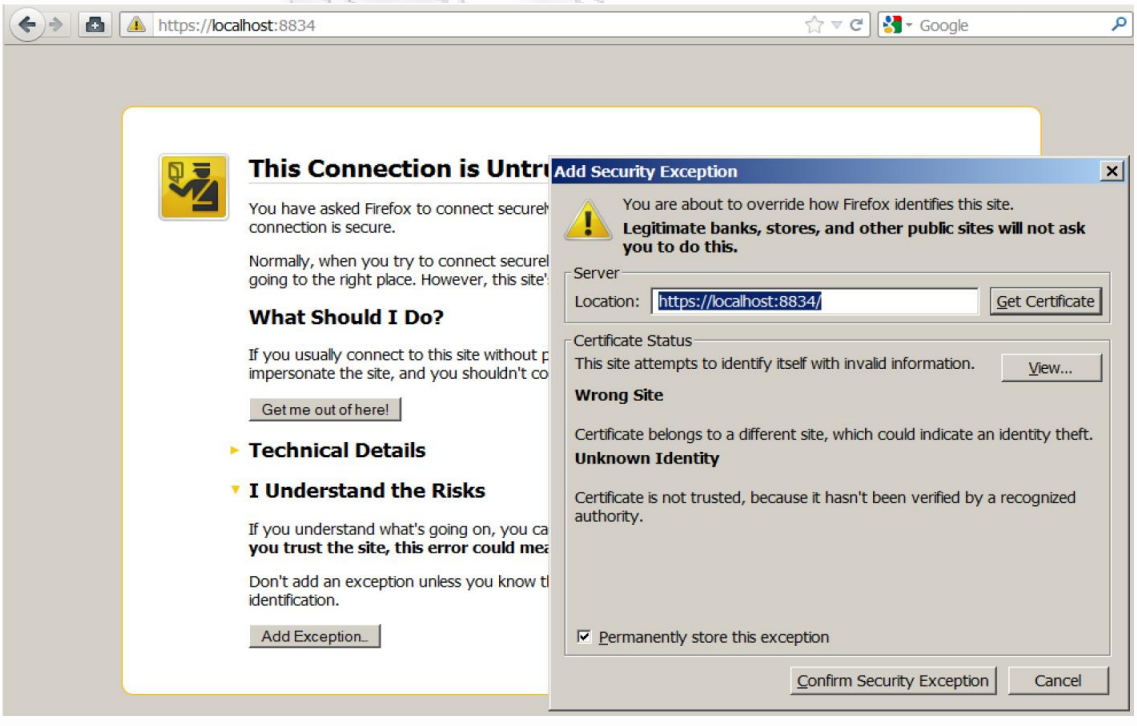
της μεταφόρτωσης των plugins. Εάν ξεπεραστεί αυτό το παράθυρο θα πρέπει να γίνει επανεκκίνηση της διεργασίας **nessusd** και της διαδικασίας εγγραφής.

Η αρχική οθόνη του γραφικού περιβάλλοντος απλά ενημερώνει το χρήστη ότι από εδώ και στο εξής όλη η κίνηση που αφορά το γραφικό περιβάλλον θα πραγματοποιείται μέσω του πρωτοκόλλου SSL.



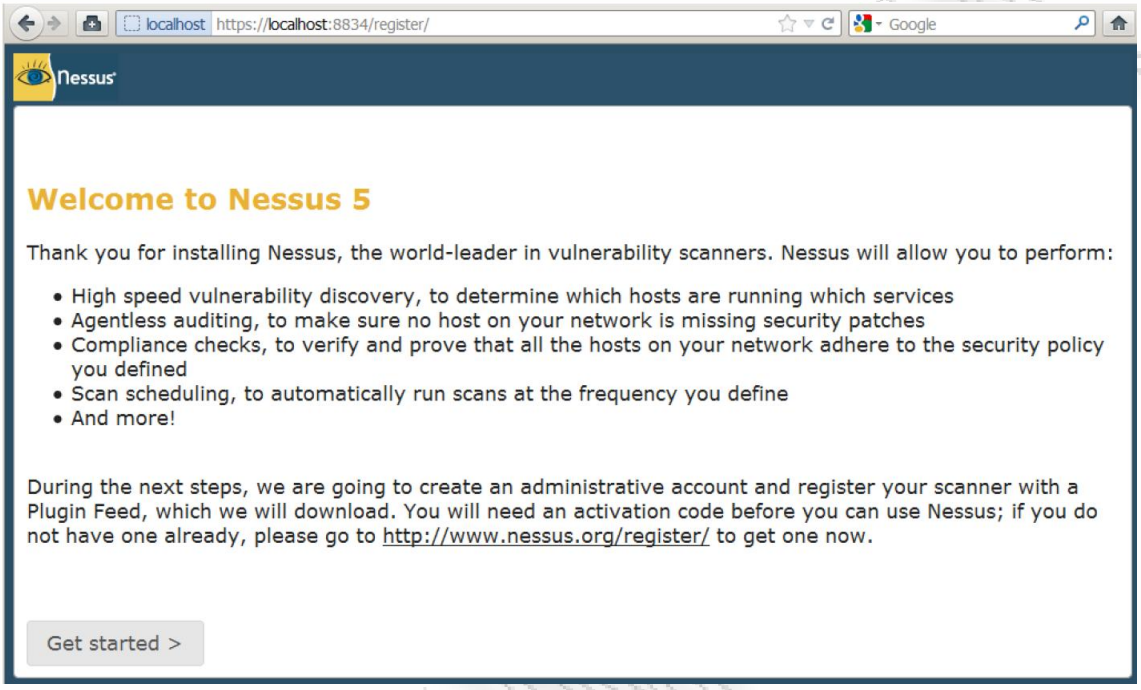
Εικόνα 8: Αρχική οθόνη προόθησης σε SSL σύνδεση

Την πρώτη φορά της σύνδεσης με τον web server του Nessus, ο web browser θα εμφανίσει κάποιο μήνυμα λάθους, το οποίο προειδοποιεί το χρήστη ότι η σύνδεση δεν είναι αξιόπιστη, λόγω της χρήσης ενός αυτό-υπογεγραμμένου πιστοποιητικού SSL.



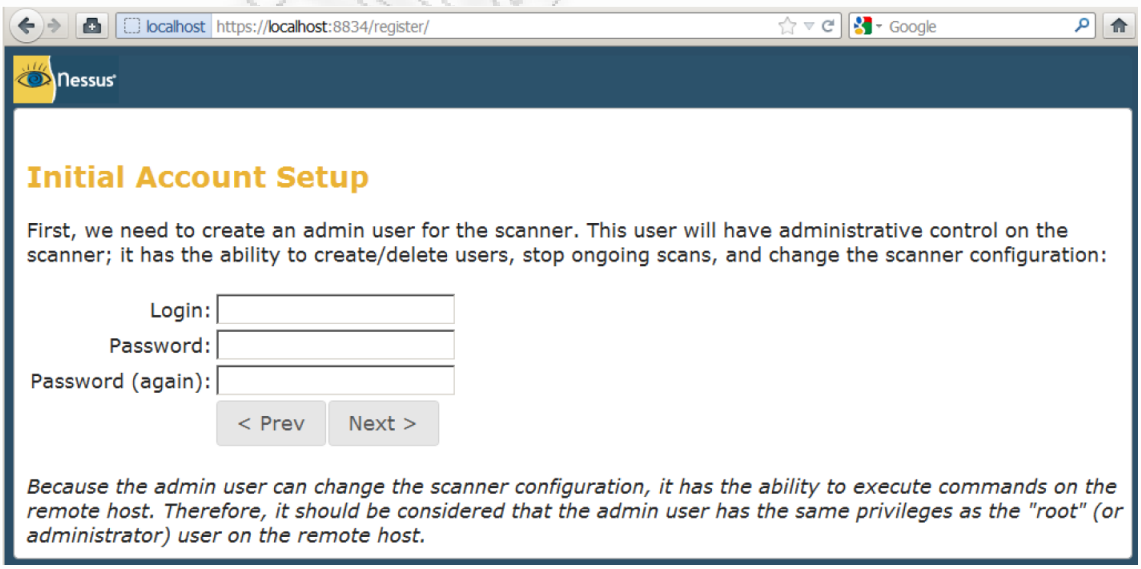
Εικόνα 9: Επιβεβαίωση χρήσης μη έμπιστου πιστοποιητικού

Εξαρτάται από τον τύπο του web browser με τον οποίο πραγματοποιείται η σύνδεση, με ποιόν τρόπο θα γίνει δεκτό το πιστοποιητικό και ο αριθμός των διαλόγων που θα γίνει η αλληλεπίδραση.



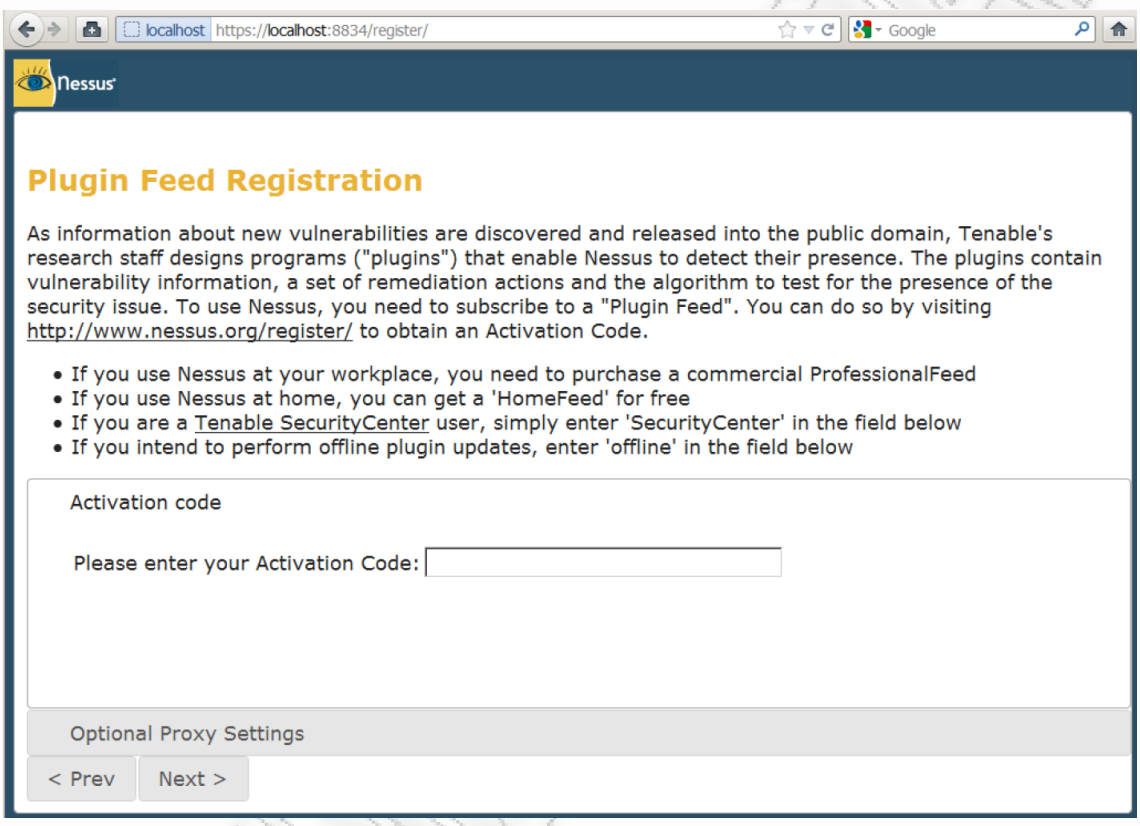
Εικόνα 10: Οθόνη έναρξης εγγραφής

Στο πρώτο βήμα της διαμόρφωσης πραγματοποιείται η δημιουργία ενός λογαριασμού. Ο αρχικός λογαριασμός χρήστη θα είναι επίσης και ο διαχειριστής του λογισμικού.



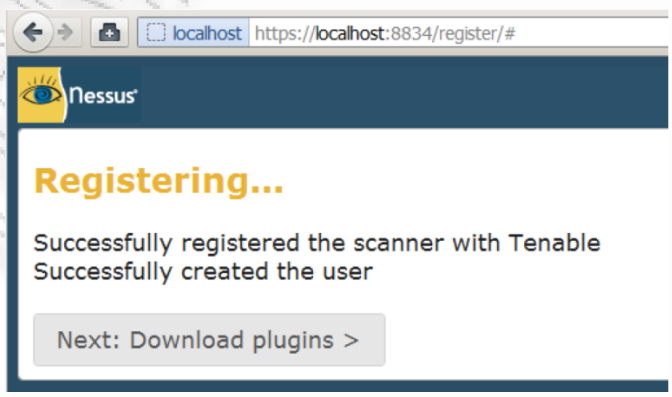
Εικόνα 11: Δημιουργία αρχικού λογαριασμού χρήστη

Στο επόμενο βήμα απαιτείτε η εισαγωγή του κωδικού ενεργοποίησης και μεταφόρτωσης των plugin. Στην προκειμένη περίπτωση έγινε χρήση του λεγόμενου ***"Home Feed"*** κωδικού ενεργοποίησης ο οποίος συνδέεται άμεσα με ένα πραγματικό λογαριασμό ηλεκτρονικού ταχυδρομείου και δεν μπορεί να χρησιμοποιηθεί σε άλλη εγκατάσταση του Nessus. Αν δεν χρησιμοποιηθεί κωδικός ενεργοποίησης τότε δεν υπάρχει η δυνατότητα εκκίνησης του λογισμικού.



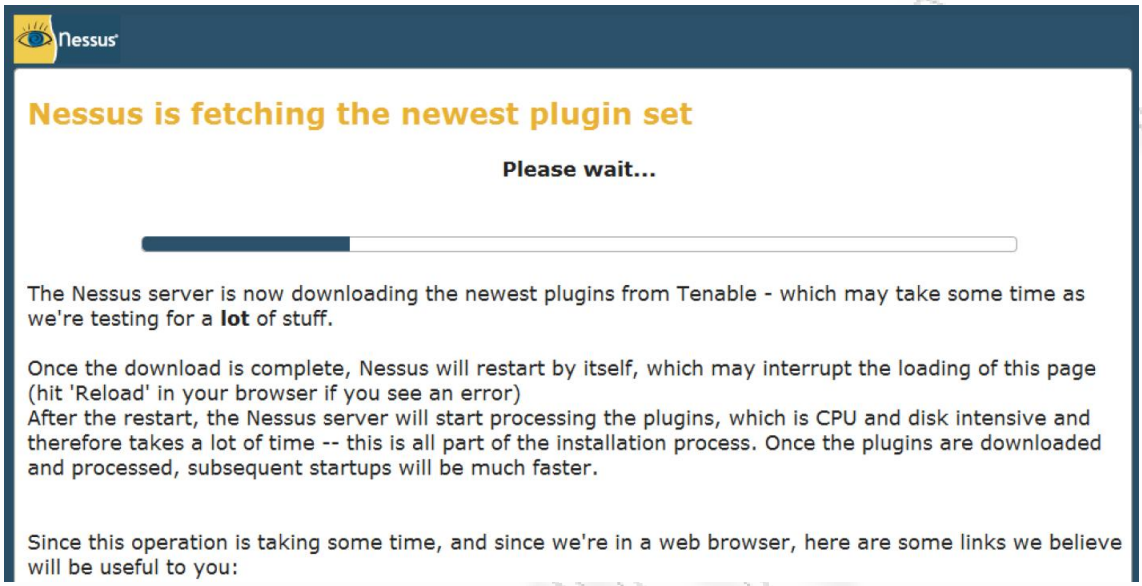
Εικόνα 12: Οθόνη εισαγωγής κωδικού ενεργποίησης

Μετά την εισαγωγή του κωδικού ενεργοποίησης πραγματοποιείται η "εγγραφή" του συστήματος.



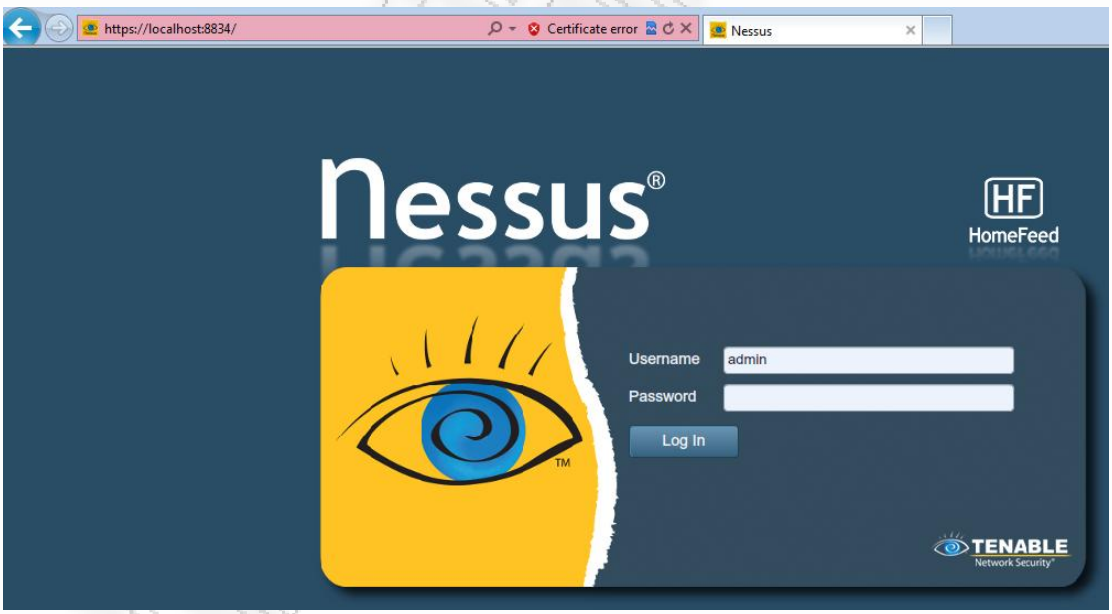
Εικόνα 13: Οθόνη μεταφόρτωσης των plugins

Στη συνέχεια το Nessus πρέπει να κάνει την μεταφόρτωση των plugins, τη σύνταξη και την εγγραφή τους στην τοπική βάση του εξυπηρετητή.



Εικόνα 14: Οθόνη προόδου μεταφόρτωσης και εγκατάσης των plugins

Τέλος γίνεται η αρχικοποίηση του λογισμικού με την ενσωμάτωση των plugins και το Nessus είναι έτοιμο προς χρήση.



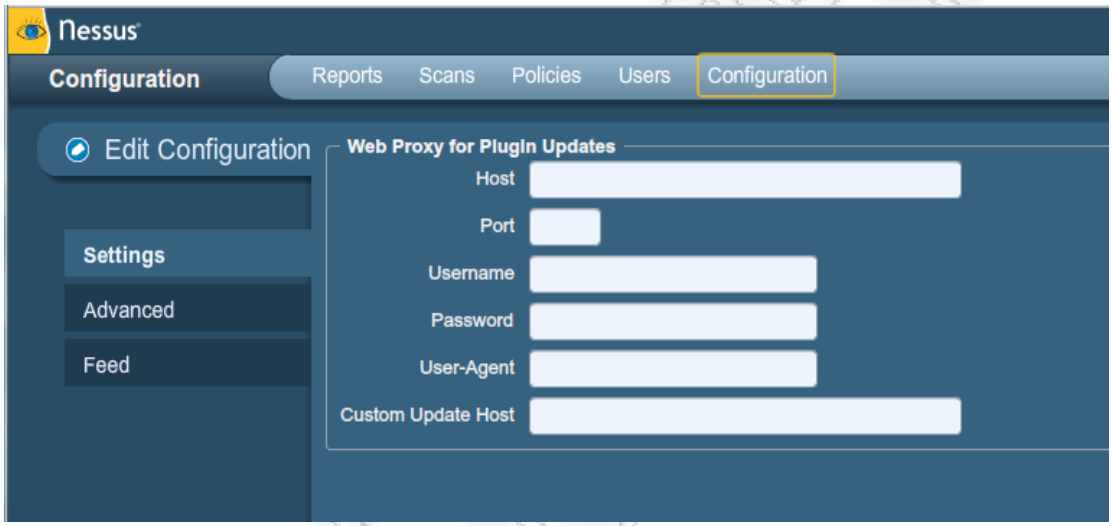
Εικόνα 15: Οθόνη εισόδου στο Nessus

2.3. Βασικές Ρυθμίσεις Περιβάλλοντος του Nessus Server

Στην έκδοση 5 του Nessus όλες οι ρυθμίσεις που αφορούν τον Nessus Server διαχειρίζονται μέσω του γραφικού περιβάλλοντος.

2.3.1. Ρυθμίσεις Web Proxy

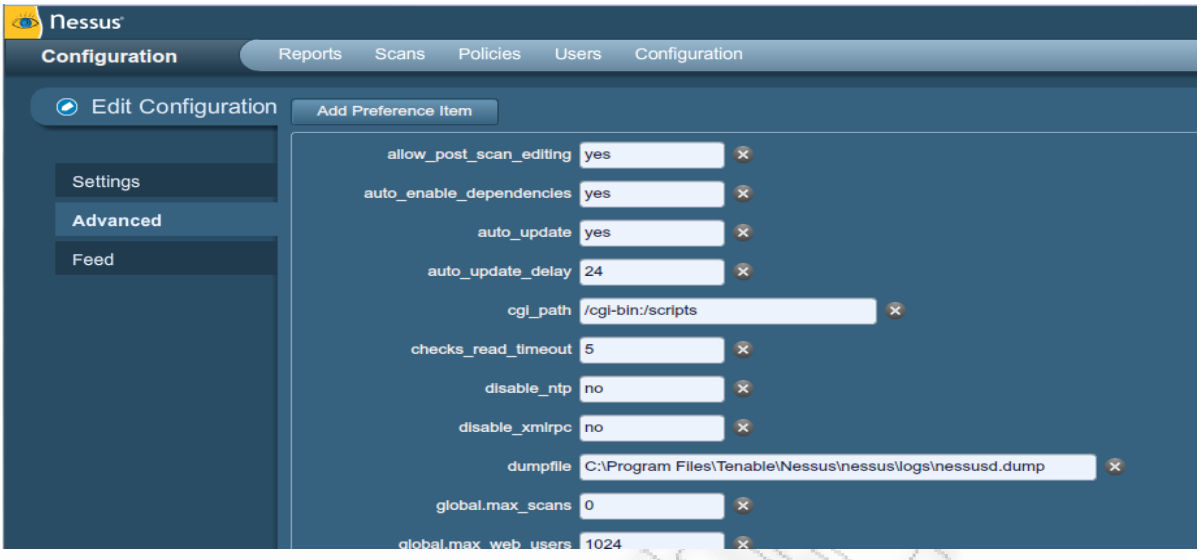
Επιλέγοντας την καρτέλα ρυθμίσεων "**Configuration"** εμφανίζεται η καρτέλα settings για τη ρύθμιση των επιλογών ενός proxy σε περίπτωση που ο Nessus server βρίσκετε σε συνθήκες περιβάλλοντος που επιβάλλουν τη κίνηση δικτύου να κατευθύνετε μέσο συγκεκριμένου μεσολαβητή.



Εικόνα 16: Οθόνη καρτέλας Configuration/Settings

2.3.2. Ρυθμίσεις Advanced

Η καρτέλα Advanced προσφέρει ένα πλήθος από επιλογές ρυθμίσεων για μεγαλύτερο έλεγχο στον τρόπο λειτουργίας του σαρωτή και απευθύνεται σε εξειδικευμένους χρήστες που γνωρίζουν την παραμετροποίηση αυτών των ρυθμίσεων. Οποιοσδήποτε χρήστης έχει προνόμια διαχειριστή μπορεί να τροποποιήσει αυτές τις ρυθμίσεις, αυτό όμως πρέπει να συμβαίνει με μεγάλη προσοχή αφού οι επιλογές αυτές επηρεάζουν τη σωστή λειτουργία του σαρωτή και επίσης κάθε αλλαγή επηρεάζει και όλους τους υπόλοιπους χρήστες. Παρόλο που αυτές οι ρυθμίσεις αφορούν την λειτουργία του σαρωτή για όλους, αρκετές από αυτές παρακάμπτονται και επαναπροσδιορίζονται μέσα από τις ρυθμίσεις που αφορούν την κάθε καινούρια πολιτική σάρωσης που δημιουργείται, μέσω της καρτέλας **policies**.



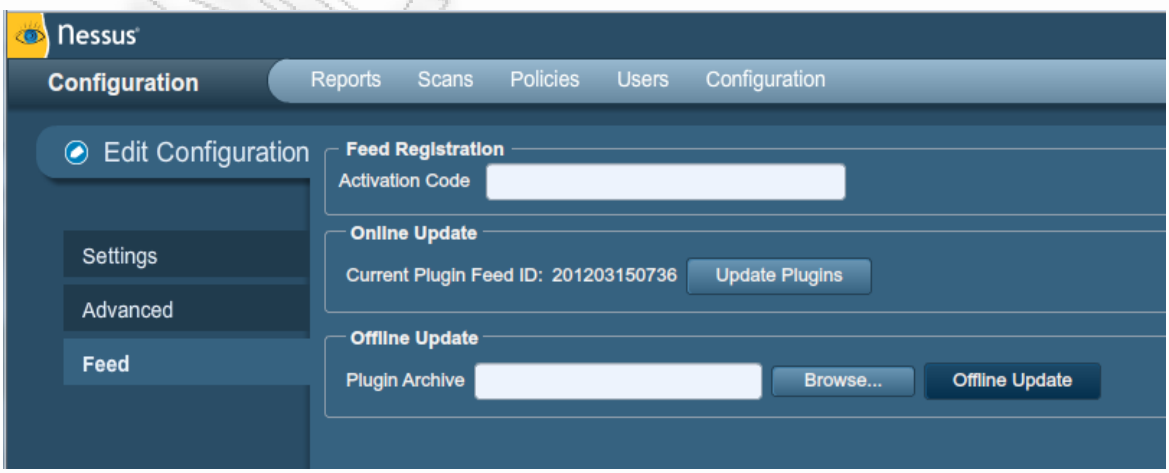
Εικόνα 17 Οθόνη καρτέλας Configuration/Advanced

2.3.3. Ρυθμίσεις Feed

Μετά τον αρχικό κωδικό ενεργοποίησης που εισάγετε κατά την εκτέλεση της φάσης αρχικοποίησης του σαρωτή, επιτρέπονται και μεταγενέστερες αλλαγές στον κωδικό ενεργοποίησης από την συγκεκριμένη καρτέλα μέσω της επιλογής "Feed Registration" στην οποία εισάγεται ο νέος κωδικός.

Η επιλογή "online Update" εμφανίζει την τρέχουσα ταυτότητα των εγκατεστημένων plugins και επιτρέπει την άμεση ενημέρωση αυτών από το κουμπί "Update Plugins". Αν για οποιοδήποτε λόγο αποτύχει η ενημέρωση το Nessus θα επαναλάβει την προσπάθεια ενημέρωσης μετά από 10 λεπτά.

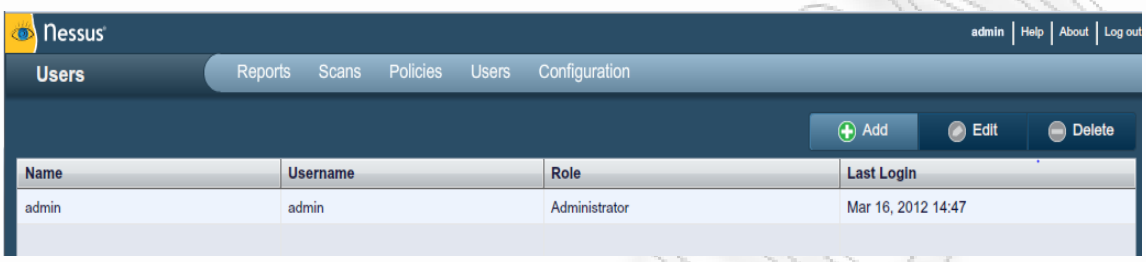
Η επιλογή "Offline Update" επιτρέπει την ενημέρωση των plugins από τοπικό αρχείο σε περίπτωση που το Nessus δεν έχει άμεση σύνδεση στο διαδίκτυο.



Εικόνα 18 Οθόνη καρτέλας Configuration/Feed

2.3.4. Ρυθμίσεις Διαχείρισης Χρηστών

Κατά τη διάρκεια της εγκατάστασης, δημιουργείται ένας χρήστης με δικαιώματα διαχειριστή. Χρησιμοποιώντας τα στοιχεία του συγκεκριμένου χρήστη γίνεται η διαχείριση και των υπόλοιπων χρηστών από την καρτέλα "**Users**" που βρίσκεται πάνω στην επικεφαλίδα.



Εικόνα 19 Οθόνη καρτέλας Users

Μέσω της καρτέλας "**Users**" μπορεί να εκτελεστεί η δημιουργία ενός νέου χρήστη μέσω της επιλογής "**Add**", η αλλαγή του κωδικού πρόσβασης ενός ήδη υπάρχοντος χρήστη μέσω της επιλογής "**Edit**" και η διαγραφή ενός χρήστη μέσω της επιλογής "**Delete**". Σημειώνεται ότι δεν υπάρχει η δυνατότητα μετονομασίας κάποιου χρήστη, παρά μόνο η διαγραφή και δημιουργία καινούριου.

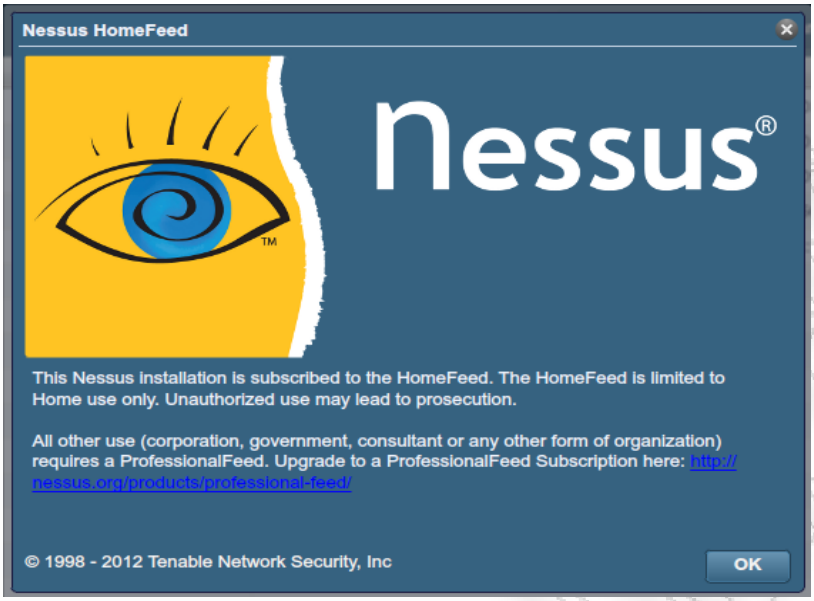
2.4. Βασικές Ρυθμίσεις Λειτουργίας του Σαρωτή Nessus

Όπως φάνηκε μέχρι τώρα το Nessus παρέχει ένα απλό αλλά ισχυρό γραφικό περιβάλλον για τη διαχείριση της δραστηριότητας ανίχνευσης ευπαθειών, με μεγαλύτερο προτέρημα του το δικτυακό γραφικό περιβάλλον (Web Based GUI), αφού με αυτό κατορθώνει να παρέχει τις ίδιες λειτουργίες μεταξύ διαφορετικών λειτουργικών συστημάτων (Windows, Linux, Mac OS X).

Η σύνδεση στο γραφικό περιβάλλον του Nessus γίνεται πολύ απλά μέσω ενός οποιουδήποτε web browser, εισάγοντας την διεύθυνση : https://localhost:8834/ Σημειώνετε ότι, η σύνδεση πρέπει να γίνεται μέσω του πρωτοκόλλου **HTTPS** καθώς το μη κρυπτογραφημένο πρωτόκολλο **HTTP**, δεν υποστηρίζεται. Επίσης, η εμπειρία χρήσης βελτιώνεται κατά πολύ όταν γίνεται χρήση του Nessus μέσω των Microsoft Internet Explorer 9 ή Mozilla Firefox 9.x ή Apple Safari 5.x ή Google Chrome 16.x.

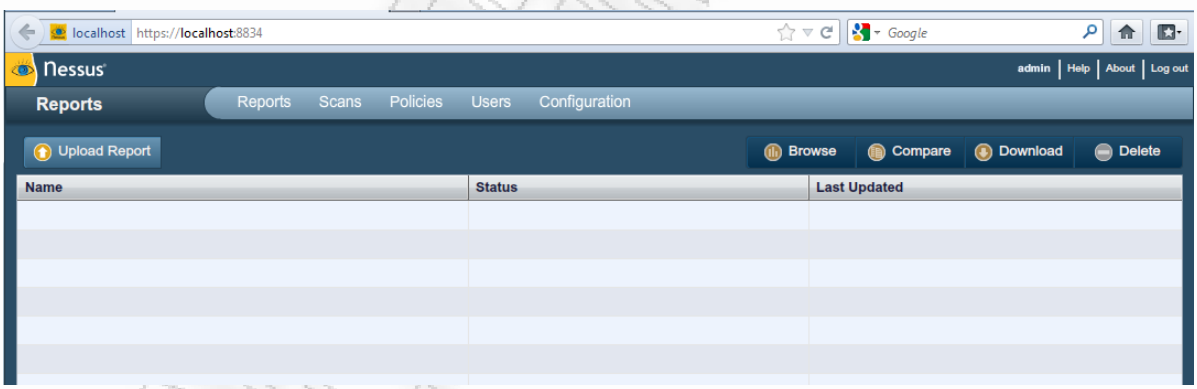
Εικόνα 20: Οθόνη εισόδου στο Nessus

Αφού γίνει η αυθεντικοποίηση χρησιμοποιώντας το λογαριασμό του διαχειριστή που φτιάχτηκε κατά την εγκατάσταση του Nessus Server, το γραφικό περιβάλλον θα εμφανίζει μια ειδοποίηση που αφορά την εγγεγραμμένη "συνδρομή" για την μεταφόρτωση των plugin,



Εικόνα 21: Προειδοποίηση χρήσης HomeFeed συνδρομής

και κατόπιν την αρχική οθόνη όπως φαίνεται παρακάτω στην οποία παρουσιάζονται όλες οι καρτέλες με τις επιλογές που αφορούν την λειτουργία του σαρωτή, ;οπως τη δημιουργία πολιτικών σάρωσης (**Policies**), τη διεξαγωγή σαρώσεων (**Scans**) και την ανασκόπηση των αναφορών (**Reports**).



Εικόνα 22: Καρτέλα Reports

2.4.1. Επισκόπηση επιλογής Policies (πολιτικές)

Μια "πολιτική" στο Nessus αποτελείται από επιλογές ρυθμίσεων που σχετίζονται με την εκτέλεση μιας σάρωσης για ευπάθειες δηλαδή, παρέχουν τη διαμόρφωση των ρυθμίσεων για τη διεξαγωγή μιας σάρωσης. Αυτές οι επιλογές περιλαμβάνουν, αλλά δεν περιορίζονται μόνο στα εξής:

- Παράμετροι που ελέγχουν τεχνικές πτυχές της σάρωσης, όπως χρονικά όρια, αριθμός των στόχων, ο τύπος του σαρωτή για τις ports και άλλα.

- Παράμετροι αυθεντικοποίησης για τοπικές σαρώσεις (πχ Windows, SSH), βάσεων δεδομένων της Oracle, HTTP, FTP, POP, IMAP ή αυθεντικοποίησης βασισμένη στο πρωτόκολλο Kerberos.
- Παράμετροι επιλογής ολόκληρων οικογενειών plugins ή συγκεκριμένων μεμονωμένων plugin για τη διεξαγωγή σάρωσης.
- Παράμετροι επιλογής πολιτικών ελέγχου συμμόρφωσης, εκτενών αναφορών, ανίχνευσης υπηρεσιών και άλλα.

Εικόνα 23: Καρτέλα Policies

2.4.1.1. Προ-εγκατεστημένες πολιτικές

Το Nessus έρχεται με διάφορες προ-εγκατεστημένες πολιτικές, γραμμένες από την ίδια την Tenable Network Security Inc. Οι συγκεκριμένες πολιτικές παρέχονται ως πρότυπα για να βοηθήσουν στη δημιουργία καινούριων, προσαρμοσμένων πολιτικών όπου θα αποτελούν τη βάση, προκειμένου να γίνουν οι αρχικές σαρώσεις στους πόρους ενός οργανισμού.

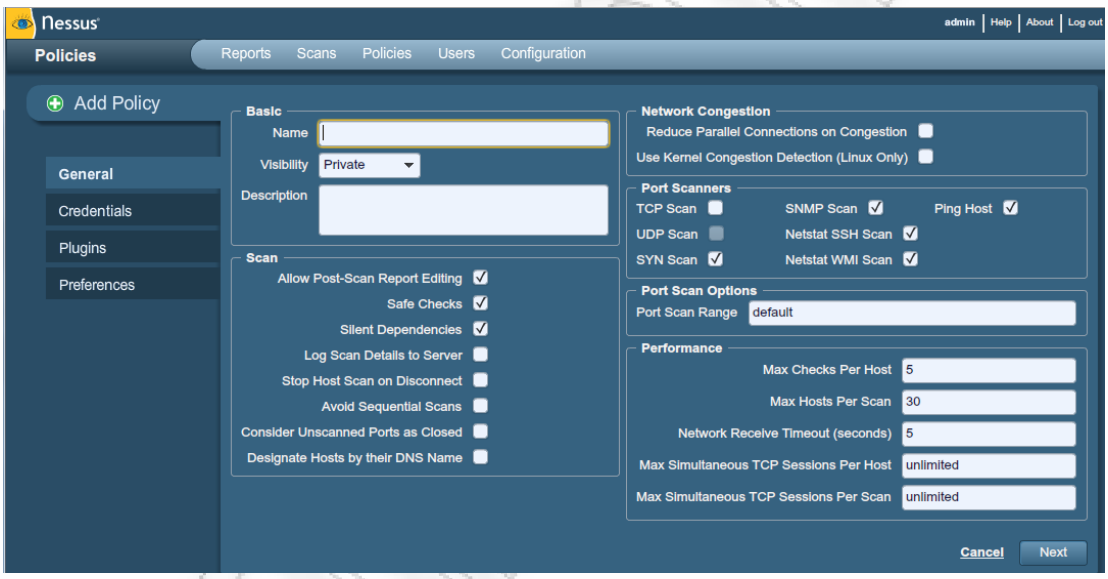
Όνομα πολιτικής	Περιγραφή
Σάρωση εξωτερικού δικτύου (External Network Scan)	Αυτή η πολιτική έχει ρυθμιστεί ώστε να σαρώνει στόχους που εκτίθενται προς το διαδίκτυο και που συνήθως προσφέρουν κάποιες περιορισμένες υπηρεσίες προς αυτό. Σε αυτή τη πολιτική είναι ενεργοποιημένα τα plugins που σχετίζονται με γνωστές αδυναμίες διαδικτυακών εφαρμογών, όπως τα CGI Abuse και CGI Abuse: XSS Family plugins. Επίσης σαρώνονται και οι 65535 πόρτες για κάθε host, συμπεριλαμβανομένου και της πόρτας 0 μέσω ξεχωριστού plugin.

Σάρωση εσωτερικού δικτύου (Internal Network Scan)	Αυτή η πολιτική έχει ρυθμιστεί έτσι ώστε να σαρώνει τους στόχους ενός εσωτερικού δικτύου (LAN). Πετυχαίνει καλύτερη απόδοση λαμβάνοντας υπόψη ότι μπορεί να χρησιμοποιηθεί, για τη σάρωση μεγάλων εσωτερικών δικτύων με πολλούς στόχους, με πολλές υπηρεσίες στον κάθε στόχο και με ενσωματωμένα συστήματα όπως είναι οι εκτυπωτές. Σε αυτή τη πολιτική δεν είναι ενεργά τα plugins που σχετίζονται με CGI Abuse καθώς επίσης δεν σαρώνονται και οι 65535 πόρτες παρά μόνο ένα σταθερό υποσύνολο από αυτές.
Προετοιμασία για ελέγχους PCI DSS (Prepare for PCS DSS audits)	Η πολιτική αυτή επιτρέπει στους ενσωματωμένους PCI DSS ελέγχους συμβατότητας προτύπων, να συγκρίνουν τα αποτελέσματα μιας σάρωσης με τα πρότυπα PCI ώστε στο τέλος να παραχθεί μια αναφορά σχετικά με το βαθμό συμμόρφωσης-συμβατότητας που υπάρχει με τα πρότυπα. Είναι πολύ σημαντικό να σημειωθεί ότι μια επιτυχημένη σάρωση αυτού του τύπου δεν εγγυάται ούτε τη συμβατότητα, ούτε μια ασφαλή υποδομή.
Έλεγχος εφαρμογών ιστού (Web App Tests)	Η συγκεκριμένη πολιτική, ανιχνεύει γνωστές καθώς και άγνωστες αδυναμίες στις εφαρμογές ιστού. Σε αυτή τη πολιτική τίθενται σε λειτουργία οι επιπλέον δυνατότητες του Nessus οι οποίες έχουν ως αποτέλεσμα την ανακάλυψη κάθε λειτουργίας και συνδέσμου (spidering-crawling) ολόκληρου του ιστότοπου όπου έχει καταδειχτεί. Στη συνέχεια γίνεται εξέταση των παραμέτρων που ανακαλύφθηκαν για αδυναμίες με τη βοήθεια απο plugins τα οποία εκτελούν XSS, SQL και command injections και πολλά άλλα. Αυτό επιτυγχάνεται μέσω μιας ενσωματωμένης λειτουργίας στο Nessus, η οποία κάνει

	χρήση της τεχνικής black box δοκιμής διείσδυσης, εισάγοντας στο λογισμικό τυχαία δεδομένα εισόδου και παρακολουθώντας την έξοδο που παράγει αυτό, για τυχόν προβλήματα ή αποτυχίας απόκρισης του λογισμικού. Η λειτουργία αυτή ονομάζεται fuzzing [11].

2.4.1.2. Δημιουργία μιας νέας πολιτικής

Μέσω της επιλογής "**Policies**" στο πάνω μέρος του γραφικού περιβάλλοντος του Nessus, εξυπηρετείται η δημιουργία καινούριων πολιτικών σάρωσης με την επιλογή "**+ Add**" στα δεξιά της οθόνης. Η οθόνη "**Add Policy**" θα εμφανιστεί ως εξής



Εικόνα 24: Οθόνη παραμετροποίησης πολιτικών - General

Σημειώνεται ότι υπάρχουν τέσσερις καρτέλες ρυθμίσεων: **General**, **Credentials**, **Plugins** και **Preferences**. Χρησιμοποιούνται για τον πιο λεπτομερή έλεγχο της λειτουργίας του Nessus και περιγράφονται παρακάτω:

Καρτέλα General

Η καρτέλα General (Γενικά) επιτρέπει την ονομασία της καινούριας πολιτικής και τη διαμόρφωση των σχετικών επιλογών σάρωσης. Υπάρχουν έξι ομαδοποιημένες επιλογές που ελέγχουν τον τρόπο συμπεριφοράς της σάρωσης.

[11] Fuzzing, https://www.owasp.org/index.php/Fuzzing

Στο πλαίσιο "**Basic**" ορίζεται το όνομα της σάρωσης, η ορατότητα της, δηλαδή η εμφάνιση αυτής σε άλλους χρήστες ή όχι καθώς και μια σύντομη περιγραφή της.

Στο πλαίσιο "**Scan**" ορίζονται περαιτέρω επιλογές που σχετίζονται με το πώς η σάρωση θα ενεργήσει όπως περιγράφονται στον πίνακα που ακολουθεί.

Επιλογή	Περιγραφή
Allow Post-Scan Report Editing	Αυτό το χαρακτηριστικό επιτρέπει στους χρήστες να διαγράψουν στοιχεία από την τελική έκθεση αφού ελεγχθούν.
Safe Checks	Με αυτή την επιλογή θα πραγματοποιηθούν ασφαλής έλεγχοι στους οποίους απενεργοποιούνται όλες οι πρόσθετες λειτουργίες που μπορούν να έχουν αρνητικές επιπτώσεις στον υπολογιστή στόχο.
Silent Dependencies	Με αυτή την επιλογή δεν περιλαμβάνεται στην τελική αναφορά η λίστα των plugins στα οποία βασίζουν τη λειτουργία τους άλλα plugins.
Log Scan Details to Server	Με αυτή την επιλογή αποθηκεύονται επιπλέον λεπτομέρειες της σάρωσης στο αρχείο καταγραφής του Nessus Server (`nessusd.messages`) συμπεριλαμβανομένου την έναρξη, τερματισμό ή τον απότομο τερματισμό κάποιου plugin. Το συγκεκριμένο αρχείο μπορεί να χρησιμοποιηθεί για να επιβεβαιωθεί ποια plugin χρησιμοποιήθηκαν και ποίοι στόχοι υπέστησαν σάρωση.
Stop Host Scan on Disconnect	Με αυτή την επιλογή το Nessus θα τερματίσει τη σάρωση προς το συγκεκριμένο στόχο εάν αυτός σταματήσει να

	ανταποκρίνεται για οποιοδήποτε λόγο. Είτε γιατί κάποιος μηχανισμός ασφαλείας αρχίσει να μπλοκάρει την κίνηση (π.χ. IDS), είτε γιατί κάποιο plugin άρνησης παροχής υπηρεσιών (DoS) εκτελέστηκε επιτυχώς κτλ.
Avoid Sequential Scans	Το Nessus έχει ως προεπιλογή όταν του δοθεί μια λίστα με IP διευθύνσεις για σάρωση, να την πραγματοποιεί διαδοχικά.. Με αυτή την επιλογή, η σάρωση της λίστας των στόχων θα πραγματοποιηθεί με τυχαία σειρά. Αυτό συνήθως είναι χρήσιμο στο να κατανεμηθεί αποδοτικότερα η κίνηση που δημιουργείται στο δίκτυο όταν αυτή απευθύνεται σε συγκεκριμένο υποδίκτυο κατά τη διάρκεια μιας μεγάλης σάρωσης.
Consider Unscanned Ports as Closed	Εάν μια πόρτα δεν έχει σαρωθεί, ακόμα και αν αυτή ήταν εκτός της προκαθορισμένης λίστας πορτών, το Nessus θα την θεωρήσει κλειστή.
Designate Hosts by their DNS Name	Με αυτή την επιλογή χρησιμοποιείτε το όνομα του στόχου αντί για τη διεύθυνση IP μέσα στην τελική αναφορά.

Στο πλαίσιο “**Network**” ορίζονται περαιτέρω επιλογές που σχετίζονται με τον καλύτερο έλεγχο της σάρωσης ως προς το δίκτυο που αυτή εκτελείται..

Επιλογή	Περιγραφή
Reduce Parallel Connections on Congestion	Με αυτή την επιλογή δίνεται η δυνατότητα στο Nessus, όταν στέλνει πολλά πακέτα δεδομένων, να ανιχνεύσει πότε το δίκτυο φτάνει στα όρια της χωρητικότητας του και όταν αυτό συμβεί να μειώσει την “κίνηση” που δημιουργεί η σάρωση ώστε να αποφευχθεί η συμφόρηση. Μόλις

	υποχωρήσει η συμφόρηση τότε το Nessus θα προσπαθήσει αυτόματα να χρησιμοποιήσει και πάλι το ελεύθερο εύρος ζώνης.
Use Kernel Congestion Detection (Linux Only)	Αυτή η επιλογή επιτρέπει στο Nessus να μετράει τις εσωτερικές διεργασίες και τη χρήση της CPU, κατά τη διάρκεια μιας σάρωσης, ώστε να αποδεσμεύει πόρους από το σύστημα όταν αυτό απαιτείται. Αυτή η δυνατότητα είναι διαθέσιμη μόνο για τους σαρωτές που έχουν εγκατασταθεί σε συστήματα Linux.

Στο πλαίσιο "**Port Scanners**" ορίζονται οι μέθοδοι σάρωσης πορτών που θα χρησιμοποιηθούν στη σάρωση.

Επιλογή	Περιγραφή
TCP Scan	Αυτή η επιλογή χρησιμοποιεί τον ενσωματωμένο σαρωτή TCP του Nessus για να αναγνωρίσει ανοικτές TCP πόρτες. Αυτός ο σαρωτής είναι βελτιστοποιημένος και διαθέτει κάποια αυτό-ρυθμιζόμενα χαρακτηριστικά. Σημείωση: Σε κάποια συστήματα (πχ. Windows και Mac OSX) ακόμη και αν επιλεγεί ο συγκεκριμένος σαρωτής δεν τίθεται σε λειτουργία αλλά αυτόματα το Nessus χρησιμοποιεί το σαρωτή SYN ώστε να αποφευχθούν σοβαρά προβλήματα απόδοσης που σχετίζονται με τα συγκεκριμένα λειτουργικά συστήματα.
UDP Scan	Αυτή η επιλογή χρησιμοποιεί τον ενσωματωμένο σαρωτή UDP του Nessus για να αναγνωρίσει ανοικτές UDP πόρτες στους απομακρυσμένους στόχους. Σημείωση: Αυτή η μέθοδος δεν θεωρείται και από τις πιο

	αξιόπιστες λόγω της φύσης του UDP πρωτοκόλλου που είναι ασύγχρονο και χωρίς τη διεξαγωγή handshake διαλόγων.
SYN Scan	Αυτή η επιλογή χρησιμοποιεί τον ενσωματωμένο σαρωτή SYN μηνυμάτων του Nessus για να αναγνωρίσει ανοιχτές TCP πόρτες στους απομακρυσμένους στόχους. Οι σαρώσεις μέσω μηνυμάτων SYN είναι δημοφιλής μέθοδοι σάρωσης ανοικτών πορτών και γενικά θεωρούνται λιγότερο “θορυβώδες” από τις TCP σαρώσεις. Ο σαρωτής στέλνει ένα πακέτο SYN στην πόρτα και περιμένει την απάντηση ή την απουσία αυτής από ένα πακέτο SYN-ACK και έτσι συμπεραίνει αν αυτή η πόρτα είναι ανοικτή ή όχι.
SNMP Scan	Η επιλογή αυτή προσανατολίζει τη σάρωση στη ανακάλυψη υπηρεσιών SNMP που ίσως διαθέτουν οι στόχοι. Αν παρέχονται από το χρήστη ρυθμίσεις για τις υπηρεσίες SNMP στην καρτέλα “Preferences” τότε το Nessus θα είναι σε θέση να παρέχει λεπτομερή αποτελέσματα ελέγχου σύμφωνα με τις απαντήσεις που θα παίρνει από τα SNMP μηνύματα, αλλιώς θα μαντέψει για σχετικές ρυθμίσεις κατά τη σάρωση..
Netstat SSH Scan	Αυτή η επιλογή χρησιμοποιεί την εντολή netstat του τοπικού μηχανήματος, που τρέχει τη σάρωση, μέσω μιας σύνδεσης SSH προς τον απομακρυσμένο στόχο, ώστε να ελέγξει για την ύπαρξη ανοικτών ports. Για τη πραγματοποίηση αυτής της σάρωσης απαιτείται η ύπαρξη συστημάτων *Unix/Linux* μέσα στους στόχους της σάρωσης και επίσης απαιτείται η πρότερη αυθεντικοποίηση σε αυτά για να τεθεί σε λειτουργία.

Netstat WMI Scan	Αυτή η επιλογή χρησιμοποιεί την εντολή netstat του τοπικού μηχανήματος, που τρέχει τη σάρωση, μέσω μιας σύνδεσης WMI προς τον απομακρυσμένο στόχο, ώστε να ελέγξει για την ύπαρξη ανοικτών ports. Για τη πραγματοποίηση αυτής της σάρωσης απαιτείται η ύπαρξη συστημάτων *Windows* μέσα στους στόχους στη σάρωση και επίσης απαιτείται η πρότερη αυθεντικοποίηση σε αυτά για να τεθεί σε λειτουργία
Ping Host	Η επιλογή αυτή πραγματοποιεί ping αιτήματα προς τους απομακρυσμένους στόχους σε διάφορες πόρτες ώστε να διαπιστωθεί αν είναι ενεργές και δίνει τη δυνατότητα καταγραφής χρόνου εκτέλεσης για τη σάρωση

Το πλαίσιο "**Port Scan Options**" κατευθύνει τη σάρωση σε ένα ορισμένο εύρος πορτών οι οποίες σαρώνονται τόσο για το TCP όσο και για το UDP πρωτόκολλο. Οι παρακάτω τιμές επιτρέπονται για χρήση.

Τιμή	Περιγραφή
"default"	Με τη χρήση της λέξης κλειδί "default" σαρώνονται περίπου 4790 κοινές πόρτες. Η λίστα με τις πόρτες αυτές βρίσκεται στο αρχείο **`nessus-services`**.
"all"	Με τη λέξη κλειδί "all" το Nessus σαρώνει και τις 65535 πόρτες.
Προσαρμοσμένη λίστα	Μια προσαρμοσμένη λίστα πορτών μπορεί να οριστεί χρησιμοποιώντας το κόμμα (,) ή/και τη παύλα (-) για ένα ορισμένο εύρος πορτών, όπως π.χ. "80,8080,443,8443" ή "1-1024,80,8080,9000-9200".

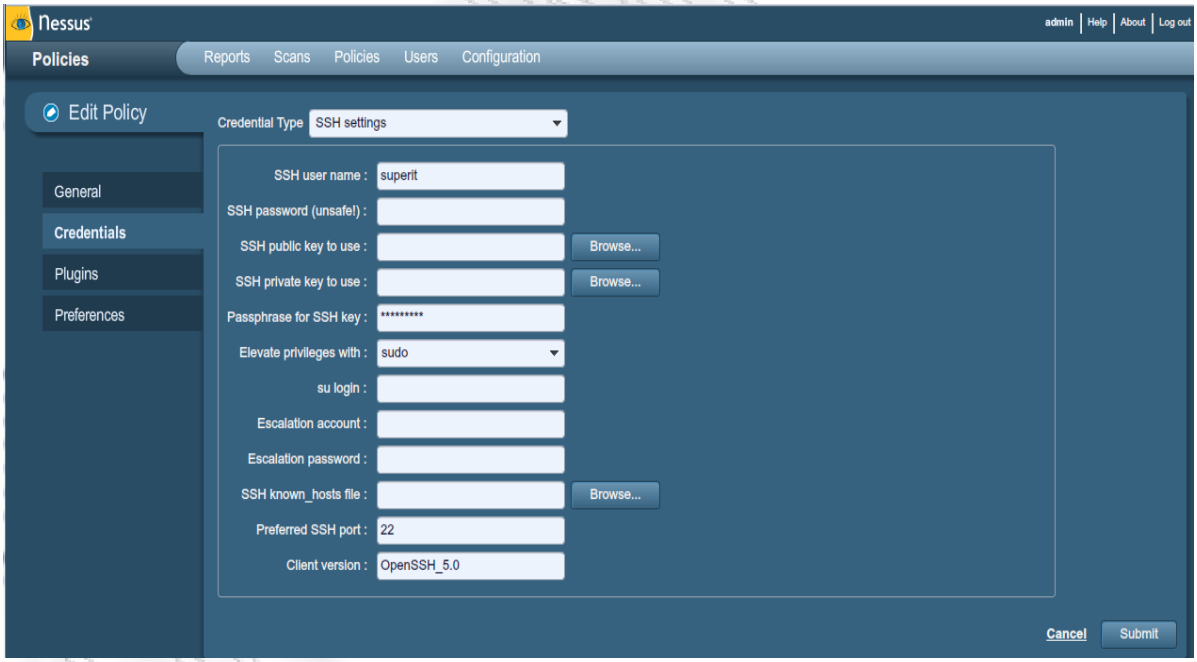
Το πλαίσιο "**Performance**" παρέχει πέντε επιλογές, δύο εξ' αυτών ελέγχουν τον αριθμό των σαρώσεων που θα γίνουν. Αυτές οι επιλογές είναι ίσως οι πιο σημαντικές κατά τη ρύθμιση μιας σάρωσης καθώς έχουν το μεγαλύτερο αντίκτυπο στο πλήθος των σαρώσεων και τη "κίνηση" του δικτύου που θα υπάρχει.

Επιλογή	Περιγραφή
Max Checks Per Host	Αυτή η ρύθμιση περιορίζει το μέγιστο αριθμό ελέγχων που θα εκτελέσει μια σάρωση προς ένα συγκεκριμένο στόχο κάθε φορά.
Max Hosts Per Scan	Αυτή η ρύθμιση περιορίζει το μέγιστο αριθμό των στόχων που το Nessus θα σαρώνει για αδυναμίες την ίδια στιγμή.
Network Receive Timeout (seconds)	Αυτό είναι το χρονικό περιθώριο που το Nessus θα περιμένει για απάντηση από ένα στόχο, εκτός και αν αυτό, ορίζεται διαφορετικά μέσα σε ένα plugin. Εάν η σάρωση πραγματοποιείται πάνω από μια αργή σύνδεση, μπορεί να χρειαστεί να οριστεί ένα μεγαλύτερο διάστημα χρόνου αναμονής.
Max Simultaneous TCP Sessions Per Host	Αυτή η ρύθμιση περιορίζει το μέγιστο αριθμό των παράλληλων συνδέσεων TCP προς ένα στόχο. Σημειώνεται ότι η συγκεκριμένη επιλογή επηρεάζεται επίσης από την επιλογή TCP Throttling.
Max Simultaneous TCP Sessions Per Scan	Αυτή η ρύθμιση περιορίζει το μέγιστο αριθμό των παράλληλων συνδέσεων TCP για το σύνολο της σάρωσης, ανεξάρτητα από τον αριθμό των στόχων που έχουν ήδη σαρωθεί. Σημαντικό: για εγκαταστάσεις του Nessus που υπάρχουν

	σε συστήματα Windows XP και Windows 7, αυτή η τιμή πρέπει να οριστεί σε 19 ή κάποιον μικρότερο αριθμό ώστε τα αποτελέσματα μιας σάρωσης να είναι ακριβή.

Καρτέλα Credentials

Η καρτέλα "**Credentials"** επιτρέπει τη ρύθμιση και εισαγωγή παραμέτρων αυθεντικοποίησης τις οποίες θα χρησιμοποιήσει το Nessus κατά τη διάρκεια της σάρωσης. Με αυτό τον τρόπο δίνετε η δυνατότητα να εκτελεστούν, μια ευρύτερη ποικιλία ελέγχων που οδηγούν σε πιο ακριβή αποτελέσματα. Αυτές οι ρυθμίσεις και οι παράμετροι αυθεντικοποίησης, πρέπει να είναι γνωστές εξαρχής στον υπεύθυνο ο οποίος θα πραγματοποιήσει τη σάρωση. Χρησιμοποιούνται αποκλειστικά για τη συλλογή πληροφοριών και αδυναμιών τις οποίες πιθανό να παρουσιάζει ο απομακρυσμένος στόχος και κυρίως αφορά τις αδυναμίες των λειτουργικών συστημάτων που έχουν εγκατεστημένα οι απομακρυσμένοι στόχοι, όπως Windows και Linux/Unix.



Εικόνα 25: Οθόνη παραμετροποίησης πολιτικών - Credentials

Η δυνατότητα χρήσης παραμέτρων αυθεντικοποίησης στο Nessus παρέχετε για διάφορα συστήματα και πρωτόκολλα τα οποία και αναφέρονται στη συνέχεια:

- Windows Credentials
- SSH Settings
- Kerberos Configuration

- Cleartext Protocol Settings
    - *Telnet*
    - *rsh*
    - *rexec*

Στην επιλογή των ***"Windows Credentials"*** υπάρχουν ρυθμίσεις που παρέχουν στο Nessus πληροφορίες όπως, ονόματα λογαριασμών του πρωτοκόλλου SMB, κωδικούς πρόσβασης, καθώς και ονόματα domain. Με τη βοήθεια αυτών των παραμέτρων δίνεται η δυνατότητα της συλλογής πληροφοριών, κατά τη διάρκεια της σάρωσης. που σχετίζονται με τον απομακρυσμένο στόχο στον οποίο, προϋποθέτεται η ύπαρξη λειτουργικού συστήματος Windows. Ένα παράδειγμα πληροφορίας, θα ήταν το κατά πόσο στο λειτουργικό σύστημα που σαρώθηκε είναι εγκατεστημένες οι τελευταίες ενημερώσεις ασφάλειας.

Αντίστοιχα στην επιλογή ***"SSH Settings"*** υπάρχουν οι κατάλληλες ρυθμίσεις που μπορούν να δώσουν στο Nessus τη δυνατότητα σάρωσης για αδυναμίες σε λειτουργικά συστήματα τύπου Linux/Unix. Ρυθμίσεις όπως τον καθορισμό των κατάλληλων κωδικών ή/και κλειδιών πρόσβασης, λογαριασμών με προνόμια διαχείρισης του συστήματος.

Με την επιλογή ***"Kerberos Configuration"*** δίνετε η δυνατότητα εισαγωγής και χρήσης κλειδιών του πρωτοκόλλου Kerberos από κάποιο απομακρυσμένο σύστημα.

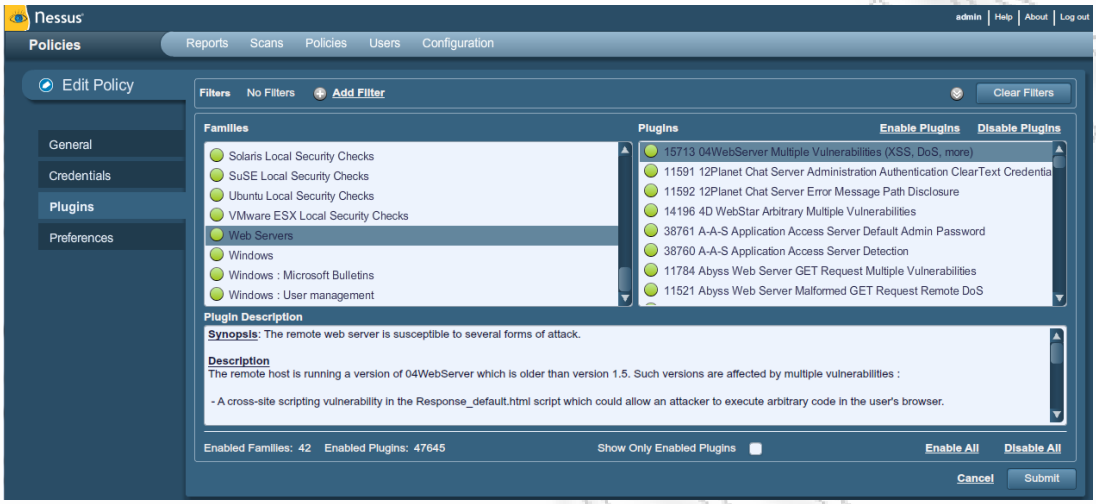
Τέλος, το Nessus δίνει τη δυνατότητα ελέγχου απομακρυσμένων συστημάτων με μη ασφαλή πρωτόκολλα, αν τα ασφαλή δεν είναι διαθέσιμα, μέσω της επιλογής ***"Cleartext Protocol Settings"***. Τα πρωτόκολλα που υποστηρίζονται με αυτή τη μέθοδο είναι τα **telnet**, **rsh** και **rexec**.

Σημειώνεται ότι η χρήση αυτής της επιλογής δεν συνιστάτε, καθώς οι παράμετροι αυθεντικοποίησης αποστέλλονται στο δίκτυο χωρίς να υποστούν καμία κρυπτογράφηση και άρα είναι εκτεθειμένες σε οποιονδήποτε βρίσκετε εκείνη τη στιγμή στο ίδιο δίκτυο με αυτό που πραγματοποιείτε η σάρωση.

Καρτέλα Plugins

Μέσω της καρτέλας "**Plugins"** δίνετε η δυνατότητα στο χρήστη να επιλέξει τους ειδικούς ελέγχους ασφάλειας που επιθυμεί, χρησιμοποιώντας μια ολόκληρη οικογένεια από Plugins, είτε ακόμη και μεμονωμένα plugins για την πραγματοποίηση ελέγχων.

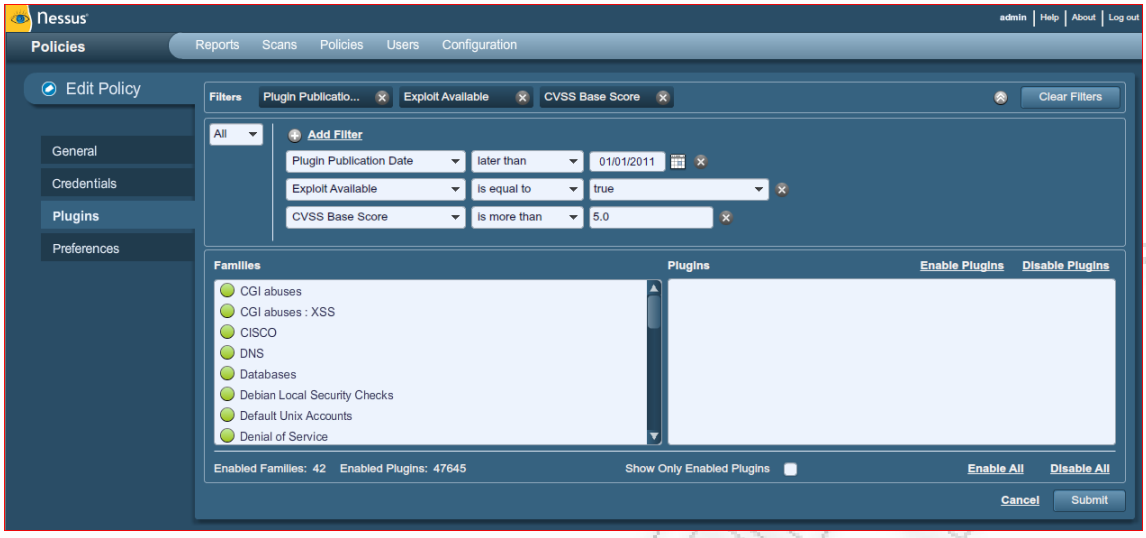
Τα plugins στην πραγματικότητα αντιπροσωπεύουν είτε κάποιον έλεγχο ασφάλειας γραμμένο στη γλώσσα NASL, είτε κάποιες συγκεκριμένες ρυθμίσεις σάρωσης. Επιπρόσθετα, μια οικογένεια από plugins αντιπροσωπεύει μια ομάδα από αυτά, με κοινά στοιχεία.



Εικόνα 26: Οθόνη παραμετροποίησης πολιτικών - Plugins

Η ενεργοποίηση και απενεργοποίηση μιας ολόκληρης οικογένειας ελέγχων γίνεται πατώντας τον πράσινο κύκλο στα αριστερά του ονόματος αυτής. Επιλέγοντας μια οικογένεια θα εμφανιστεί στο δεξιό τμήμα του παραθύρου, η λίστα με τα plugins όπου απαρτίζουν την οικογένεια αυτή. Μεμονωμένα plugins μπορεί να ενεργοποιηθούν ή να απενεργοποιηθούν ώστε να δημιουργηθούν πολύ συγκεκριμένες πολιτικές σάρωσης. Ο συνολικός αριθμός των επιλεγμένων οικογενειών και μεμονωμένων plugins, εμφανίζεται στο κάτω αριστερό μέρος της οθόνης. Αν ο κύκλος δίπλα από μια οικογένεια plugins είναι κατά ένα μέρος γκρι, αυτό σημαίνει ότι ορισμένα από τα plugins είναι απενεργοποιημένα. Επιλέγοντας κάποιο συγκεκριμένο plugin από τη λίστα στα δεξιά, εμφανίζεται στο κάτω μέρος του παράθυρου η περιγραφή αυτού όπως θα εμφανίζεται και στην τελικά αναφορά μετά από μια σάρωση. Υπάρχει ακόμα η δυνατότητα και για το “φιλτράρισμα” των plugins σε περίπτωση που ψάχνουμε κάτι συγκεκριμένο είτε με λέξεις κλειδιά είτε με το ID του plugin και άλλα.

Αν θέλουμε να δημιουργήσουμε μια πολιτική που να περιέχει plugins που μετά την εφαρμογή περισσότερων φίλτρων από ένα, επιλέγουμε τα εξής, "**All**" για την εφαρμογή όλων των επιλεγμένων φίλτρων και προσθέτουμε τα επιθυμητά φίλτρα. Για παράδειγμα, όπως φαίνεται παρακάτω, η πολιτική θα περιλαμβάνει οποιοδήποτε plugin που δημοσιεύθηκε μετά την 1η Ιανουαρίου 2011, που έχει ένα δημόσια γνωστό exploit και βαθμολογία μέσω του συστήματος CVSS υψηλότερη από 5,0.



Εικόνα 27: Οθόνη παραμετροποίησης πολιτικών – Plugins, Φίλτρα

Όταν μια πολιτική δημιουργηθεί και αποθηκευτεί για μετέπειτα χρήση, καταγράφει για χρήση όσα plugins είχαν επιλεγεί για τη δημιουργία της. Καινούρια plugins που ίσως ληφθούν κατά τη διάρκεια ενημερώσεων τους (updates) προστίθενται και ενεργοποιούνται αυτόματα εάν η οικογένεια που ανήκουν είναι πλήρως επιλεγμένη και ενεργοποιημένη. Ένα μια οικογένεια plugins είναι απενεργοποιημένη ή μερικώς επιλεγμένη, τα καινούρια plugins που αφορούν την συγκεκριμένη οικογένεια θα εισαχθούν στη συγκεκριμένη πολιτική απενεργοποιημένα.

Επίσης στο κάτω μέρος της οθόνης εμφανίζονται τρείς επιλογές οι οποίες βοηθάνε στην προβολή και επιλογή των plugins.

Επιλογή	Περιγραφή
Show Only Enabled Plugins	Με την επιλογή αυτή προβάλλονται στην οθόνη μόνο τα plugins τα οποία έχουν επιλεγεί μέχρι στιγμής είτε μέσω φίλτρων είτε μέσω απλής επιλογής του χρήστη
Enable all	Αυτή επιλογή μας προσφέρει ένα πολύ εύκολο τρόπο επιλογής όλων plugins καθώς και των οικογενειών του
Disable all	Με αυτή την επιλογή γίνεται από-επιλογή όλων των plugins και των οικογενειών τους. Αν πραγματοποιηθεί

	κάποια σάρωση χωρίς να έχει επιλεχθεί κανένα plugin για αυτή δεν θα εμφανίσει κανένα αποτέλεσμα στην τελική αναφορά.

Καρτέλα Preferences

Η καρτέλα **Preferences** προσφέρει περισσότερα και πιο λεπτομερή μέσα ως προς τον έλεγχο των ρυθμίσεις μιας σάρωσης. Κάθε στοιχείο-επιλογή από το drop-down μενού εμφανίζει περαιτέρω ρυθμίσεις διαμόρφωσης των στοιχείων που αποτελούν την επιλεγμένη κατηγορία.

Σημειώνεται ότι η καρτέλα preferences είναι μια δυναμική λίστα από επιλογές διαμόρφωσης, που εξαρτώνται άμεσα από τα plugins και τις πολιτικές ελέγχου, οι οποίες έχουν επιλεγεί για μια σάρωση. Επίσης το Nessus το οποίο προορίζεται για επαγγελματική χρήση (***Professional Feed***) παρέχει περισσότερες επιλογές ρυθμίσεων σε σύγκριση με το Nessus που προορίζεται για προσωπική χρήση ***(Home Feed)*** το οποίο και χρησιμοποιείται στην προκειμένη περίπτωση για τις ανάγκες της παρούσας μελέτης. Ο κατάλογος ενδέχεται επίσης να αλλάζει ως προς τα plugins που προστίθενται μέσω ανανεώσεων.

Επιλογή	Περιγραφή
(*) Cisco IOS Compliance Checks	Επιλογή η οποία χρησιμοποιείται σε μια πολιτική για τον καθορισμό σχετικών ελέγχων συμβατότητας προτύπων σε συσκευές Cisco IOS.
(*) Database Compliance Checks	Επιλογή η οποία επιτρέπει σε μια πολιτική τον καθορισμό σχετικών ελέγχων συμβατότητας προτύπων σε βάσεις δεδομένων όπως, DB2, SQL Server, MySQL, Oracle κ.α.
Database Settings	Επιλογή η οποία επιτρέπει σε τον καθορισμό του τύπου της βάσης δεδομένων καθώς και των παραμέτρων αυθεντικοποίησης
Do not scan fragile devices	Επιλογές που κατευθύνουν το Nessus στο να μην πραγματοποιήσει έλεγχο σε συγκεκριμένες συσκευές, λόγω αυξημένου ρίσκου κατάρρευσης του στόχου.

Global variable settings	Περιλαμβάνει μια ευρεία ποικιλία επιλογών διαμόρφωσης του Nessus.
HTTP cookies import	Επιλογή που χρησιμεύει κατά τη σάρωση δικτυακών εφαρμογών και επιτρέπει την εισαγωγή αρχείου HTTP cookie που είναι υπεύθυνο για την παροχή αυθεντικοποίησης στην εφαρμογή
HTTP login page	Επιλογή που σχετίζεται με την παροχή παραμέτρων αυθεντικοποίησης για τον έλεγχο σε εφαρμογές ιστού.
(*) IBM iSeries Compliance Checks	Επιλογή η οποία επιτρέπει σε μια πολιτική τον καθορισμό σχετικών ελέγχων συμβατότητας προτύπων σε συστήματα IBM iSeries.
IBM iSeries Credentials	Επιλογή που σχετίζεται με την παροχή παραμέτρων αυθεντικοποίησης για συστήματα IBM iSeries
(*) ICCP/COTP TSAP Addressing Weakness	Επιλογές που σχετίζονται με τις δοκιμές Εποπτικού Ελέγχου και Συλλογής Δεδομένων (Supervisory Control And Data Acquisition - SCADA)
Login configurations	Επιλογή που σχετίζεται με την παροχή παραμέτρων αυθεντικοποίησης για υπηρεσίες HTTP, NNTP, FTP, POP και IMAP
(*) Modbus/TCP Coil Access	Επιλογές που σχετίζονται με τις δοκιμές Εποπτικού Ελέγχου και Συλλογής Δεδομένων (Supervisory Control And Data Acquisition - SCADA)
Nessus SYN scanner	Επιλογές που σχετίζονται με τον ενσωματωμένο σαρωτή SYN
Nessus TCP scanner	Επιλογές που σχετίζονται με τον ενσωματωμένο σαρωτή TCP
News Server (NNTP) Information Disclosure	Μια σειρά από επιλογές για τη δοκιμή σε NNTP Servers σχετικά με σημεία αποκάλυψης πληροφοριών και άλλα τρωτά σημεία.
Oracle Settings	Επιλογές που σχετίζονται με τον έλεγχο βάσεων δεδομένων της Oracle
(*) PCI DSS compliance	Επιλογή η οποία επιτρέπει σε μια πολιτική τον καθορισμό σχετικών ελέγχων συμβατότητας προτύπων PCI DSS
Patch Management: Red Hat	Επιλογές για την ενσωμάτωση του Red Hat Satellite server

Satellite Server Settings	στο Nessus για τη διαχείριση ενημερώσεων συστημάτων Red Hat Linux.
Patch Management: SCCM Server Settings	Επιλογές για την ενσωμάτωση του System Center Configuration Manager (SCCM) στο Nessus για τη διαχείριση ενημερώσεων συστημάτων Windows.
Patch Management: VMware Go Server Settings	Επιλογές για την ενσωμάτωση του VMware Go Server στο Nessus για τη διαχείριση ενημερώσεων συστημάτων VMware
Patch Management: WSUS Server Settings	Επιλογές για την ενσωμάτωση του Windows Server Update Services (WSUS) στο Nessus για τη διαχείριση ενημερώσεων σε συστήματα Windows
Ping the remote host	Ρυθμίσεις που αφορούν τον έλεγχο της ανακάλυψης δικτυακών πόρων μέσω της διαδικασίας ping
Port scanner settings	Δύο επιλογές που προσφέρουν περισσότερο έλεγχο της διαδικασίας σάρωσης πορτών (ports)
SMB Registry : Start the Registry Service during the scan	Ρύθμιση που προτρέπει το Nessus να εκκινήσει την υπηρεσία SMB σε στόχους όπου δεν βρίσκεται σε λειτουργία εξ' αρχής
SMB Scope	Ρύθμιση που κατευθύνει το Nessus στη χρησιμοποίηση domain χρηστών αντί για τοπικούς χρήστες.
SMB Use Domain SID to Enumerate Users	Επιλογή που επιτρέπει τον καθορισμό του εύρους των SID για SMB αναζητήσεις που αφορούν domain χρήστες
SMB Use Host SID to Enumerate Local Users	Επιλογή που επιτρέπει τον καθορισμό του εύρους των SID για SMB αναζητήσεις που αφορούν τοπικούς χρήστες
SMTP Settings	Επιλογές για έλεγχο του πρωτοκόλλου SMTP
SNMP Settings	Επιλογές για έλεγχο και εισαγωγή παραμέτρων αυθεντικοποίησης του πρωτοκόλλου SNMP
Service Detection	Επιλογές που κατευθύνουν το Nessus για έλεγχο σε υπηρεσίες που χρησιμοποιούν το πρωτόκολλο SSL.
(*) Unix Compliance Checks	Επιλογή η οποία επιτρέπει σε μια πολιτική τον καθορισμό σχετικών ελέγχων συμβατότητας προτύπων σε συστήματα UNIX

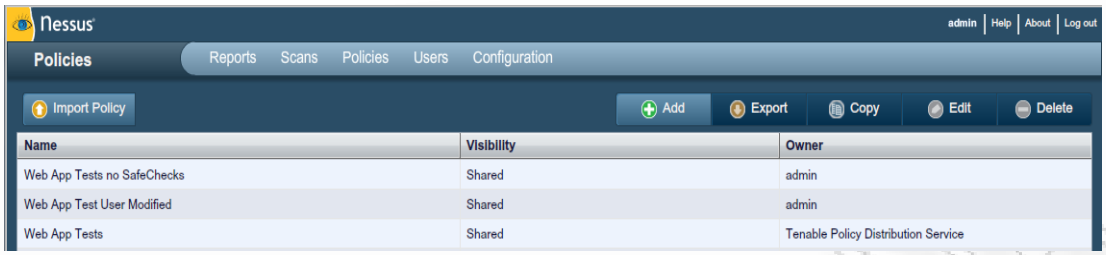
VMware SOAP API Settings	Επιλογές για έλεγχο και εισαγωγή παραμέτρων αυθεντικοποίησης για το VMware SOAP API
Wake-on-LAN	Επιλογή που κατευθύνει το Nessus στο να στείλει πακέτα Wake-On-LAN σε στόχους πριν από την εκτέλεση μιας σάρωσης.
Web Application Test Settings	Επιλογές που σχετίζονται με ελέγχους σε δικτυακές εφαρμογές.
Web mirroring	Επιλογές που σχετίζονται με τη διαμόρφωση λεπτομερειών που ελέγχουν την ποσότητα από τις σελίδες που θα αναπαραγάγει το Nessus, προκειμένου να αναλύσει το περιεχόμενο τους για τρωτά σημεία.
(*) Windows Compliance Checks	Επιλογή η οποία επιτρέπει σε μια πολιτική τον καθορισμό σχετικών ελέγχων συμβατότητας προτύπων σε συστήματα Windows
(*) Windows File Contents Compliance Checks	Επιλογή η οποία επιτρέπει σε μια πολιτική τον καθορισμό σχετικών ελέγχων συμβατότητας προτύπων σε αρχεία που αφορούν συστήματα Windows.

(*) Οι επιλογές διαμόρφωσης που είναι σημειωμένες με αστερίσκο και με μπλε χρώμα περιλαμβάνονται μόνο στο Nessus που προορίζεται για επαγγελματική χρήση (Professional Feed) και άρα δεν είναι διαθέσιμες στην παρούσα μελέτη.

Οι προηγμένες επιλογές ρυθμίσεων οι οποίες μας ενδιαφέρουν, για τους σκοπούς της συγκεκριμένη μεταπτυχιακής διπλωματικής εργασίας και οι οποίες λαμβάνουν μέρος στις σαρώσεις που θα πραγματοποιηθούν στις εφαρμογές δικτύων και που δεν περιορίζονται από την έκδοση του Nessus (Home Feed – non commercial), περιγράφονται ακολούθως στο υποκεφάλαιο **2.4.4**. Λεπτομέρειες προηγμένων επιλογών Preferences.

2.4.1.3. Εισαγωγή, εξαγωγή και αντιγραφή πολιτικών

Μέσω της καρτέλας Policies δίνεται η δυνατότητα εξαγωγής μιας πολιτικής που ήδη υπάρχει ή έχει δημιουργηθεί στο Nessus μέσω της επιλογής "**Export**" που βρίσκεται στο επάνω δεξιό μέρος της οθόνης και το οποίο πραγματοποιεί τη μεταφόρτωση ενός αρχείου **`xxxx.nessus`** το οποίο περιέχει τις ρυθμίσεις της πολιτικής.



Εικόνα 28: Καρτέλα Policies

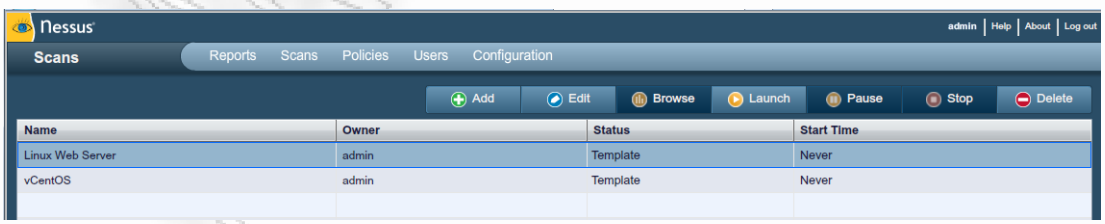
Επίσης στο επάνω αριστερό μέρος της οθόνης υπάρχει η επιλογή **"Import Policy"** μέσω της οποία μπορεί να πραγματοποιηθεί η εισαγωγή μιας πολιτικής που δημιουργήθηκε προηγούμενα και έχει ήδη πραγματοποιηθεί εξαγωγή της.

Σημειώνεται ότι οι κωδικοί πρόσβασης και τα αρχεία τύπου `.audit` που τυχών να περιέχονται σε μια πολιτική, δεν πραγματοποιείται εξαγωγή τους μέσω της επιλογής "**Export**"

Δίνεται επίσης η δυνατότητα μέσω της επιλογής αντιγραφής μιας πολιτικής, από την επιλογή "Copy", που βρίσκεται στο επάνω δεξιό μέρος της οθόνης, η δημιουργία μιας πολιτικής παρόμοιας με κάποια υφιστάμενη, η οποία όμως θα περιέχει μικρές τροποποιήσεις. Μέσω αυτής της επιλογής θα δημιουργηθεί ένα αντίγραφο της πρωτότυπης πολιτικής που μπορεί να τύχει επεξεργασίας με τις απαιτούμενες τροποποιήσεις.

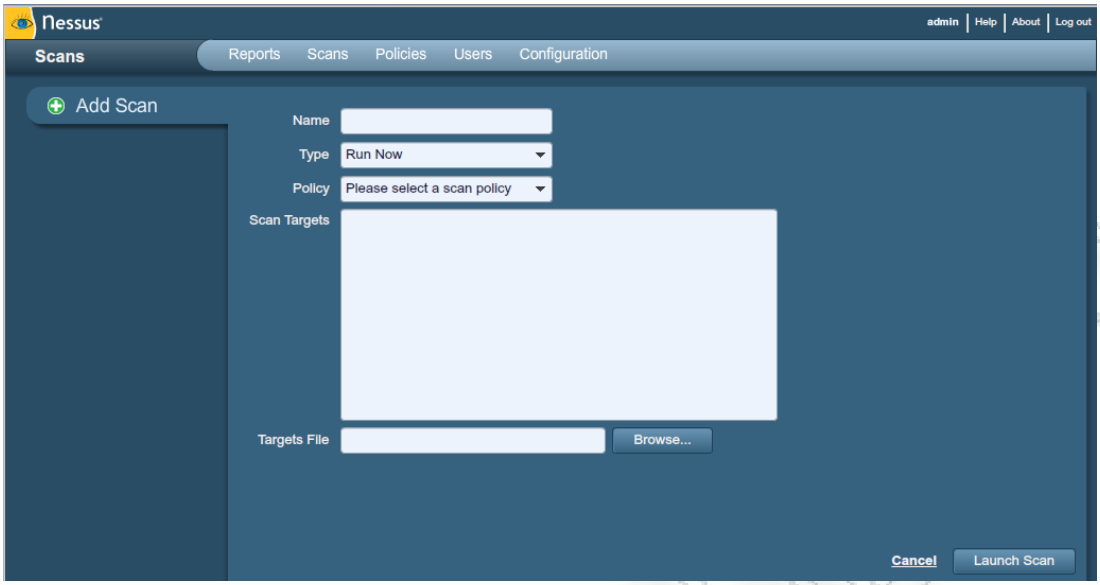
2.4.2. Επισκόπηση επιλογής Scans (Σαρώσεις)

Για να πραγματοποιηθεί μια σάρωση με το εργαλείο Nessus θα πρέπει να γίνει χρήση της επιλογής **"Scans"** στο κύριο μενού που υπάρχει στο πάνω μέρος της οθόνης. Στην ουσία, πραγματοποιείται η συσχέτιση-σύνδεση μιας λίστας από IP ή/και ονομάτων τομέα με μια συγκεκριμένη πολιτική. Με τη βοήθεια των πολιτικών που έχουν ήδη δημιουργηθεί και μέσω της επιλογής **"+ Add"** στα δεξιά της οθόνης πραγματοποιείται η προσθήκη μιας καινούριας σάρωσης.



Εικόνα 29: Καρτέλα Scans

Υπάρχουν πέντε πεδία προς συμπλήρωση για τη δημιουργία μιας καινούριας σάρωσης.



Εικόνα 30: Προσθήκη σάρωσης

Το πεδίο **"Name"** στο οποίο καταχωρείται το όνομα της καινούριας σάρωσης.

Το πεδίο **"Type"** στο οποίο προσδιορίζεται κατά πόσο μια σάρωση θα πραγματοποιηθεί άμεσα μετά τη δημιουργία της **(Run Now)** ή θα αποθηκευτεί ως πρότυπο **(Template)** για την μετέπειτα χρήση της. Η επιλογή **Schedule** και η οποία προγραμματίζει χρονικά την εκτέλεση της σάρωσης, είναι διαθέσιμη μόνο στο Nessus που προορίζεται για εμπορική χρήση (Professional Feed)[12].

Στο πεδίο **"Policy"** προσδιορίζεται η πολιτική όπου θα χρησιμοποιηθεί κατά τη διάρκεια της σάρωσης.

Στο πεδίο **"Scan Targets"** προσδιορίζονται οι απομακρυσμένοι στόχοι οι οποίοι θα σαρωθούν. Αυτό πραγματοποιείται είτε εισάγοντας την διεύθυνση IP του στόχου/στόχων προς σάρωση, είτε το όνομα τομέα (domain name) του (πχ www.ds.unipi.gr), είτε ακόμη και το εύρος από IP διευθύνσεις που αποτελούν ένα υποδίκτυο (πχ 192.168.0.0/24).

Σημειώνεται ότι ο καθορισμός του ονόματος τομέα μαζί με την διεύθυνση IP που κατέχει το συγκριμένο όνομα τομέα, δίνει τη δυνατότητα σάρωσης πολλαπλών διαφορετικών εικονικών κόμβων (Virtual Hosts)[13] που αφορούν μια συγκεκριμένη IP διεύθυνση όπως για παράδειγμα **www.example.com[192.168.1.1].** Είναι επίσης ο μοναδικός τρόπος ορισμού, που κατορθώνει το εργαλείο να παρακάμψει προβλήματα, ψευδώς αρνητικά, ανακάλυψης συνδέσμων που προκύπτουν από λάθη τύπου 404 για τη μη ύπαρξη συνδέσμου.

[12] Nessus Professional Feed , http://www.nessus.org/products/nessus-professionalfeed
[13] Virtual Hosts, http://en.wikipedia.org/wiki/Virtual_hosting

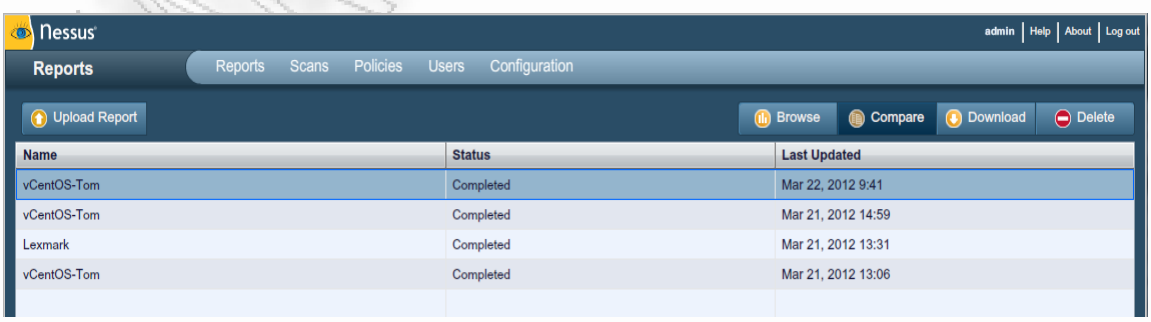
Τέλος στο πεδίο **"Target File"** μπορεί να εισαχθεί, μέσω μεταφόρτωσης, ένα αρχείο τύπου κείμενο που να περιέχει τους στόχους προς σάρωση.

Κατόπιν κάνοντας χρήση της επιλογής **"Submit"** ή **"Launch Now"** (στη περίπτωση που έχει επιλεγεί το Run Now στο πεδίο Type), τότε θα εμφανιστεί μια λίστα με όλες τις σαρώσεις που είναι ενεργές εκείνη τη στιγμή καθώς και τις σαρώσεις που έχουν αποθηκευτεί ως πρότυπα και είναι έτοιμες για να εκκινήσουν μέσω της επιλογής **"Launch".** Επίσης στις αποθηκευμένες σαρώσεις μπορεί να γίνει χρήση των επιλογών "**Edit**" για τη επεξεργασία και "**Delete**" για τη διαγραφή μιας σάρωσης. Όταν μια σάρωση είναι ενεργή τότε υπάρχει δυνατότητα χρήσης των επιλογών **"Browse"** για την εμφάνιση των μέχρι στιγμής αποτελεσμάτων, **"Pause"** για την αναστολή, και τέλος η επιλογή **"Stop"** για τον τερματισμό της σάρωσης.

2.4.3. Επισκόπηση επιλογής Reports (Αναφορές)

Στην επιλογή "**Report**" πραγματοποιείται η εμφάνιση των αποτελεσμάτων μιας συγκεκριμένης σάρωσης, όπου οι χρήστες μπορούν να περιηγηθούν, να αναγνώσουν, ακόμα και να δημιουργήσουν τις δικές τους τελικές αναφορές. Οι αναφορές αυτές χωρίζονται σε κεφάλαια προσανατολισμένα είτε στις ευπάθειες, είτε στον υπολογιστή στόχο που υπέστη σάρωση, είτε στους ελέγχους συμβατότητας προτύπων. Η εξαγωγή και μεταφόρτωση των τελικών αναφορών μπορεί να πραγματοποιηθεί σε διάφορες μορφές, όπως HTML αρχεία, αρχεία τύπου **.nessus**, ακόμα και σε μορφή **PDF** με την προϋπόθεση όμως να υπάρχει εγκατεστημένη στο σύστημα μια έκδοση **JAVA**.

Με τη χρήση των φίλτρων που παρέχονται και τα χαρακτηριστικά που προσφέρονται για εξαγωγή, ο χρήστης μπορεί να δημιουργήσει δυναμικές αναφορές προσαρμοσμένες στις δικές τους επιλογές αντί για την επιλογή απλά της χρήσης μιας υπάρχουσα αναφοράς από τη λίστα.

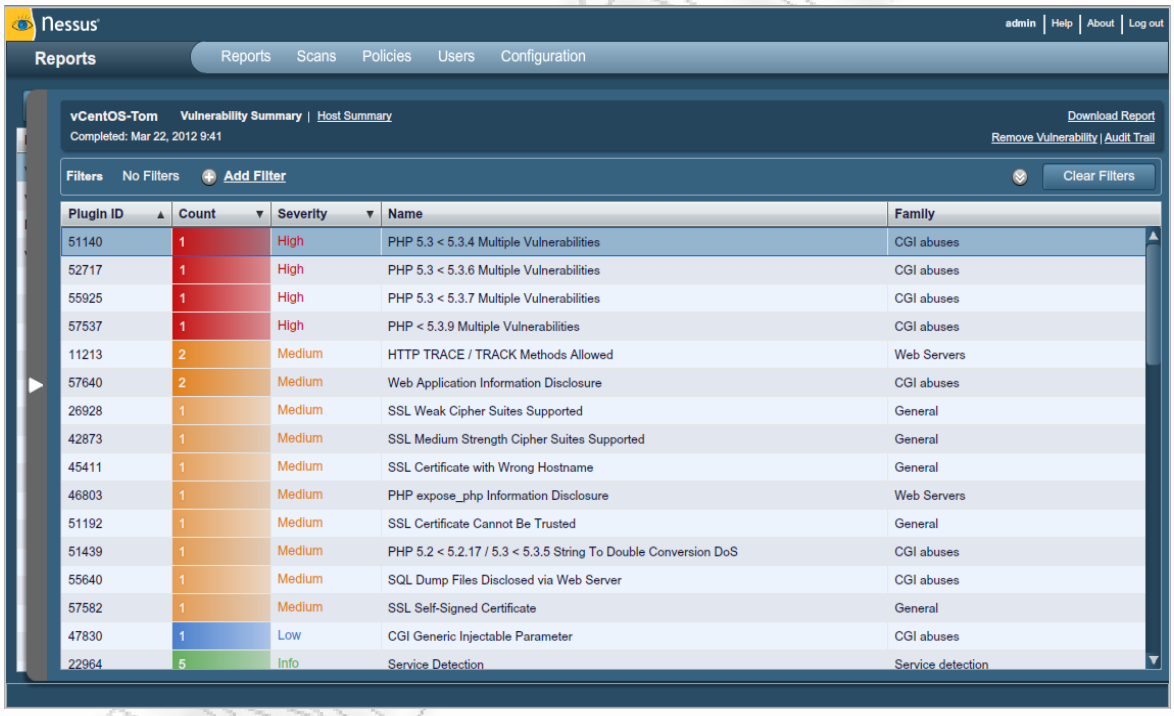


Εικόνα 31: Καρτέλα Reports

Η επιλογή "**Reports**" λειτουργεί ως ένα και μόνο σημείο αναφοράς όπου μπορεί κανείς να προβάλει, συγκρίνει και μεταφορτώσει μια αναφορά των αποτελεσμάτων σάρωσης.

Επιλογή Browse (Προβολή)

Με την επιλογή "**Browse**" πραγματοποιείται η προβολή των αποτελεσμάτων από τη συγκεκριμένη σάρωση που έχει επιλεγεί μέσα από τη λίστα. Με αυτό τον τρόπο παρουσιάζονται τα αποτελέσματα είτε με βάση, είτε τις ευπάθειες που ανακαλύφθηκαν, είτε το στόχο που αφορούν και παρουσιάζουν την εικόνα από τις πόρτες (ports) που σαρώθηκαν και τις αδυναμίες που τις αφορούν. Η προεπιλογή της προβολής γίνεται με βάση τις ευπάθειες **(Vulnerability Summary)** και παρουσιάζονται όσες ανακαλύφθηκαν, ταξινομημένες σύμφωνα με την σοβαρότητα που έχουν χαρακτηριστεί.

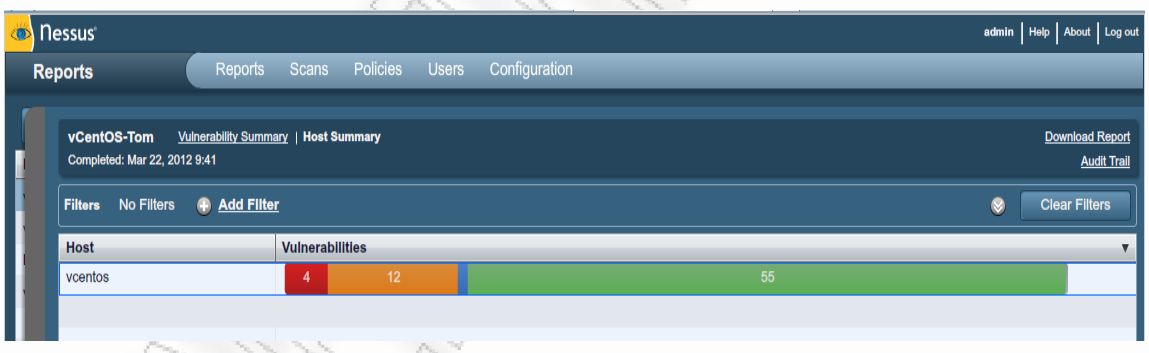


Εικόνα 32: Εμφάνιση αποτελεσμάτων σάρωσης

Από την συγκεκριμένη προβολή δίνεται η δυνατότητα της επιλεκτικής διαγραφής ευπαθειών από την αναφορά μέσω της επιλογής "**Remove Vulnerability**" στο επάνω δεξί μέρος της οθόνης. Με την επιλογή μιας γραμμής από τις ευπάθειες προβάλλονται επιπλέον πληροφορίες όπως, τους στόχους που επηρεάζονται από την συγκριμένη ευπάθεια, τις πόρτες (ports) που αφορά, καθώς επίσης και διάφορες τεχνικές λεπτομέρειες.

Εικόνα 33: Επισκόπηση αποτελέσματος από Plugin

Επιλέγοντας ως προβολή με βάση το στόχο (**Host Summary**) προβάλλονται όλες οι ευπάθειες που σχετίζονται με τον κάθε στόχο υπολογιστή που σαρώθηκε.

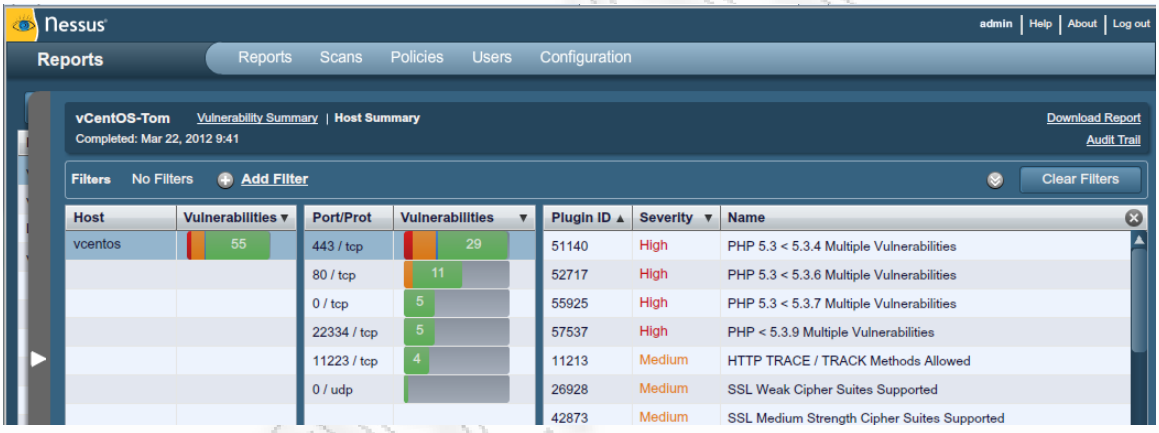


Εικόνα 34: Ταξινόμηση βάση Host name

Επιλέγοντας ένα στόχο από τη λίστα τότε εμφανίζονται λεπτομέρειες σχετικά με τις αδυναμίες που ανακαλύφθηκαν στην κάθε πόρτα (port) του συγκεκριμένου στόχου, ποιό πρωτόκολλο αφορά, το όνομα της υπηρεσίας και ένα χρωματικό κώδικα που σχετίζεται με τη σοβαρότητα της ευπάθειας.

Εικόνα 35: Αποτελέσματα που αφορούν το συγκεκριμένο Host name

Κατόπιν, με την επιλογή μιας πόρτας (port), γίνεται παρουσίαση των λεπτομερειών που σχετίζονται με τις ευπάθειες που ανακαλύφθηκαν σε αυτή.



Εικόνα 36: Λεπτομέρειες αποτελεσμάτων επιλεγμένης πόρτας

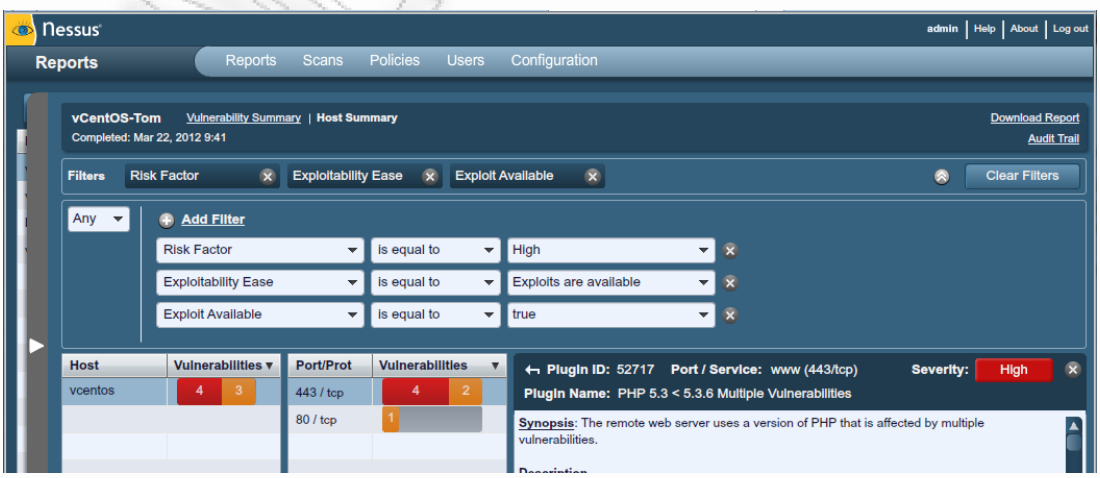
Αν στη συνέχεια γίνει επιλογή μιας ευπάθειας από τη λίστα, τότε παρουσιάζονται όλες οι σχετικές λεπτομέρειες που αφορούν τη συγκεκριμένη ευπάθεια. Σε αυτές περιλαμβάνονται, η περιγραφή της ευπάθειας, η συμβουλή για κάποια λύση αν προσφέρεται, αναφορές σε εξωτερικές πηγές, τον παράγοντα κινδύνου, τη βαθμολογία κατάταξης του CVSS, τα αποτελέσματα που παράχθηκαν από το plugin όταν αυτό εκτελέστηκε στον στόχο (αν υπάρχουν), μια σειρά από ημερομηνίες σχετικές με την ευπάθεια και το plugin και τέλος κατά πόσο υπάρχει κάποιο δημόσιο exploit για την εκμετάλλευση αυτής της ευπάθειας.

Εικόνα 37: Λεπτομέρειες αποτελεσμάτων επιλεγμένου plugin

Επιλογή Report Filters (Φίλτρα τελικής αναφοράς)

Το Nessus προσφέρει ένα ευέλικτο σύστημα φίλτρων ώστε να βοηθήσει στην εμφάνιση προσαρμοσμένων αποτελεσμάτων στην τελική αναφορά. Τα φίλτρα μπορούν να χρησιμοποιηθούν για να εμφανίσουν διαφορετικές πτυχές των ευρημάτων των plugins στο τελικό πόρισμα. Επίσης μπορεί να επεκταθεί το επίπεδο λεπτομέρειας και προσαρμοστικότητας με τη χρήση πολλαπλών φίλτρων ταυτόχρονα.

Η δημιουργία των φίλτρων μπορεί να γίνει με την επιλογή "**Add Filter**" από το πάνω αριστερό μέρος της οθόνης. Δίνεται η δυνατότητα δημιουργίας πολλαπλών φίλτρων ταυτόχρονα που επιτρέπουν έτσι τη λογική δημιουργίας σύνθετου φιλτραρίσματος για την εμφάνιση στοχευμένων αναφορών και την αποφυγή έτσι της δημιουργίας μιας χαώδους τελικής αναφοράς.



Εικόνα 38: Παρουσίαση χρήσης φίλτρων

Τα φίλτρα αναφοράς επιτρέπουν τη χρήση πολλαπλών κριτηρίων για λεπτομερή έλεγχο των αποτελεσμάτων όπως περιγράφονται παρακάτω:

Επιλογή	Περιγραφή
Plugin ID	Φιλτράρισμα αποτελεσμάτων με ένα δεδομένο αριθμό αναγνωριστικού του plugin (“ίσο”, ”διάφορο”, ”περιέχει”, ”δεν περιέχει”) π.χ., “*42111*”.
Plugin Description	Φιλτράρισμα αποτελεσμάτων με την περιγραφή κάποιου plugin, κατά πόσο περιέχει ή όχι κάποιο δεδομένο αλφαριθμητικό. π.χ., “*remote*”
Plugin Name	Φιλτράρισμα αποτελεσμάτων με ένα δεδομένο όνομα κάποιου plugin (“ίσο”, ”διάφορο”, ”περιέχει”, ”δεν περιέχει”) π.χ., “*windows*”
Plugin Family	Φιλτράρισμα αποτελεσμάτων σύμφωνα με το όνομα μιας συγκριμένης οικογένειας από plugins (“περιέχει”, ”δεν περιέχει”) (τα ονόματα είναι προκαθορισμένα)
Plugin Output	Φιλτράρισμα αποτελεσμάτων σύμφωνα με την περιγραφή κάποιου plugin εάν ”περιέχει” ή ”δεν περιέχει” κάποιο δεδομένο αλφαριθμητικό. π.χ., “*php*”.
Plugin Type	Φιλτράρισμα αποτελεσμάτων σύμφωνα με τον τύπο touplugin (τοπικό ή απομακρυσμένο)
Solution	Φιλτράρισμα αποτελεσμάτων με το κατά πόσο η δοσμένη επίλυση της ευπάθειας περιέχει ή όχι κάποιο δεδομένο αλφαριθμητικό. π.χ., “*upgrade*”.
Synopsis	Φιλτράρισμα αποτελεσμάτων με το κατά πόσο η δοσμένη σύνοψη της ευπάθειας περιέχει ή όχι κάποιο δεδομένο αλφαριθμητικό. π.χ., “*multiple*”.
Host Name	Φιλτράρισμα αποτελεσμάτων με το όνομα του στόχου κατά πόσο “περιέχει”, ”δεν περιέχει”, “είναι ίσο”, “είναι διάφορο” από κάποιο δεδομένο αλφαριθμητικό. π.χ., “*lab*” ή “*192.168*”
Port	Φιλτράρισμα αποτελεσμάτων με τον αριθμό της πόρτας κατά πόσο “περιέχει”, ”δεν περιέχει”, “είναι ίσο”, “είναι διάφορο” από κάποιο δεδομένο αριθμό. π.χ., “*80*”

Protocol	Φιλτράρισμα αποτελεσμάτων με το όνομα του πρωτοκόλλου κατά πόσο "είναι ίσο", "διάφορο" από κάποιο δεδομένο αλφαριθμητικό. π.χ., "*http*"
CPE	Φιλτράρισμα αποτελεσμάτων σύμφωνα με το Common platform Enumeration (CPE) κατά πόσο "περιέχει", "δεν περιέχει", "είναι ίσο", "είναι διάφορο" από κάποιο δεδομένο αλφαριθμητικό. π.χ., "*solaris*"
CVSS Base Score	Φιλτράρισμα αποτελεσμάτων σύμφωνα με το Common Vulnerability Scoring System (CVSS Base Score) κατά πόσο "είναι μικρότερο από", "είναι μεγαλύτερο από", "είναι ίσο", "δεν είναι ίσο", "περιέχει" ή "δεν περιέχει" ένα δεδομένο αριθμό. π.χ., "5".
CVSS Temporal Score	Φιλτράρισμα αποτελεσμάτων σύμφωνα με το Common Vulnerability Scoring System (CVSS Temporal Score) κατά πόσο "είναι μικρότερο από", "είναι μεγαλύτερο από", "είναι ίσο", "δεν είναι ίσο", "περιέχει" ή "δεν περιέχει" ένα δεδομένο αριθμό. π.χ., "3.3"
CVSS Temporal Vector	Φιλτράρισμα αποτελεσμάτων σύμφωνα με το CVSS Temporal Vector κατά πόσο "είναι μικρότερο από", "είναι μεγαλύτερο από", "είναι ίσο", "δεν είναι ίσο", "περιέχει" ή "δεν περιέχει" ένα δεδομένο αλφαριθμητικό. π.χ., "*E:F*".
CVSS Vector	Φιλτράρισμα αποτελεσμάτων σύμφωνα με το CVSS Vector κατά πόσο "είναι μικρότερο από", "είναι μεγαλύτερο από", "είναι ίσο", "δεν είναι ίσο", "περιέχει" ή "δεν περιέχει" ένα δεδομένο αλφαριθμητικό. π.χ., "*AV:N*".
Vulnerability Publication Date	Φιλτράρισμα αποτελεσμάτων σύμφωνα με την ημερομηνία πρώτης κοινής εμφάνισης της ευπάθειας, κατά πόσο είναι "πριν από", "έπειτα από", "στις", "όχι στις", "περιέχει" ή "δεν περιέχει" κάποιο δεδομένο αλφαριθμητικό π.χ., "01/01/2012".
Patch Publication Date	Φιλτράρισμα αποτελεσμάτων σύμφωνα με την ημερομηνία εμφάνισης της διόρθωσης της ευπάθειας, κατά πόσο είναι "πριν από", "έπειτα από", "στις", "όχι

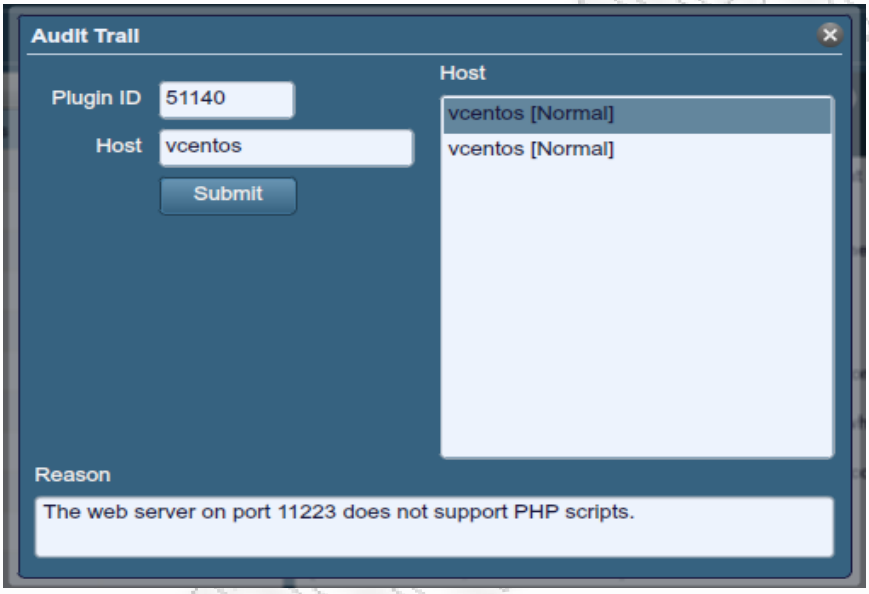
	στις", "περιέχει" ή "δεν περιέχει" κάποιο δεδομένο αλφαριθμητικό π.χ., "06/03/2011".
Plugin Publication Date	Φιλτράρισμα αποτελεσμάτων σύμφωνα με την ημερομηνία εμφάνισης του Nessus plugin,, κατά πόσο είναι "πριν από", "έπειτα από", "στις", "όχι στις", "περιέχει" ή "δεν περιέχει" κάποιο δεδομένο αλφαριθμητικό π.χ., "03/15/2010".
Plugin Modification Date	Φιλτράρισμα αποτελεσμάτων σύμφωνα με την ημερομηνία που υπέστη αλλαγές το plugin του Nessus, κατά πόσο είναι "πριν από", "έπειτα από", "στις", "όχι στις", "περιέχει" ή "δεν περιέχει" κάποιο δεδομένο αλφαριθμητικό π.χ., "03/15/2010".
CVE	Φιλτράρισμα αποτελεσμάτων σύμφωνα με τις αναφορές του CVE reference κατά πόσο "είναι ίσο", "δεν είναι ίσο", "περιέχει" ή "δεν περιέχει" ένα δεδομένο αλφαριθμητικό. π.χ., "*2011-0123*".
Bugtraq ID	Φιλτράρισμα αποτελεσμάτων σύμφωνα με το Bugtraq ID κατά πόσο "είναι ίσο", "δεν είναι ίσο", "περιέχει" ή "δεν περιέχει" ένα δεδομένο αλφαριθμητικό. π.χ., "*51300*".
CERT Advisory ID	Φιλτράρισμα αποτελεσμάτων σύμφωνα με το CERT Advisory ID κατά πόσο "είναι ίσο", "δεν είναι ίσο", "περιέχει" ή "δεν περιέχει" ένα δεδομένο αλφαριθμητικό. π.χ., "TA12-010A".
OSVDB ID	Φιλτράρισμα αποτελεσμάτων σύμφωνα με το Open Source Vulnerability Database κατά πόσο "είναι ίσο", "δεν είναι ίσο", "περιέχει" ή "δεν περιέχει" ένα δεδομένο αλφαριθμητικό. π.χ., "78300".
Secunia ID	Φιλτράρισμα αποτελεσμάτων σύμφωνα με το Secunia ID κατά πόσο "είναι ίσο", "δεν είναι ίσο", "περιέχει" ή "δεν περιέχει" ένα δεδομένο αλφαριθμητικό. π.χ., "47650".
Exploit Database ID	Φιλτράρισμα αποτελεσμάτων σύμφωνα με το Exploit Database ID κατά πόσο "είναι ίσο", "δεν είναι ίσο", "περιέχει" ή "δεν περιέχει" ένα δεδομένο αλφαριθμητικό. π.χ., "18380".

Metasploit Name	Φιλτράρισμα αποτελεσμάτων σύμφωνα με το Metasploit Name κατά πόσο “είναι ίσο”, “δεν είναι ίσο”, “περιέχει” ή “δεν περιέχει” ένα δεδομένο αλφαριθμητικό. π.χ., “xslt_password_reset”.
Exploit Hub	Φιλτράρισμα αποτελεσμάτων σύμφωνα με το κατά πόσο ένα exploit hub είναι ή όχι αληθές.
IAVA	Φιλτράρισμα αποτελεσμάτων σύμφωνα με τις αναφορές IAVA reference κατά πόσο “είναι ίσο”, “δεν είναι ίσο”, “περιέχει” ή “δεν περιέχει” ένα δεδομένο αλφαριθμητικό. π.χ., “2012-A0008”.
See Also	Φιλτράρισμα αποτελεσμάτων σύμφωνα με το πεδίο “see also” που περιέχεται στις πληροφορίες κάποιου plugin στο κατά πόσο “είναι ίσο”, “δεν είναι ίσο”, “περιέχει” ή “δεν περιέχει” ένα δεδομένο αλφαριθμητικό. π.χ., “seclists.org”
Exploits Available	Φιλτράρισμα αποτελεσμάτων σύμφωνα με την ύπαρξη ή όχι κάποιου κοινά γνωστού exploit.
Exploitability Ease	Φιλτράρισμα αποτελεσμάτων σύμφωνα με το κατά πόσο ένα exploit “είναι” ή “δεν είναι” ένας από τους τρείς τύπους: “Exploits are available”, “No exploit is required” ή “No known exploits are available”.
Metasploit Exploit Framework	Φιλτράρισμα αποτελεσμάτων σύμφωνα με την ύπαρξη ή όχι μια ευπάθειας μέσα στο Metasploit Exploit Framework.

Σημειώνεται ότι στη λίστα με τα διαθέσιμα φίλτρα εμφανίζονται μόνο όσα φίλτρα σχετίζονται με τις αδυναμίες που έχουν ανακαλυφθεί, με αποτέλεσμα να μην είναι όλα διαθέσιμα σε κάθε αναφορά. Επίσης με τη βοήθεια των λέξεων κλειδιών “**Any**” και “**All**” διαφοροποιείται η εφαρμογή των πολλαπλών φίλτρων. Με την λέξη “**All**” μόνο τα αποτελέσματα που ικανοποιούν όλα τα φίλτρα θα εμφανιστούν σε αντίθεση με την λέξη “**Any**” που θα εμφανίσει τα αποτελέσματα από οποιοδήποτε φίλτρο φέρει κάποιο αποτέλεσμα.

Αφού γίνει το φιλτράρισμα των αποτελεσμάτων και αυτά εμφανιστούν στην αναφορά όπως επιθυμεί ο χρήστης δίνεται η δυνατότητα μεταφόρτωσης της συγκεκριμένης αναφοράς με την επιλογή “**Download Report**” στο πάνω δεξί μέρος της οθόνης.

Το Nessus στην τελική αναφορά παράγει μια συνοπτική λίστα με τα plugins που χρησιμοποιήθηκαν και τα προβλήματα που εντοπίστηκαν μέσω αυτών. Πολλές φορές όμως χρειάζεται να γίνεται γνωστός ο λόγος που κάποιο plugin δεν εμφάνισε αποτελέσματα σε κάποια συγκεκριμένη πόρτα επάνω σε κάποιον συγκεκριμένο στόχο. Σε αυτό μπορεί να βοηθήσει η επιλογή "**Audit Trail**" μέσω της οποίας εμφανίζονται λεπτομέρειες σχετικά με το λόγο, μη εκτέλεσης ή μη εμφάνισης αποτελεσμάτων από κάποιο plugin, στην τελική αναφορά, σε συγκεκριμένο στόχο και σε συγκεκριμένη πόρτα αυτού.



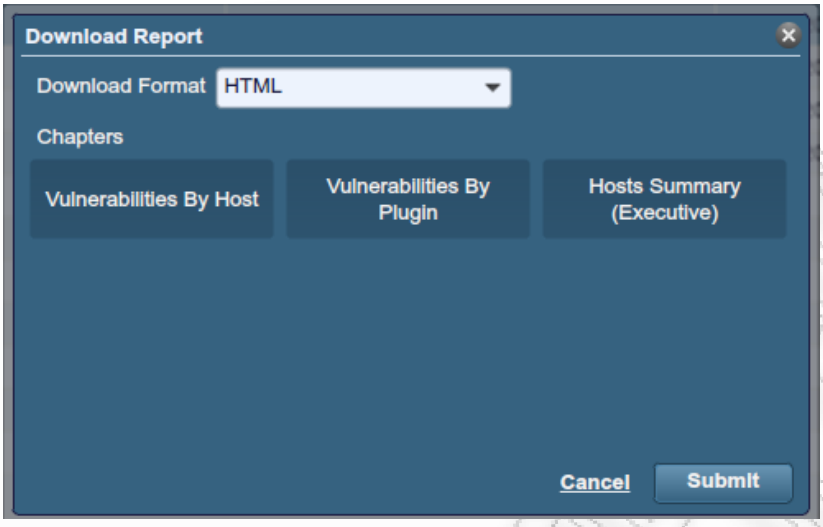
Εικόνα 39: Προβολή Audit trail

Επιλογές Upload και Download (Αποστολή και Λήψη)

Τα αποτελέσματα από τις σαρώσεις που έχουν πραγματοποιηθεί μπορούν να μεταφορτωθούν από ένα σαρωτή Nessus με σκοπό να εισαχθούν ή μελετηθούν σε κάποιο άλλο διαφορετικό σαρωτή.

Μέσω των επιλογών Upload (Αποστολή) και Download (Λήψη) επιτυγχάνεται καλύτερη διαχείριση των αποτελεσμάτων των σαρώσεων και πραγματοποιείται η μεταφόρτωση των αρχείων ακόμα και για λόγους δημιουργίας αντιγράφων ασφαλείας των τελικών αναφορών.

Για να πραγματοποιηθεί η εξαγωγή και μεταφόρτωση μιας αναφοράς, γίνεται επιλογή αυτής μέσα από τη λίστα των διαθέσιμων αναφορών και με τη βοήθεια της επιλογής Download εμφανίζεται ένας διάλογος με τις επιλογές μεταφόρτωσης και δημιουργίας της όπως παρουσιάζεται παρακάτω.



Εικόνα 40: Προβολή επιλογών μεταφόρτωσης αποτελεσμάτων

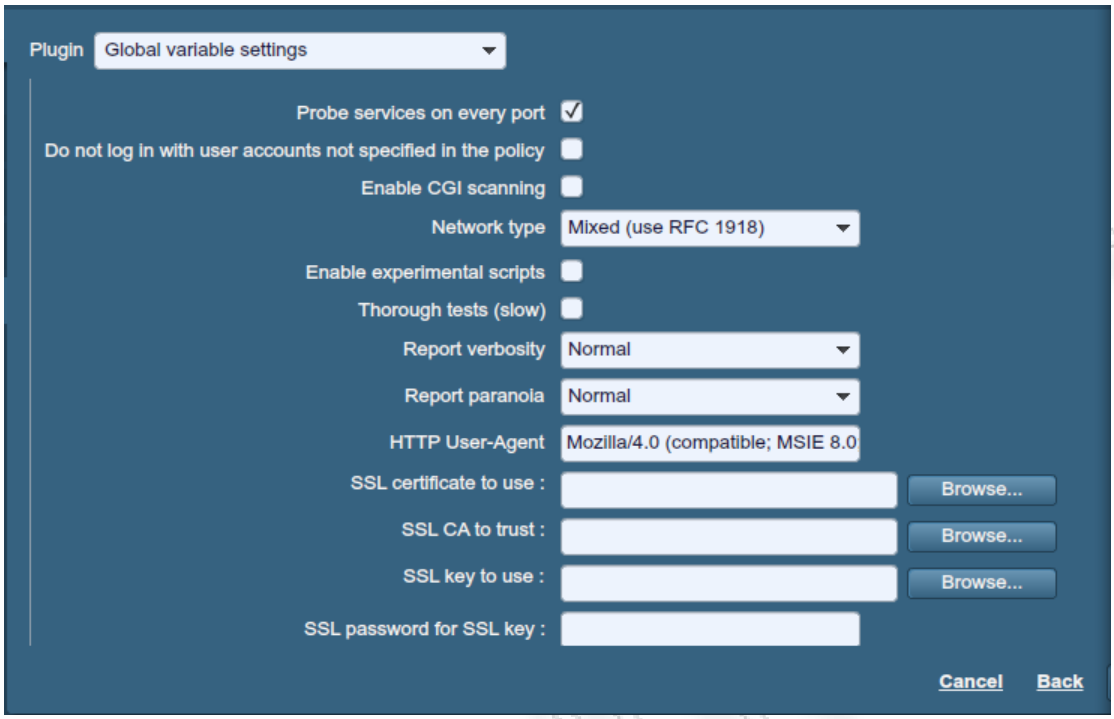
Η μεταφόρτωση μπορεί να πραγματοποιηθεί σε διαφόρων τύπων αρχεία τα οποία είναι τα εξής:

- Αρχεία τύπου **`.nessus`** , τα οποία είναι βασισμένα σε XML μορφοποίηση
- Αρχεία τύπου **`.nessus(v1)`** , τα οποία είναι επίσης βασισμένα σε XML μορφοποίηση αλλά για χρήση με παλαιότερες εκδόσεις του σαρωτή Nessus.
- Αρχεία τύπου **`HMTL`** , όπου είναι βασισμένα στην HMTL και δίνεται η δυνατότητα προβολής τους σε κάποιον Web Browser.
- Αρχεία τύπου **`PDF`**, στα οποία για να πραγματοποιηθεί η εξαγωγή και παραγωγή τέτοιου τύπου αρχεία, πρέπει να υπάρχει εγκατεστημένη κάποια έκδοση JAVA στον υπολογιστή που φιλοξενεί το Nessus.
- Αρχεία τύπου **`NBE`** , στα οποία όλα τα πεδία είναι χωρισμένα με το χαρακτήρα «|» και χρησιμοποιείται συνήθως για την εισαγωγή των αναφορών σε τρίτα λογισμικά.

2.4.4. Λεπτομέρειες προηγμένων επιλογών Preferences

Ρυθμίσεις Global Variable

Οι ρυθμίσεις Global Variable settings περιέχουν μια γκάμα από επιλογές για τον Nessus server.



Εικόνα 41: Επιλογές Policy/Preferences/Global Variable

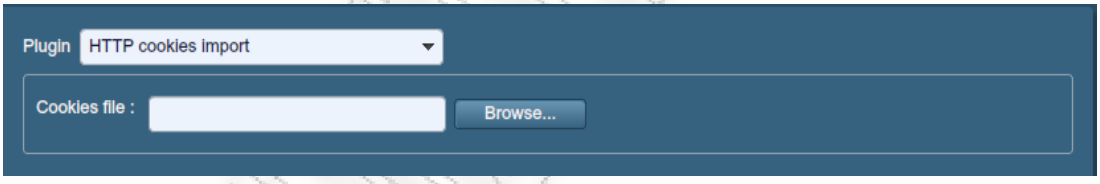
Επιλογή	Περιγραφή
Probe services on every port	Με αυτή την επιλογή επιχειρείται η αντιστοίχιση κάθε ανοιχτής πόρτας με την υπηρεσία που εκτελείται σε αυτή. Σημειώνεται ότι σε μερικές σπάνιες περιπτώσεις η συγκεκριμένη επιλογή ενδέχεται να διακόψει τη λειτουργία κάποιων υπηρεσιών και να προκαλέσει απρόβλεπτα και ανεπιθύμητα αποτελέσματα.
Do not log in with user accounts not specified in the policy	Η συγκεκριμένη επιλογή χρησιμοποιείται για την αποφυγή τυχόν κλειδώματος σε λογαριασμούς χρηστών, στην περίπτωση που υπάρχει πολιτική κλειδώματος για λόγους ασφάλειας της εφαρμογής ή του συστήματος το οποίο σαρώνεται.
Enable CGI scanning	Με αυτή την επιλογή ενεργοποιούνται οι CGI έλεγχοι οι οποίοι είναι απαραίτητοι για τον έλεγχο διαδικτυακών εφαρμογών.
Network type	Με αυτή την επιλογή επιτρέπεται ο καθορισμός στον τύπο του δικτύου ανάλογα με τον τύπο των IP διευθύνσεις που χρησιμοποιούνται. Δηλαδή εάν υπάρχουν ιδιωτικές διευθύνσεις μόνο οι οποίες δεν δρομολογούνται στο

	internet, αν υπάρχουν public internet διευθύνσεις ή αν χρησιμοποιείται συνδυασμός των δύο σύμφωνα με το RFC 1918 με πολλαπλούς δρομολογητές μέσα στο δίκτυο.
Enable experimental scripts	Η επιλογή αυτή επιτρέπει τη χρήση των plugins που θεωρούνται ότι βρίσκονται σε πειραματικό στάδιο, στο να χρησιμοποιηθούν κατά τη διάρκεια της σάρωσης. Αυτή η επιλογή καλό θα είναι να μην χρησιμοποιείται σε παραγωγικό περιβάλλον.
Thorough tests (slow)	Με αυτή την επιλογή ενεργοποιούνται πιο ενδελεχείς έλεγχοι σε συγκεκριμένα plugins που έχουν αυτή τη δυνατότητα, πχ. όταν ελέγχονται τα SMB shares το plugin θα προχωρήσει στην ανάλυση τριών επιπέδων αντί του ενός που θα πραγματοποιούσε χωρίς τη συγκεκριμένη επιλογή. Σημειώνεται ότι με τους πιο ενδελεχείς ελέγχους η σάρωση γίνεται αρκετά πιο "επιθετική" και "θορυβώδες" προς το δίκτυο, με καλύτερα όμως αποτελέσματα ως προς τις αδυναμίες.
Report verbosity	Με αυτή την επιλογή εμφανίζονται περισσότερες πληροφορίες σχετικά με την δραστηριότητα των plugin στην τελική αναφορά.
Report paranoia	Σε κάποιες περιπτώσεις το Nessus δεν έχει τη δυνατότητα να προσδιορίσει κατά πόσο μια ευπάθεια ευσταθεί στον απομακρυσμένο στόχο ή όχι (false positive / negative). Εάν η επιλογή τεθεί στο "**Paranoid**" τότε κάθε φορά που υπάρχει 'έστω και η υποψία για την ύπαρξη μιας ευπάθειας αυτή θα εμφανίζεται στην αναφορά (*πιθανότητα false positive*). Αν η επιλογή τεθεί στο "**Avoid false alarm**" τότε το Nessus δεν εμφανίζει την ευπάθεια στην τελική αναφορά παρόλο που μπορεί να υπάρξει κάποια ένδειξη αβεβαιότητας σε σχέση με τον απομακρυσμένο στόχο. (*πιθανότητα false negative*). Η προεπιλογή είναι το "**Normal**" και το οποίο είναι μια σταθμισμένη μέση λύση σε σχέση με τις υπόλοιπες

	επιλογές.
HTPP User-Agent	Η επιλογή ατή καθορίζει τον τύπο του προγράμματος περιήγησης (web browser client) το οποίο, το Nessus προσποιείται κατά τη διάρκεια μιας σάρωσης.
SSL certificate to use	Επιτρέπει στο Nessus τη χρήση πιστοποιητικού πελάτη SSL.
SSL CA to trust	Καθορίζει την αρχή έκδοσης πιστοποιητικών (certificate authority) την οποία θα εμπιστεύεται το Nessus.
SSL key to use	Καθορίζει το τοπικό κλειδί SSL που θα χρησιμοποιήσει το Nessus για την επικοινωνία με τον απομακρυσμένο στόχο.
SSL password for SSL key	Ο κωδικός διαχείρισης του SSL κλειδιού που καθορίστηκε.

HTTP cookies import

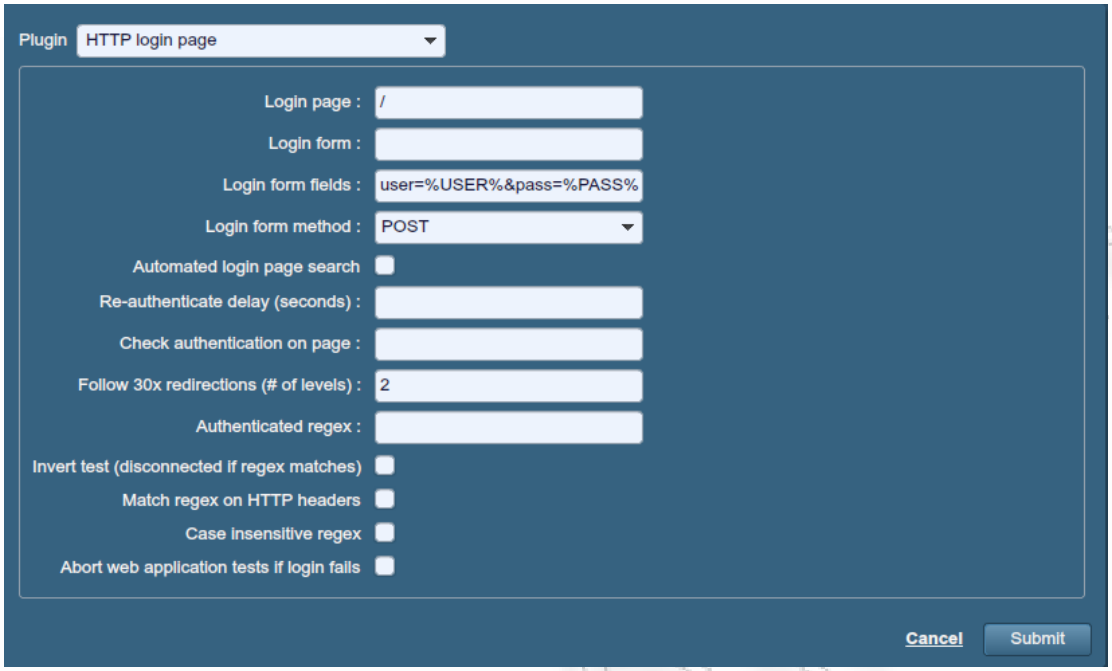
Με τη συγκεκριμένη επιλογή, δίνεται η δυνατότητα εισαγωγής κάποιου cookie από ένα άλλο λογισμικό πχ., ένα web browser, web proxy κτλ, κατά τη διεξαγωγή μιας σάρωσης σε διαδικτυακές εφαρμογές. Ένα αρχείο cookie μπορεί να μεταφορτωθεί ώστε το Nessus να το χρησιμοποιήσει κατά την προσπάθεια απόκτησης πρόσβασης σε μια web εφαρμογή. Το αρχείο των cookies πρέπει να βρίσκεται σε μορφή τύπου Netscape.



Εικόνα 42: Επιλογές Policy/Preferences/HTTP cookies Import

HTTP login page

Με τις επιλογές που προσφέρονται στο **HTTP login page** δίνεται η δυνατότητα στο Nessus να προβεί σε ελέγχους, σε προσαρμοσμένες διαδικτυακές εφαρμογές, οι οποίοι απαιτούν προηγουμένως την αυθεντικοποίηση σε αυτές.



Εικόνα 43: Επιλογές Policy/Preferences/HTTP login page

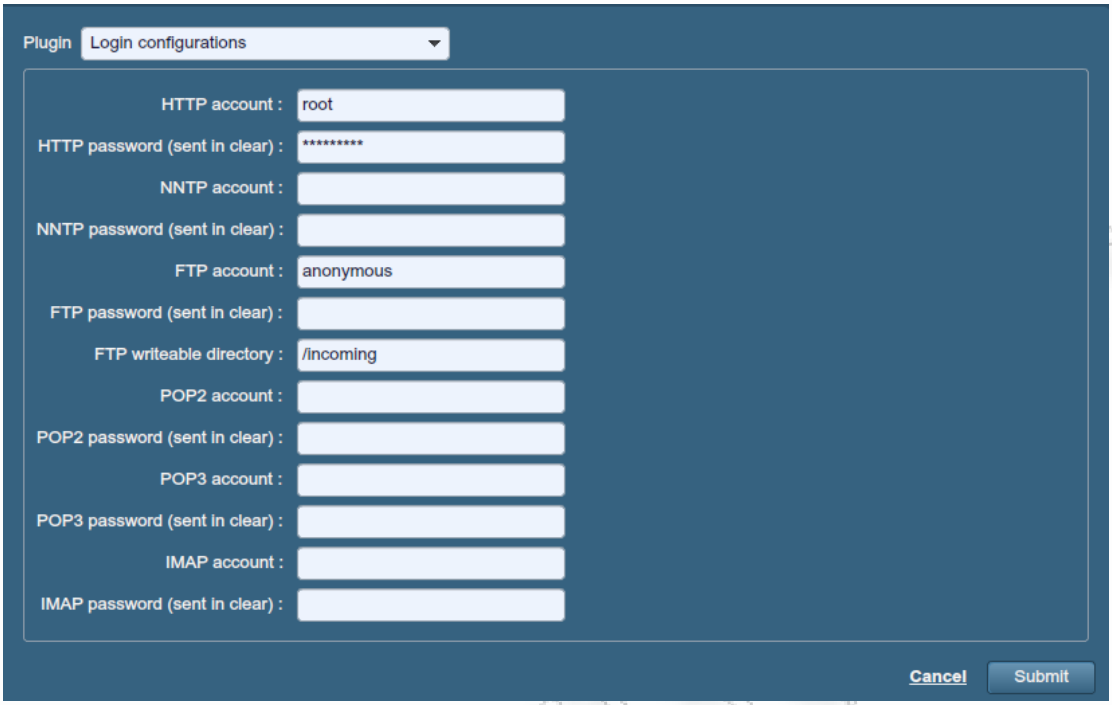
Επιλογή	Περιγραφή
Login page	Το σχετικό μονοπάτι της διεύθυνσης URI στο οποίο πραγματοποιείται η αυθεντικοποίηση στην εφαρμογή.
Login form	Η παράμετρος "**action**" που καθορίζεται στη μέθοδος της φόρμας στον κώδικα της εφαρμογής. Για παράδειγμα, στη φόρμα αυθεντικοποίησης **<form method="POST" name="auth_form" action="/login.php">** η παράμετρος που θα εισαχθεί θα είναι η "/login.php".
Login form fields	Καθορισμός των παραμέτρων ελέγχου ταυτότητας (π.χ., *login = %USER% & password =%PASS%*). Αν στον καθορισμό του πεδίου χρησιμοποιηθούν οι λέξεις-κλειδιά %USER% και %PASS%, αντί για τα πραγματικά στοιχεία αυθεντικοποίησης στην εφαρμογή, τότε αυτά θα αντικατασταθούν με τις τιμές που παρέχονται στα αντίστοιχα σχετικά πεδία της επιλογής "**Login configuration**". Το πεδίο αυτό χρησιμοποιείτε συνήθως, ώστε να οριστεί ίσως κάποια επιπλέον παράμετρος η οποία είναι απαραίτητη, για τη διαδικασία της αυθεντικοποίησης (π.χ., το όνομα μιας ομάδας ως η παράμετρος "group", είτε οποιαδήποτε άλλη πληροφορία

	είναι απαραίτητη και η οποία αποστέλλεται με τα δεδομένα αυθεντικοποίησης).
Login form method	Καθορίζει τη μέθοδο, GET ή POST, η οποία χρησιμοποιείται κατά την διεξαγωγή της αυθεντικοποίησης.
Automated login page search	Κατευθύνει το Nessus στο να ψάξει μόνο του και να ανακαλύψει την σελίδα αυθεντικοποίησης κάποιας εφαρμογής.
Re-authenticate delay (seconds)	Καθορισμός του χρόνου αναμονής μεταξύ των προσπαθειών αυθεντικοποίησης. Είναι μια χρήσιμη επιλογή σε περίπτωση που υπάρχει μηχανισμός κλειδώματος στην ανίχνευση επίθεσης τύπου brute force.
Check authentication on page	Το σχετικό μονοπάτι κάποιας προστατευόμενης διεύθυνσης URI στην ιστοσελίδα που η προβολή της απαιτεί προηγουμένως την αυθεντικοποίηση, ώστε να βοηθήσει το Nessus να καθορίσει αποδοτικότερα την κατάσταση αυθεντικοποίησης.
Follow 30x redirections (# of levels)	Καθορίζεται το πλήθος των ανακατευθύνσεων που θα ακολουθήσει το Nessus, σε περίπτωση λήψης κάποιου μηνύματος HTTP τύπου 30x για ανακατεύθυνση από το σύνδεσμο που του παρέχεται από τον εξυπηρετητή (Web Server).
Authenticated regex	Αναζήτηση συγκεκριμένης "συμβολοσειράς" (regular *expression pattern*) μετά την προσπάθεια αυθεντικοποίησης στην διαδικτυακή εφαρμογή, ώστε να διαπιστωθεί η επιτυχής αυθεντικοποίηση και να μην βασίζεται το Nessus μόνο στα μηνύματα HTTP τύπου 200 που λαμβάνονται από τον εξυπηρετητή (Web Server). π.χ., αναζήτηση της συμβολοσειράς *"Authentication Successful"*.
Invert test (disconnected if regex matches)	Αναζήτηση συγκεκριμένης "συμβολοσειράς" (regular *expression pattern*) μετά την προσπάθεια αυθεντικοποίησης στην διαδικτυακή εφαρμογή ώστε να διαπιστωθεί η ανεπιτυχής αυθεντικοποίηση π.χ., αναζήτηση της συμβολοσειράς *"Authentication failed"*

Match regex on HTTP header	Αναζήτηση συγκεκριμένης "συμβολοσειράς" (regular *expression pattern*) στην επικεφαλίδα της HTTP απάντησης (<header>) από τον εξυπηρετητή (Web Server) αντί για την αναζήτηση της συμβολοσειράς αυτής μέσα στο σώμα (<body>) του κειμένου της ιστοσελίδας που επιστρέφεται.
Case insensitive regex	Η συμβολοσειρά (regular *expression pattern*) στην οποία γίνεται η αναζήτηση είναι εξορισμού ευαίσθητη στο τύπο της συμβολοσειράς (**case sensitive**), με αυτή την επιλογή αγνοείται η συγκεκριμένη ιδιομορφία.
Abort web application tests if login fails	Εάν τα παρεχόμενα στοιχεία αυθεντικοποίησης δεν εξασφαλίσουν την πρόσβαση του Nessus στην εφαρμογή, τότε εγκαταλείπονται οι προσαρμοσμένοι έλεγχοι, που προϋποθέτουν αυθεντικοποίηση, στην διαδικτυακή εφαρμογή. Σημειώνεται ότι οι έλεγχοι της οικογένειας CGI plugin δεν απενεργοποιούνται.

Login Configuration

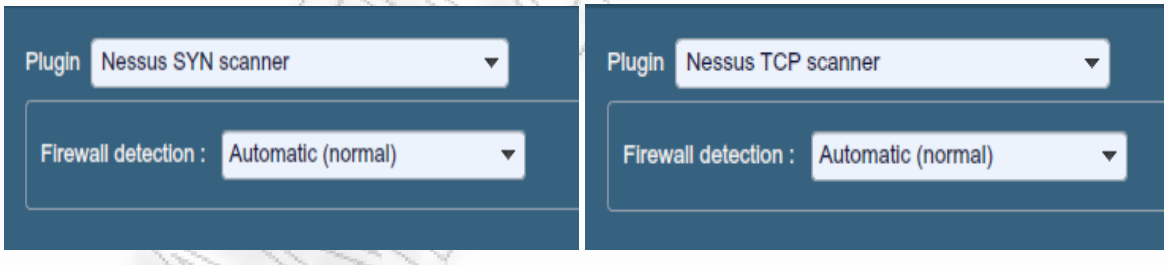
Μέσω της επιλογής "**Login configuration**" δίνετε η δυνατότητα στο Nessus να χρησιμοποιήσει διαφορετικές παραμέτρους αυθεντικοποίησης όταν διεξάγει ελέγχους μέσω των πρωτοκόλλων HTTP, NNTP, FTP, POP2, POP3 και IMAP. Οι παράμετροι αυθεντικοποίησης που παρέχονται μέσω της επιλογής Login Configuration, για το πρωτόκολλο HTTP χρησιμοποιούνται μόνο για την διεξαγωγή των βασικών και συνοπτικών ελέγχων ταυτότητας και αντικαθιστούν τις παραμέτρους *%USER%* και *%PASS%* που χρησιμοποιήθηκαν στην επιλογή "HTTP Login Page". Για διαδικτυακές εφαρμογές, οι οποίες είναι προσαρμοσμένες, είναι καλύτερο να παραμετροποιείται η επιλογή "HTTP Login page" που αναφέρθηκε πιο πάνω.



Εικόνα 44: Επιλογές Policy/Preferences/Login configurations

Nessus SYN Scanner και Nessus TCP Scanner

Τα Nessus SYN Scanner και Nessus TCP Scanner παρέχουν επιλογές οι οποίες επιτρέπουν τη καλύτερη ρύθμιση των σαρωτών SYN και TCP, ενσωματωμένοι σαρωτές που παρέχει το Nessus, για την ανίχνευση παρουσίας τείχους προστασίας.



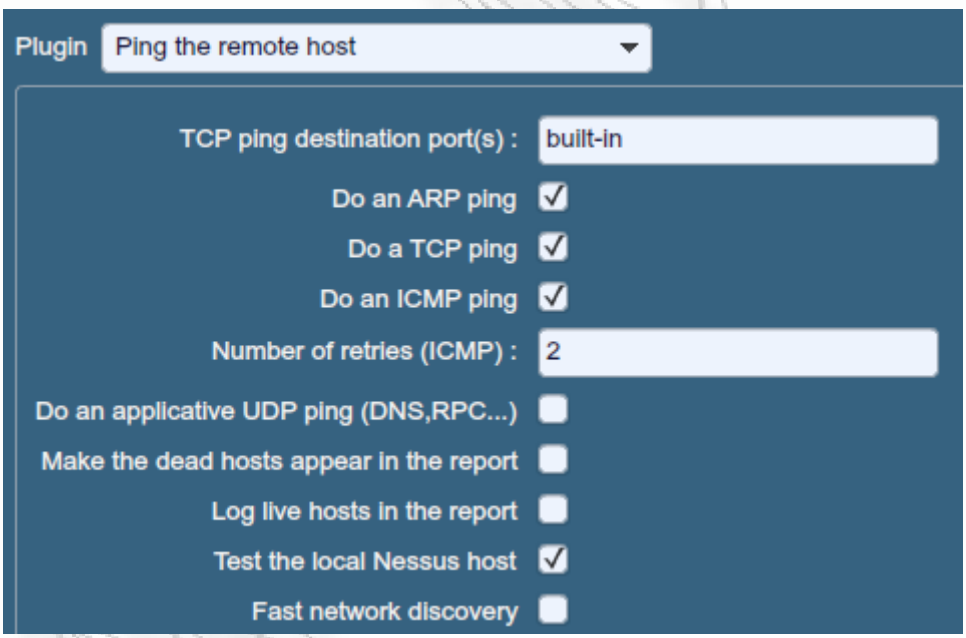
Εικόνα 45: : Επιλογές Policy/Preferences/Nessus SYN scanner και Nessus TCP scanner

Επιλογή	Περιγραφή
Automatic (normal)	Αυτή η επιλογή μπορεί να βοηθήσει στον εντοπισμό τείχους προστασίας μεταξύ του σαρωτή και του στόχου (προεπιλογή).
Disabled (softer)	Με την επιλογή αυτή γίνεται απενεργοποίηση του εντοπισμού τείχους προστασίας.
Do not detect RST rate	Με αυτή την επιλογή απενεργοποιείται η δυνατότητα

limitation (soft)	παρακολούθησης της συχνότητας λήψης πακέτων **resets** και της διαπίστωσης του κατά πόσο έχει ρυθμιστεί κάποιος περιορισμός από κάποια συσκευή δικτύου.
Ignore closed ports (aggressive)	Με αυτή τη ρύθμιση γίνεται προσπάθεια εκτέλεσης των έλεγχων από κάποιο plugin, ακόμα και αν η πόρτα η οποία αφορά το συγκεκριμένο έλεγχο φαίνεται να είναι κλειστή. Δεν συστήνεται η χρήση του σε παραγωγικό περιβάλλον.

Ping the remote host

Επιλογές που δίνουν τη δυνατότητα στο Nessus για τη λεπτομερή ρύθμιση της διαδικασίας ping κατά τη διάρκεια της ανακάλυψης των απομακρυσμένων στόχων στο δίκτυο. Αυτή η διαδικασία μπορεί να πραγματοποιηθεί μέσω διαφόρων πρωτοκόλλων όπως το ARP, TCP, ICMP ακόμα και μέσω του UDP.



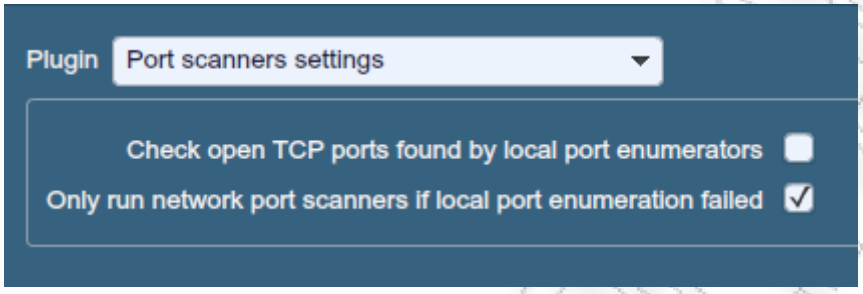
Εικόνα 46: Επιλογές Policy/Preferences/Ping the remote host

Επιλογή	Περιγραφή
TCP ping destination port(s)	Πεδίο στο οποίο ορίζεται η λίστα με τις πόρτες τις οποίες ελέγχονται μέσω της διαδικασίας **TCP ping**. Αν δεν είναι γνωστές εκ των προτέρων οι πόρτες που θα ελεγχθούν παραμένει ως προεπιλεγμένη τιμή η "built-in".
Number of Retries (ICMP)	Πεδίο που επιτρέπει τον ορισμό του πλήθους των

	προσπαθειών της διαδικασίας ping. Ως προεπιλογή καθορίζεται η τιμή **2**
Do an applicative UDP ping (DNS, RPC…)	Επιλογή που ενεργοποιεί τη διαδικασία **UDP ping** σε συγκεκριμένες πόρτες οι οποίες αφορούν συγκεκριμένες εφαρμογές-πρωτόκολλά, όπως για παράδειγμα το DNS (port 53), το RPC (port 111), το NTP (port 123), το RIP (port 520)
Make the dead hosts appear in the report	Με αυτή την επιλογή εμφανίζονται στην τελική αναφορά ως απενεργοποιημένοι, οι απομακρυσμένοι στόχοι οι οποίοι δεν ανταποκρίθηκαν στη διαδικασία ping.
Log live hosts in the report	Με αυτή την επιλογή εμφανίζονται στην τελικά αναφορά οι απομακρυσμένοι στόχοι οι οποίοι ανταποκρίθηκαν στη διαδικασία ping.
Test the local Nessus host	Αυτή η επιλογή επιτρέπει την εξαίρεση ή την ύπαρξη του υπολογιστή που φιλοξενεί τον Nessus server, μέσα στη λίστα με τους στόχους της σάρωσης. Αυτό χρησιμοποιείται όταν ο τοπικός υπολογιστής εμπίπτει στο εύρος του δικτύου στο οποίο πραγματοποιείται η σάρωση.
Fast network discovery	Μέσα στις προεπιλεγμένες ρυθμίσεις του το Nessus, όταν εκτελεί τη διαδικασία ping και λαμβάνει μια απάντηση από κάποιο στόχο, αυτόματα πραγματοποιεί την περαιτέρω "εξερεύνηση" του στόχου αυτού. Αυτό συμβαίνει ώστε να διαπιστωθεί ότι η συγκεκριμένη απάντηση δεν είναι απλά "θόρυβος" από κάποια συσκευή, αφού υπάρχουν συσκευές δικτύου οι οποίες απαντάνε σε όλες τις πόρτες από 1 – 65535 χωρίς όμως να κρύβεται όμως κάποια υπηρεσία πίσω από αυτές (για παράδειγμα κάποιοι proxies, load balancers κτλ.). Οι συγκεκριμένοι έλεγχοι μπορεί να πάρουν αρκετό χρόνο κατά τη διάρκεια της ανακάλυψης των στόχων και ειδικά αν οι στόχοι βρίσκονται πίσω από κάποιο τοίχος προστασίας. Με την συγκεκριμένη επιλογή, το Nessus δεν θα πραγματοποιήσει αυτούς τους ελέγχους.

Port scanner settings

Σε αυτή τη ρύθμιση παρέχονται δύο επιπλέον επιλογές οι οποίες σχετίζονται με την σάρωση των πορτών (ports) που εμφανίζονται ως ενεργές κατά τη διάρκεια πραγματοποίησης μιας σάρωσης.



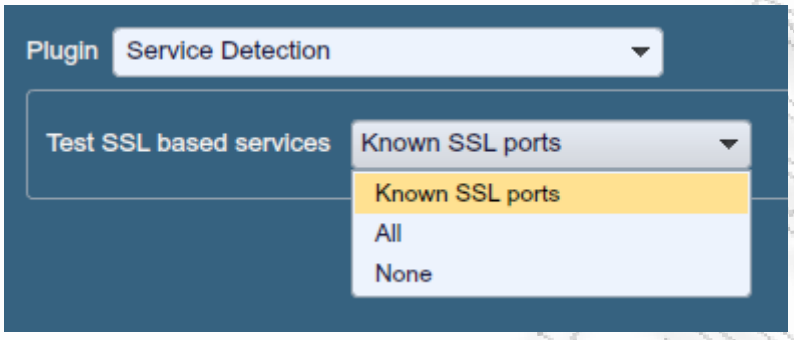
Εικόνα 47: Επιλογές Policy/Preferences/Port scanners settings

Επιλογή	Περιγραφή
Check open TCP ports found by local port enumerators	Με την επιλογή αυτή, αν μια πόρτα έχει ανακαλυφθεί ως ενεργή από κάποιο τοπικό απαριθμητή (enumerator), όπως για παράδειγμα το WMI ή το netstat, τότε το Nessus επαληθεύει κατά πόσο αυτή η πόρτα είναι ενεργή στον απομακρυσμένο στόχο. Αυτό βοηθάει στο να καθοριστεί κατά πόσο χρησιμοποιείται, στο δίκτυο, και κάποιας μορφής έλεγχος πρόσβασης όπως για παράδειγμα κάποιο τοίχος προστασίας, TCP wrappers κτλ.
Only run network port scanners if local port enumeration failed	Με αυτή την επιλογή το Nessus ξεκινάει την ανακάλυψη ανοικτών πορτών με τη χρήση κάποιου τοπικού απαριθμητή (π.χ., WMI, netstat κ.α.) και μόνο στην περίπτωση που όλοι αυτοί αποτύχουν θα βασιστεί στον ενσωματωμένο σαρωτή δικτύου τον οποίο διαθέτει.

Service Detection

Η ρύθμιση αυτή καθορίζει πως το Nessus θα εξετάσει για την ανακάλυψη υπηρεσιών που βασίζονται στο πρωτόκολλο SSL, στις πόρτες όπου ανακαλύπτει κατά τη διάρκεια μιας σάρωσης. Για παράδειγμα, κατά πόσοι θα ελέγξει μόνο τις γνωστές πόρτες στις οποίες ακούνε οι υπηρεσίες SSL (π.χ., port 443), όλες τις πόρτες ή ακόμα και καμία. Να σημειωθεί ότι ο έλεγχος σε όλες τις πόρτες, οι οποίες ανακαλύπτονται κατά τη διάρκεια μιας σάρωσης,

για την ύπαρξη κάποιας υπηρεσίας που βασίζεται στο πρωτόκολλο SSL, μπορεί να διαταράξουν την ορθή λειτουργία του απομακρυσμένου στόχου.



Εικόνα 48: Επιλογές Policy/Preferences/Service Detection

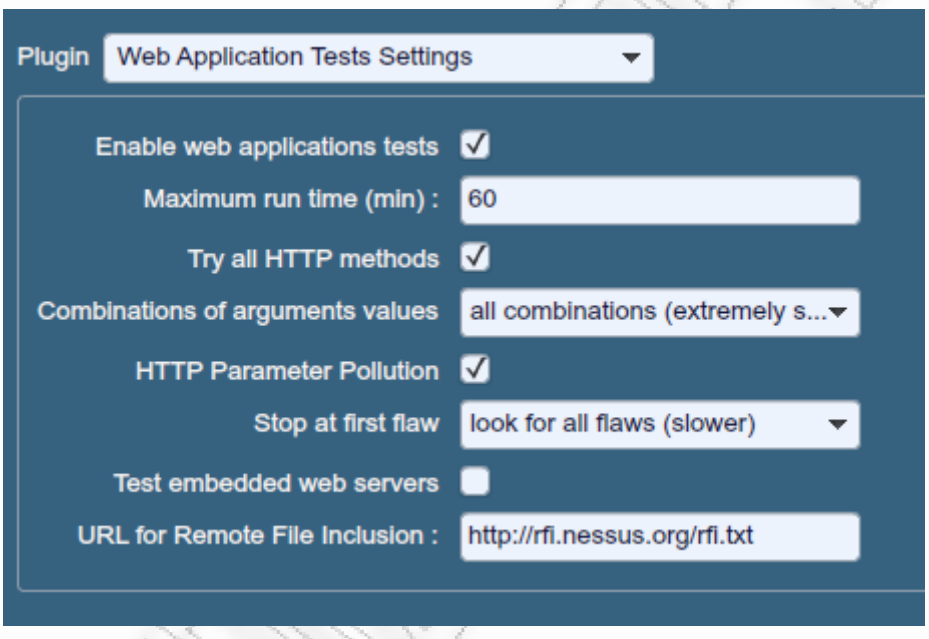
Web Application Test Settings

Αυτή αποτελεί μια από τις σημαντικότερες ρυθμίσεις του Nessus για τις ανάγκες της παρούσας εργασίας. Με τη ρύθμιση αυτή εξετάζονται, για αδυναμίες, οι παράμετροι προγραμματισμού συνάρτησης των CGIs (Common Gateway Interface) που ανιχνεύονται στον απομακρυσμένο στόχο μέσω της διαδικασίας web mirroring που ενσωματώνει το Nessus. Αυτό επιτυγχάνεται μέσω της προσπάθειας του Nessus, να εκτελέσει κώδικα που απορρέει από κοινά προγραμματιστικά λάθη τα οποία έχουν ανακαλυφθεί κατά καιρούς. Για παράδειγμα κώδικας ανακάλυψης αδυναμιών τύπου SQL Injections, cross-site scripting, command executions κτλ. Αυτοί οι έλεγχοι βασίζονται στα παρακάτω NASL (Nessus Attack Scripting Language) Plugins που είναι υπεύθυνα για την ανακάλυψη αυτών των αδυναμιών. Η ενεργοποίηση αυτών των ελέγχων γίνεται μέσω της επιλογής "Enable web applications tests".

- 11139, 42424, 42479, 42426, 42427, 43160 – SQL Injection (CGI abuses)
- 39465, 44967 – Command Execution (CGI abuses)
- 39466, 47831, 42425, 46193, 49067 – Cross-Site Scripting (CGI abuses: XSS)
- 39467, 46195, 46194 – Directory Traversal (CGI abuses)
- 39468 – HTTP Header Injection (CGI abuses: XSS)
- 39469, 42056, 42872 –File Inclusion (CGI abuses)
- 42055 – Format String (CGI abuses)
- 42423, 42054 - Server Side Includes (CGI abuses)

- 44136 – Cookie Manipulation (CGI abuses)
- 46196 – XML Injection (CGI abuses)
- 40406, 48926, 48927 – Error Messages
- 47830, 47832, 47834, 44134 – Additional attacks (CGI abuses)

Σημειώνεται ότι η παραπάνω λίστα με τα σχετικά plugins όπου χρησιμοποιούνται για την ανίχνευση αδυναμιών στις εφαρμογές ιστού, ανανεώνονται συχνά καθώς επίσης, από τις ρυθμίσεις σε αυτή την επιλογή, μπορεί να εξαρτώνται και επιπρόσθετα plugins.



Εικόνα 49: Επιλογές Policy/Preferences/Web Application Tests Settings

Επιλογή	Περιγραφή
Maximum run time (min)	Με αυτή την επιλογή ρυθμίζεται ο χρόνος, σε λεπτά, που δαπανάται στην εκτέλεση των δοκιμών στις εφαρμογές ιστού. Ο προεπιλεγμένος χρόνος σε αυτή είναι τα 60 λεπτά. **Αυτός είναι και ο μέγιστος χρόνος ο οποίος θα εφαρμοστεί γενικά για τη διάρκεια μιας σάρωσης που αφορά όλα τα CGI Plugins και όλες τις πόρτες**. Συνήθως ο χρόνος αυτός είναι υπεραρκετός για μικρές εφαρμογές, για μεγαλύτερες όμως εφαρμογές χρειάζεται και περισσότερος χρόνος για την ολοκλήρωση της σάρωσης.

Try all HTTP methods	Από προεπιλογή το Nessus χρησιμοποιεί μόνο αιτήματα GET του πρωτοκόλλου HTTP για τις ανάγκες μιας σάρωσης. Σε γενικές γραμμές, οι πιο σύνθετες εφαρμογές χρησιμοποιούν τη μέθοδο POST όταν ένας χρήστης υποβάλει δεδομένα στην εφαρμογή. Αυτή η ρύθμιση παρέχει πιο ενδελεχή έλεγχο, αλλά μπορεί να αυξήσει σημαντικά το χρόνο που απαιτείται για τη σάρωση γιατί το Nessus θα δοκιμάσει κάθε σενάριο-μεταβλητή του εκάστοτε ελέγχου, τόσο με GET όσο και με POST αιτήματα.
Combinations of arguments values	Με αυτή την επιλογή γίνεται διαχείριση των μεταβλητών που χρησιμοποιούνται κατά τη διάρκεια των αιτημάτων HTTP. Υπάρχουν οι εξής τρείς επιλογές: **One value**: Η επιλογή αυτή εξετάζει μια και μόνο τιμή στις μεταβλητές με κάθε συμβολοσειρά επίθεσης, χωρίς να γίνεται προσπάθεια περεταίρω εξέτασης με παραλλαγές της εισαγωγής των τιμών αυτών, στις μεταβλητές όπου δεν εισάγεται η συμβολοσειρά επίθεσης. Για παράδειγμα στην συμβολοσειρά επίθεσης: ***"/test.php?a=XSS&b=1&c=1"*** όπου το b και c επιτρέπουν και άλλες τιμές (σύμφωνα με τη διαδικασία ανακάλυψης των τιμών – crawling), δεν θα πραγματοποιηθεί κανένας άλλος συνδυασμός αυτών, παρά μόνο ο πρώτος. Αυτή είναι η γρηγορότερη μέθοδος σάρωσης αλλά και επίσης αυτή με τα λιγότερο αξιόπιστα παραγόμενα αποτελέσματα. **Some pairs :** Η συγκεκριμένη επιλογή είναι μια μέση κατάσταση μεταξύ των επιλογών One Value και All pairs **All pairs (slower but efficient):** Με την επιλογή αυτή θα εξεταστεί σε κάθε συμβολοσειρά επίθεσης ένας συνδυασμός των τιμών των μεταβλητών αυτής, με κάθε μεταβλητή να περνάει όλες τις πιθανές τιμές της, αλλά με τις υπόλοιπες να διατηρούν την πρώτη τους τιμή κάθε

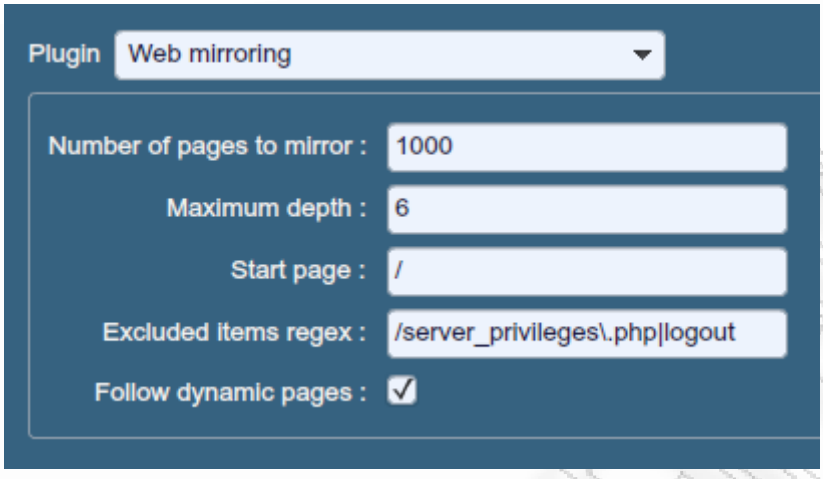
	φορά. Για παράδειγμα στη συμβολοσειρά: ***"/test.php?a=XSS&b=1&c=1&d=1"*** με τις τιμές των b,c,d να κυμαίνονται από το 1 – 3 θα αλλάζει την τιμή της μια μεταβλητή κάθε φορά μέχρι το 3 με όλες τις υπόλοιπες να διατηρούν την τιμή 1. Σε αυτό το παράδειγμα η συμβολοσειρά ***"/test.php?a=XSS&b=3&c=3&d=3"*** δεν θα εξεταστεί ποτέ. **Some combinations :** Η συγκεκριμένη επιλογή είναι μια μέση κατάσταση μεταξύ των επιλογών All pairs και All combinations **All combinations (extremely slow):** Αυτή η μέθοδος ελέγχου θα κάνει μια πλήρης και εξαντλητική δοκιμή όλων των δυνατών συνδυασμών συμβολοσειρών επίθεσης, με όλες τις έγκυρες τιμές εισόδου σε μεταβλητές, οι οποίες ανακαλύφθηκαν κατά τη διαδικασία crawling. Αυτός ο έλεγχος μπορεί να πάρει πάρα πολύ χρόνο για να τελειώσει τη σάρωση.
HTTP Parameter Pollution	Κατά τη διάρκεια πραγματοποίησης ελέγχου στις δικτυακές εφαρμογές, γίνεται προσπάθεια από το Nessus να παρακαμφθούν οι μηχανισμοί ελέγχου περιεχομένου (filtering mechanisms), με την εισαγωγή της κανονικής μεταβλητής παράλληλα με την μεταβλητή ελέγχου. Για παράδειγμα, μια κανονική δοκιμή ελέγχου SQL Injection μπορεί να ήταν η εξής: ***"/target.cgi?a='&b=2"***. Με τη χρήση της συγκεκριμένης επιλογής η δοκιμή θα έμοιαζε ως εξής: ***"/target.cgi?a='&a=1&b=2".***
Stop at first flaw	Αυτή η μεταβλητή καθορίζει το πότε μια καινούρια αδυναμία-ευπάθεια στοχοποιείται και ελέγχεται. Αυτό έχει αντίκτυπο στο επίπεδο των scripts (δηλαδή των script που πραγματοποιούν τους ελέγχους). Αν για παράδειγμα ανακαλυφθεί μια ευπάθεια τύπου XSS δεν θα διακοπεί ο έλεγχος για τυχόν SQL Injection ή Header Injection, αλλά

	θα υπάρχει τουλάχιστο μια αναφορά από τον κάθε τύπο ευπάθειας σε μια δεδομένη πόρτα στην τελική αναφορά, εκτός και αν έχει γίνει ήδη προηγηθεί η επιλογή της μεταβλητής **"Thorough Tests"**. Υπάρχουν οι εξής τέσσερεις επιλογές: **per CGI:** Με την πρώτη ανακάλυψη κάποιας ευπάθειας από κάποιο script για κάποιο συγκεκριμένο CGI, το Nessus αυτόματα προχωράει στο επόμενο CGI στον ίδιο εξυπηρετητή ή αν δεν υπάρχει άλλο CGI, τότε προχωράει στην επόμενη πόρτα/στόχο. Αυτή η ρύθμιση είναι και η προεπιλογή. **per port (quicker):** Με την πρώτη ανακάλυψη κάποιας ευπάθειας σε ένα στόχο (στην υπηρεσία Web Server) από κάποιο script, το Nessus διακόπτει την εκτέλεση και προχωράει στην επόμενη υπηρεσία Web Server η οποία υπάρχει, είτε στον ίδιο στόχο, είτε σε κάποιον άλλο, σε κάποια διαφορετική πόρτα (port). **per parameter (slow):** Με την πρώτη ανακάλυψη ενός τύπου ευπάθειας σε μια παράμετρο CGI από κάποιο script (π.χ., XSS), το Nessus προχωράει στην εξέταση της επόμενης παραμέτρου του ίδιου CGI, είτε στο επόμενο γνωστό CGI, είτε στην επόμενη πόρτα/εξυπηρετητή. **look for all flaws (slower):** σε αυτή την επιλογή, πραγματοποιούνται εκτεταμένες δοκιμές, ανεξάρτητα από τις αδυναμίες που έχουν ανακαλυφθεί μέχρι στιγμής. Με τη συγκεκριμένη επιλογή μπορεί να παραχθούν βερμπαλιστικές τελικές αναφορές και δεν συνίσταται η χρήση της στις περισσότερες περιπτώσεις.
Test embedded web servers	Με αυτή την επιλογή πραγματοποιείται η σάρωση ενσωματωμένων Web Servers που "τρέχουν" στους

	απομακρυσμένους στόχους. Συχνά οι συγκεκριμένοι servers είναι στατικοί και δεν περιέχουν προσαρμόσιμα CGI scripts, καθώς επίσης είναι και επιρρεπείς σε κινδύνους "κατάρρευσης" ή άρνησης παροχής υπηρεσιών. Συνίσταστε η χρήση της συγκεκριμένης επιλογής για σάρωση αποκλειστικά και μόνο των ενσωματωμένων Web Servers σε απομακρυσμένους στόχους.
URL for Remote File Inclusion	Κατά τη διάρκεια εκτέλεσης ελέγχων για Remote File Inclusion (RFI), μέσω αυτής της επιλογής προσδιορίζεται ένα συγκεκριμένο αρχείο από ένα απομακρυσμένο στόχο για να χρησιμοποιηθεί στους ελέγχους που εκτελούνται. Ως προεπιλογή υπάρχει ορισμένο ένα ασφαλές αρχείο στους εξυπηρετητές της Tenable συγκεκριμένα για την εκτέλεση των RFI ελέγχων. Στην περίπτωση που ο εξυπηρετητής όπου φιλοξενεί το Nessus δεν έχει σύνδεση προς το Internet μπορεί να χρησιμοποιηθεί ένα αρχείο που φιλοξενείται σε κάποιο εσωτερικό εξυπηρετητή στη θέση του απομακρυσμένου ώστε να παραχθούν σωστά αποτελέσματα σχετικά με τους συγκεκριμένους ελέγχους.

Web Mirroring

Η επιλογή "**web mirroring**" θέτει τις παραμέτρους διαμόρφωσης, του εγγενούς βοηθητικού προγράμματος του Nessus, όπου πραγματοποιεί τον αντικατοπτρισμό περιεχομένου των δικτυακών εφαρμογών (web mirroring-crawl). Το Nessus θα αντικατοπτρίσει (crawl) το περιεχόμενο κάποιας εφαρμογής ιστού ώστε να το αναλύσει αργότερα, αποδοτικότερα για τυχόν ευπάθειες και να βοηθήσει στην ελαχιστοποίηση των πιθανών επιπτώσεις στον εξυπηρετητή από κάποια επίθεση.



Εικόνα 50: Επιλογές Policy/Preferences/Web mirroring

Επιλογή	Περιγραφή
Number of pages to mirror	Προσδιορισμός του μέγιστου αριθμού από τις σελίδες που θα αντικατοπτριστούν
Maximum depth	Προσδιορισμός του μέγιστου αριθμού από τους συνδέσμους όπου το Nessus θα ακολουθήσει κατά την σάρωση για κάθε αρχική σελίδα που ανακαλύπτει.
Start page	Προσδιορισμός του σχετικού μονοπατιού URI της αρχικής σελίδας της εφαρμογής που θα ελεγχθεί. Στην περίπτωση που υπάρχουν περισσότερες από μια εφαρμογές (RUIs) που επιθυμείτε η σάρωση τους, μπορεί να γίνει διαχωρισμός μεταξύ τους με το σύμβολο "**:**" π.χ., "/:/webApp1:/webApp2" κ.ο.κ.
Excluded items regex	Προσδιορισμός των τμημάτων (regular expression strings) που θα αποκλειστούν από τον έλεγχο-σάρωση μιας ιστοσελίδας. Για παράδειγμα, για τον αποκλεισμό του φακέλου manuals και όλων των perl scripts, θα εισάγουμε στο συγκεκριμένο πεδίο τα εξής: **(^/manual)|(\.pl(\?.*)?$)**
Follow dynamic pages	Με αυτή την επιλογή το Nessus θα ακολουθήσει όποιο δυναμικό σύνδεσμο εμφανιστεί στην ιστοσελίδα (εάν η εφαρμογή έχει δυναμικούς συνδέσμους). Η επιλογή αυτή εμφανίζει την πιθανότητα να ξεπεραστούν ακόμα και οι μέγιστες τιμές που καθορίστηκαν πιο πάνω.

Κεφαλαίο 3 (Παραμετροποίηση και Ανάλυση Στοιχείων)

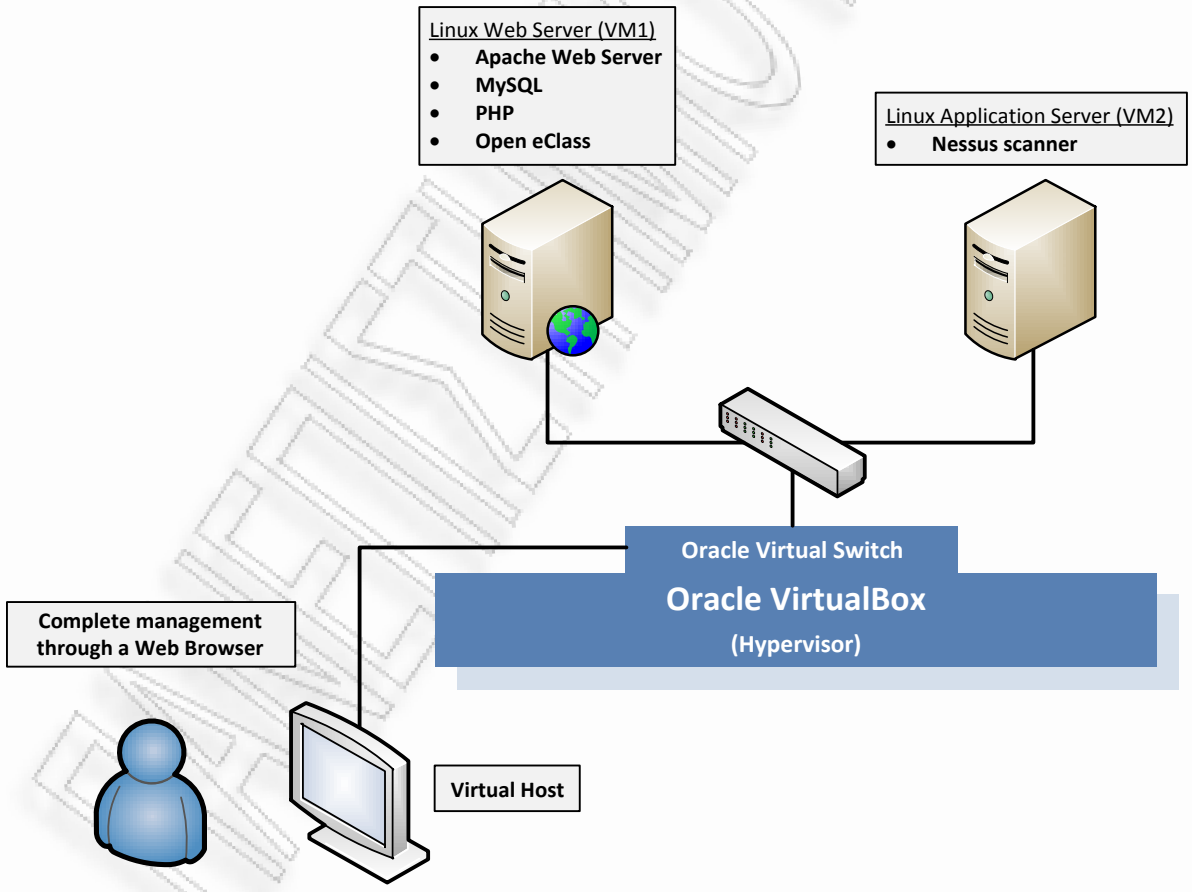
3.1. Εισαγωγή

Στα πλαίσια της παρούσας εργασίας πραγματοποιείται τόσο σε πραγματικές συνθήκες, όσο και σε ελεγχόμενο περιβάλλον, ένας αριθμός από ελέγχους ασφάλειας σε ένα αριθμό από ενεργούς ιστότοπους και συγκεκριμένα στους ιστότοπους του τμήματος Ψηφιακών Συστημάτων του Πανεπιστημίου Πειραιώς. Στους συγκεκριμένους ιστότοπους εκτός από την επίσημη ιστοσελίδα του τμήματος φιλοξενούνται επίσης και κάποιες εφαρμογές δικτύων, στις οποίες πραγματοποιήθηκε έλεγχος ασφάλειας με βάση διαφορετικών σεναρίων και συνθηκών.

Θα πρέπει να σημειωθεί ότι με αυτή την εργασία γίνεται προσπάθεια μέσω ενός αυτοματοποιημένου εργαλείου να δοθεί η δυνατότητα έλεγχου των εφαρμογών όπου ήδη έχουν αναπτυχθεί, ως προς τις ευπάθειες που ίσως ο κώδικας τους να περιέχει. Η συγκεκριμένη λύση, σίγουρα δεν αποτελεί πανάκεια, όπως και καμία άλλη αυτοματοποιημένη διαδικασία ανακάλυψης ευπαθειών, λόγω της πολυπλοκότητας της φύσης των σημερινών εφαρμογών και των πολυπληθών παραμέτρων που απαρτίζουν το συγκριμένο πρόβλημα. Παρέχει όμως τη δυνατότητα πραγματοποίησης ελέγχων σε άτομα που δεν έχουν άμεση σχέση με το πεδίο της ασφάλειας εφαρμογών καθώς όμως και σε ελεγκτές ασφάλειας να αποκτήσουν μια γενική εικόνα σχετικά με την ευρωστία των εφαρμογών που ελέγχονται και δη των εφαρμογών με μεγάλη έκταση, που πολλές φορές είναι αδύνατο και ασύμφορο να ελεγχθούν χειροκίνητα.

Απαραίτητες προϋποθέσεις για τον έλεγχο, ήταν ότι σε καμία περίπτωση δεν θα έπρεπε να παρεμποδισθεί η λειτουργία του ιστότοπου καθώς και των εφαρμογών και οπωσδήποτε να μην προκληθεί οποιαδήποτε βλάβη, μόνιμη ή προσωρινή, σε κάποιο από τα συστατικά του. Επίσης δεν λήφθηκαν υπόψη κανενός είδους πληροφορίες σχετικά με τη δομή και τα συστατικά του ιστότοπου και των εφαρμογών. Γι’ αυτό το λόγο χρησιμοποιείται και η Black Box προσέγγιση της δοκιμής διείσδυσης για τον έλεγχο ασφάλειας των ιστότοπων του τμήματος, ώστε τα αποτελέσματα να προσεγγίζουν όσο το δυνατό περισσότερο, ένα πραγματικό σενάριο από κάποιον επιτιθέμενο, οποίος δεν γνωρίζει αρχικά καμία πληροφορία για την όποια υποδομή και αρχιτεκτονική των συγκεκριμένων εφαρμογών.

Στη συνέχεια του κεφαλαίου θα παρουσιαστούν και αναλυθούν δύο πολιτικές σάρωσης (διαμόρφωση ρυθμίσεων ελέγχου). Συγκεκριμένα στη μια από αυτές παρουσιάζεται η δημιουργία μιας γενικευμένης, πρώτου επιπέδου, πολιτική σάρωσης μέσω της οποίας πραγματοποιείται ο έλεγχος ασφάλειας στον επίσημο ιστότοπο του τμήματος, ενώ στη δεύτερη παρουσιάζεται αναλυτικά η δημιουργία μιας στοχευμένης και εξειδικευμένης πολιτικής σάρωσης με την οποία πραγματοποιούνται οι έλεγχοι ασφάλειας πάνω σε συγκεκριμένη δικτυακή εφαρμογή. Για τους λόγους που αναφέρθηκαν στην προηγούμενη παράγραφο ο έλεγχος μέσω της στοχευμένης πολιτικής δεν πραγματοποιείται σε κάποια εφαρμογή του επίσημου ιστότοπου του τμήματος, αλλά σε ένα ελεγχόμενο περιβάλλον που δημιουργήθηκε ειδικά για την εξυπηρέτηση του σκοπού αυτού. Στην ακόλουθη εικόνα παρουσιάζεται η αρχιτεκτονική του περιβάλλοντος αυτού.



Εικόνα 51: Παρουσίαση υποδομής ελεγχόμενου περιβάλλοντος για τη διεξαγωγή σάρωσης στοχευμένης πολιτικής

3.2. Συλλογή πληροφοριών και παραμετροποίηση του σαρωτή Nessus

Όπως ήδη έχει αναφερθεί στο προηγούμενο κεφάλαιο, όπου έγινε η παρουσίαση του συγκεκριμένου εργαλείου˙, το Nessus είναι ένας σαρωτής με υποστήριξη πολλαπλών δικτύων και λειτουργικών συστημάτων, που βασίζεται στο μοντέλο πελάτη-εξυπηρετητή. Ο εξυπηρετητής (σαρωτής), υποστηρίζεται στα λειτουργικά συστήματα Windows, Linux, Mac OS και UNIX ενώ ο πελάτης είναι σχεδιασμένος σε μορφή δικτυακής εφαρμογής, με υποστήριξη από σχεδόν κάθε γνωστό φυλλομετρητή ιστού με ενεργοποιημένη την τεχνολογίας flash, καθώς επίσης και με τη μορφή εφαρμογής για έξυπνα τηλέφωνα στα λειτουργικά iOS και Android.

Το λογισμικό παρέχεται σε δύο κύριες μορφές αδειοδότησης:

- Professional Feed: για εμπορική χρήση, με ετήσια συνδρομή και παροχή υποστήριξης από την Tenable Inc.
- Home Feed: για προσωπική χρήση μόνο, χωρίς χρέωση και με περιορισμούς λειτουργικότητας [παράλληλη σάρωση μόνο σε 16 IP, χωρίς χρονοπρογραμματισμό σαρώσεων και χωρίς ελέγχους συμβατότητας συστημάτων].

Σύντομη ορολογία χρήσης του Nessus

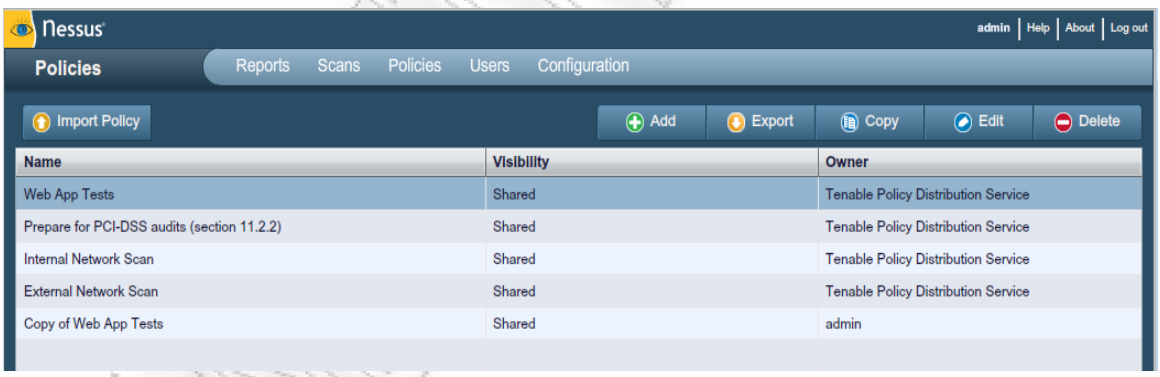
- **Policy** (Πολιτική): Διαμόρφωση των ρυθμίσεων για τη διεξαγωγή μιας σάρωσης.
- **Scan** (Σάρωση): Η συσχέτιση μιας λίστας με IP ή/και ονομάτων τομέα με μια συγκεκριμένη πολιτική.
- **Report** (Αναφορά): Η εμφάνιση των αποτελεσμάτων μιας συγκεκριμένης σάρωσης
- **Plugin** (Πρόσθετο): Ένας έλεγχος ασφάλειας, ή ένα παράθυρο για ρυθμίσεις σάρωσης
- **Plugin family** (Οικογένεια Πρόσθετων): Μια ομάδα από plugins (πρόσθετα) που περιέχουν κοινά στοιχεία (π.χ. Cisco, FTP, Web Servers, κτλ.)

3.2.1. Δημιουργία γενικής πολιτικής σάρωσης εφαρμογών ιστού

Στόχος της ύπαρξης μιας γενικής ή γενικευμένης πολιτικής σάρωσης είναι η δημιουργία μιας τέτοιας πολιτικής, που θα αποτελέσει τη βάση για τη σάρωση "άγνωστων" ή καινούριων εφαρμογών ιστού. Μέσω αυτής της σάρωσης, θα πραγματοποιείται μια πιο επιφανειακή και ήπια σάρωση που, μεταξύ άλλων, θα δίνεται και η δυνατότητα συλλογής πληροφοριών σχετικά με την εφαρμογή για την περεταίρω διερεύνηση της αργότερα και τη δημιουργία ίσως μιας πιο εξειδικευμένης πολιτικής σάρωσης. Σε αυτή την πολιτική θα οριστούν οι παράμετροι λειτουργίας, όπου παρουσιάζουν αποτέλεσμα και ευρήματα στη πλειοψηφία των εφαρμογών ιστού. Να σημειωθεί επίσης ότι η συγκεκριμένη πολιτική θα αποτελεί τη βάση για τη δημιουργία μιας εξειδικευμένης πολιτικής σάρωσης για συγκεκριμένες και στοχευμένες εφαρμογές ιστού αργότερα.

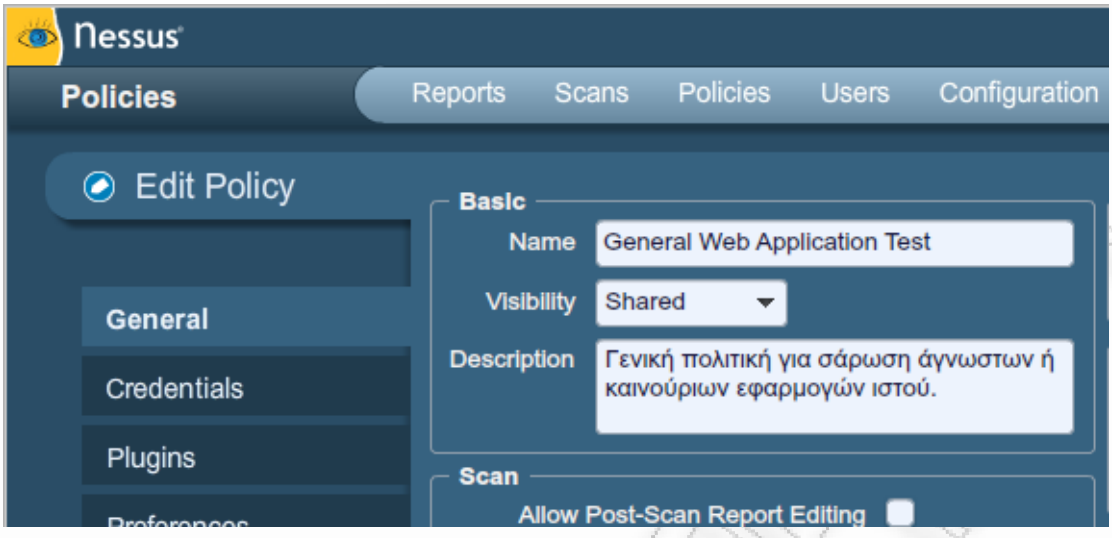
3.2.1.1. Παραμετροποίηση επιλογών καρτέλας Policies

Αρχικά μέσω της καρτέλας Policies γίνεται επιλογή της προϋπάρχουσας πολιτικής "Web App tests" που δημιουργήθηκε από την Tenable και επιλέγεται η αντιγραφή της, όπως φαίνεται στην ακόλουθη εικόνα, με αποτέλεσμα τη δημιουργία μιας νέας πολιτικής με όνομα "Copy of Web App Tests".



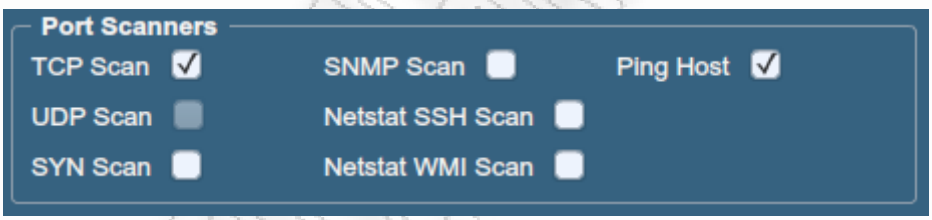
Εικόνα 52: Αντιγραφή Policy

Κατόπιν ξεκινάει η παραμετροποίηση της καινούριας πολιτικής μέσω της επιλογής "**Edit**". Δίνεται ένα καινούριο όνομα σε αυτή το οποίο θα υποδηλώνει και το σκοπό της ώστε με μια ματιά να γίνεται κατανοητό τι αντιπροσωπεύει, μαζί με μια σύντομη περιγραφή που ξεκαθαρίζει τον σκοπό της συγκεκριμένης πολιτικής.



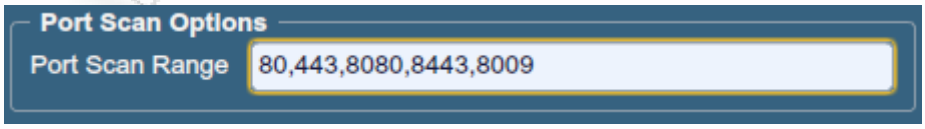
Εικόνα 53: Παραμετροποίηση policy/Basic

Στο τμήμα "Port Scanners" που αντιπροσωπεύει τη σάρωση για "ανοικτές" πόρτες στον απομακρυσμένο στόχο καλό είναι να παραμείνουν ως επιλογή το "**TCP Scan**" και το "**Ping Host**" ώστε να δοθεί η δυνατότητα στον σαρωτή να ανακαλύψει τυχόν υπηρεσίες που κρύβονται πίσω από γνωστές πόρτες αλλά και να κρατάει δεδομένα και στατιστικά σχετικά με τη σύνδεση στον απομακρυσμένο στόχο.



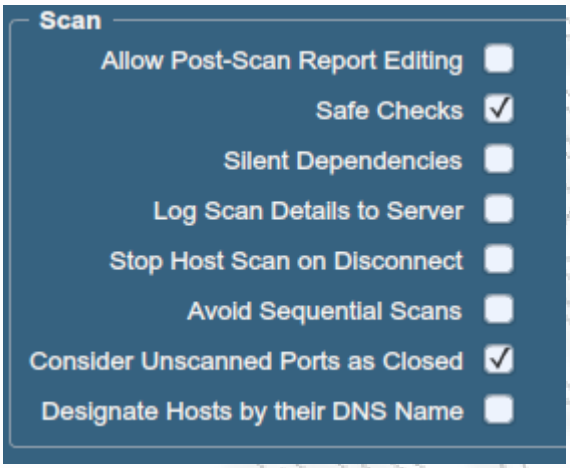
Εικόνα 54: Παραμετροποίηση policy/Port Scanners

Στο τμήμα "Port Scan Options" προσδιορίζεται το πλήθος από τις πόρτες που θα εξεταστούν κατά τη σάρωση. Κατά τη δημιουργία της γενικευμένης πολιτικής, επειδή υπάρχει περίπτωση μη πρότερης γνώσης των ports όπου ακούει η εφαρμογή συνηθίζεται να εισάγεται η τιμή "default" στην οποία σαρώνεται ένα πλήθος από γνωστές πόρτες για υπηρεσίες, η τιμή "all" κατά την οποία γίνεται σάρωση σε όλο το εύρος των ports (1-65535). Στην προκειμένη περίπτωση γίνεται εισαγωγή και εξέταση σε έξι από τις πιο γνωστές πόρτες που εξυπηρετούν εφαρμογές ιστού τις **80, 443, 8080, 8443, 8009**.



Εικόνα 55: Παραμετροποίηση policy/Port Scan Options

Στο τμήμα "**Scan**" αφήνουμε μόνο τις επιλογές "**Safe Checks**" και "**Consider Unscanned Ports as Closed**" στις οποίες επιλογές η μεν πρώτη εξασφαλίζει την μη εκτέλεση επικίνδυνων exploits, δηλαδή, την ασφαλή σάρωση της δικτυακής εφαρμογής και η δεύτερη "επιταχύνει" κάπως τη διαδικασία της σάρωσης, αφού παρακάμπτει τελικά τους ελέγχους σε πόρτες οι οποίες εμφανίζονται κλειστές.

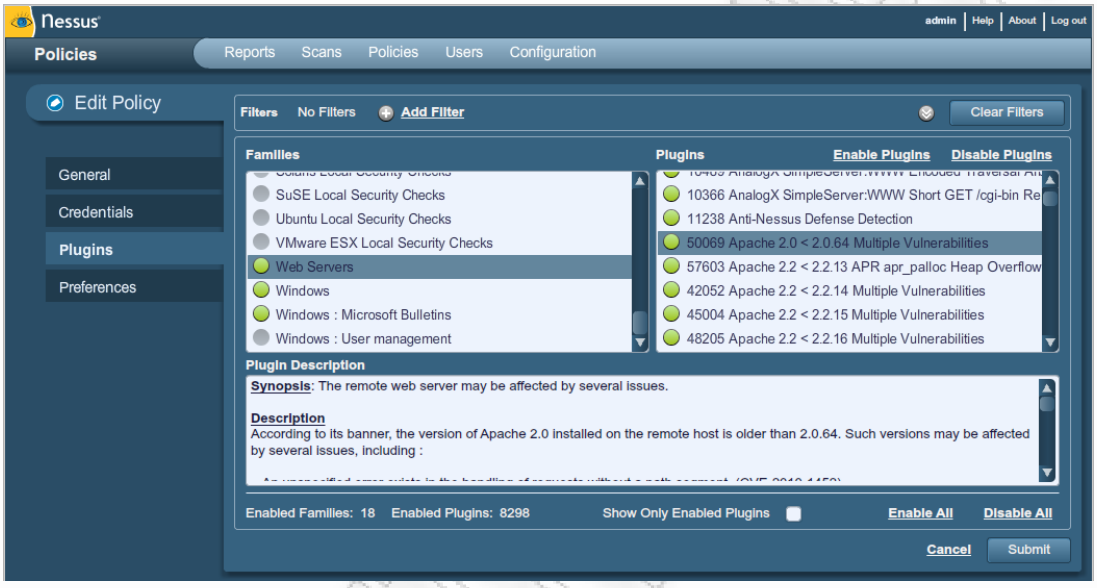


Εικόνα 56: Παραμετροποίηση policy/Scan

Στη συνέχεια θα γίνει επιλογή των plugins που πλησιάζουν και αφορούν περισσότερο τις εφαρμογές ιστού και είναι πιο κατάλληλοι για τη διεξαγωγή ελέγχων σε αυτές. Μετά την επιλογή της καρτέλας Plugins πραγματοποιείται κατάργηση όλων των οικογενειών από plugins, με τη επιλογή "Disable All" και στη συνέχεια επιλέγονται μόνο οι ακόλουθες οικογένειες:

- **Backdoors**
- **CGI Abuses**
- **CGI Abuses: XSS**
- **Cisco**
- **Databases**
- **FTP**
- **Firewalls**
- **Gain a shell remotely**
- **General**
- **Misc.**
- **Netware**
- **Peer-To-Pear File Sharing**
- **SMTP problems**
- **Service detection**
- **Settings**
- **Web Servers**
- **Windows**
- **Windows: Microsoft Bulletins**

Να σημειωθεί ότι ακόμα και όλες οι οικογένειες από plugins να έχουν επιλεγεί αυτό δεν σημαίνει ότι το Nessus θα πραγματοποιήσει όλους τους σχετικούς με αυτές τις οικογένειες ελέγχους, αφού είναι αρκετά "έξυπνο" ώστε να ξεχωρίσει ποίοι από αυτούς τους έλεγχους είναι απαραίτητοι να πραγματοποιηθούν. Μόνη εξαίρεση σε αυτό, αποτελούν οι έλεγχοι της οικογένειας CGI Abuses και CGI Abuses XSS όπου πραγματοποιούνται ούτως ή άλλως ακόμα και σε πόρτες που ίσως θεωρηθούν κλειστές, εξού και η προηγούμενη επιλογή "Consider Unscanned Ports as Closed".



Εικόνα 57: Παραμετροποίηση Policy/Plugins

Η καρτέλα Preferences, οδηγεί στη ρύθμιση των προηγμένων επιλογών που αφορούν μια γενικευμένη πολιτική σάρωσης. Οι ρυθμίσεις που χρειάζονται παραμετροποίηση ή/και επισκόπηση, βρίσκονται στις ακόλουθες καρτέλες από το μενού αναδίπλωσης:

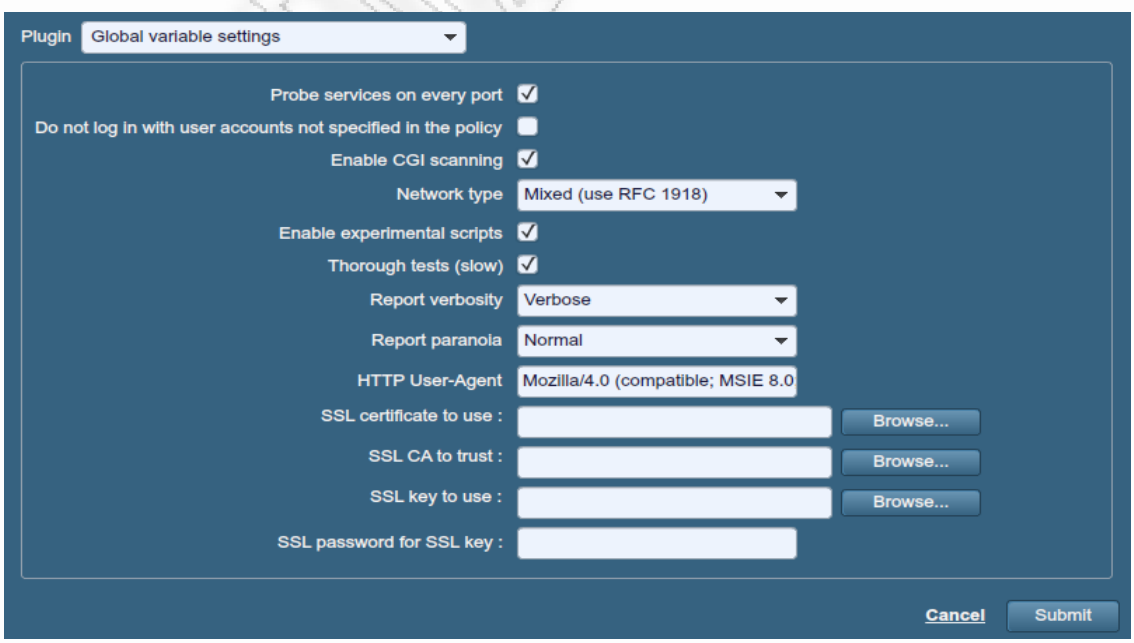
1. Global variable settings
2. HTTP login page
3. Login configuration
4. Web Application Tests Settings
5. Web mirroring

Ξεκινώντας από την καρτέλα Global variable settings θα πρέπει να βεβαιωθούμε για την επιλογή των ακόλουθων ρυθμίσεων.

- "**Probe service on every port**" όπου μας εξασφαλίζει τον έλεγχο σε κάθε πόρτα που εμφανίζεται ανοικτή κατά πόσο κρύβεται μια υπηρεσία web servicing πίσω από αυτήν.

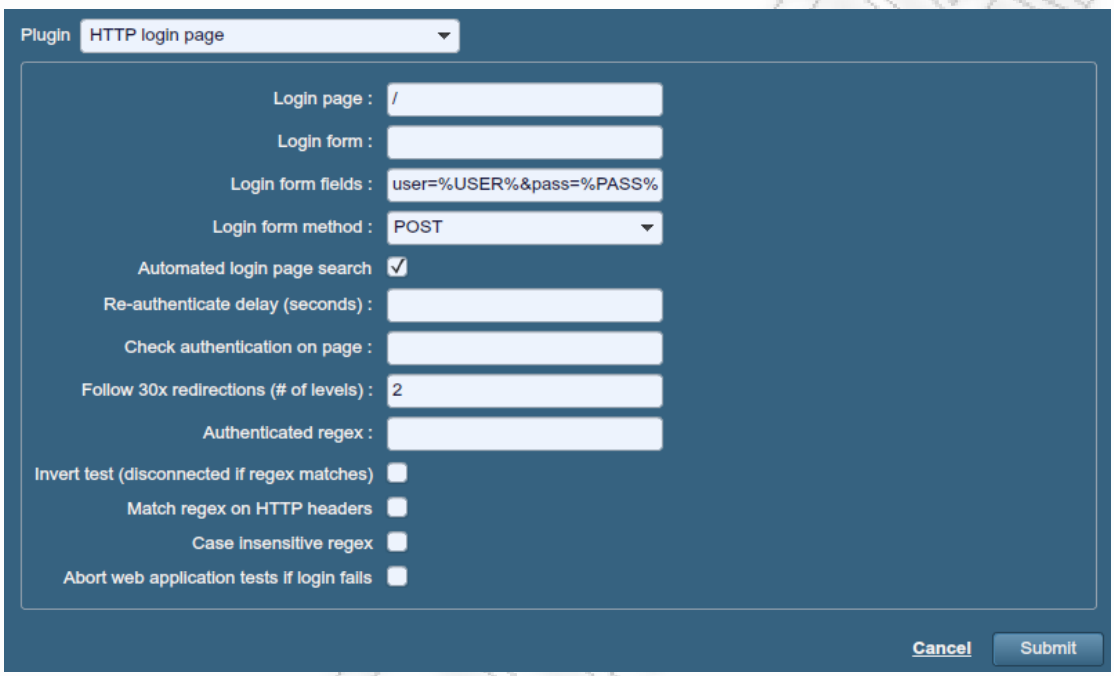
- "**Enable CGI scanning**" όπου εξασφαλίζει την εκτέλεση των plugins που βρίσκονται κάτω από τις οικογένειες CGI Abuses και CGI Abuses XSS.
- "**Enable experimental scripts**" με τη επιλογή της εκτελούνται κατά τη διάρκεια της σάρωσης τα plugins τα οποία έχουν χαρακτηριστεί ότι βρίσκονται σε πειραματικό στάδιο.

  Σημειώνεται ότι η βέλτιστη πρακτική σχετικά με αυτή την επιλογή, είναι να παραλείπεται σε περίπτωση που η σάρωση πραγματοποιείται σε κάποιο εξυπηρετητή που βρίσκεται στην παραγωγή.
- "**Thorough tests (slow)**" όπου προκαλεί την επίμονη και πιο ενδελεχή εκτέλεση συγκεκριμένων plugins. Για παράδειγμα τη βαθύτερη διείσδυση της διεργασίας crawl σε μια εφαρμογή που διαθέτει δυναμικούς συνδέσμους.
- "**Report verbosity**" τίθεται η ρύθμιση στο "**Verbose**" που θα έχει ως αποτέλεσμα να παραχθεί μια λεπτομερέστερη αναφορά, που θα παρέχει περισσότερες πληροφορίες, σχετικά με την δραστηριότητα κάποιων plugins.
- "**Report paranoia**" τίθεται η ρύθμιση στο "**Normal**" ώστε τα σφάλματα και οι ευπάθειες που θα εμφανίζονται στην τελική αναφορά, σχετικά με την εφαρμογή προς σάρωση, να διατηρούν μια μέση κατάσταση, χωρίς να αναφέρονται ευπάθειες που απλά υπάρχει υπόνοια αυτών αλλά ούτε και να μην αναφέρονται καθόλου αν υπάρχει αμφιβολία ύπαρξης τους. Είναι δηλαδή μια μέση κατάσταση στα false positives και false negatives που ίσως εμφανιστούν στην τελική αναφορά.



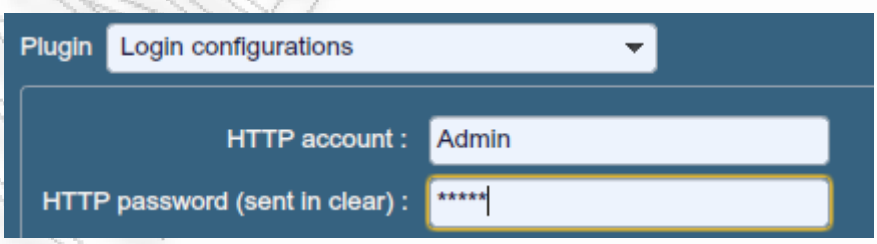
Εικόνα 58: Παραμετροποίηση Policy/Preferences/Global variable settings

Στην καρτέλα HTTP login page γίνεται επιλογή της ρύθμισης "**Automated login page search**" όπου εξασφαλίζει την αυτόματη εύρεση της σελίδας στην οποία πραγματοποιείται η διαδικασία της αυθεντικοποίησης, εάν αυτή υπάρχει, στην εφαρμογή που σαρώνεται. Αυτή η ρύθμιση μας παρέχει επιπλέον, τις λεπτομέρειες που θα χρειαστούν αργότερα για μια πιο εξειδικευμένη σάρωση στην οποία θα πραγματοποιηθεί μετά από αυθεντικοποίηση στην υπό εξέταση εφαρμογή.



Εικόνα 59: Παραμετροποίηση Policy/HTTP login page

Στην καρτέλα Login configuration και συγκεκριμένα στα πεδία **HTTP account** και **HTTP password (send in clear)**, τοποθετούνται δύο αλφαριθμητικές συμβολοσειρές που θα χρησιμοποιηθούν στην διαδικασία της αυθεντικοποίησης σε περίπτωση που ανακαλυφθεί μια τέτοια σελίδα. Για παράδειγμα οι συμβολοσειρές "Admin" και "Admin".

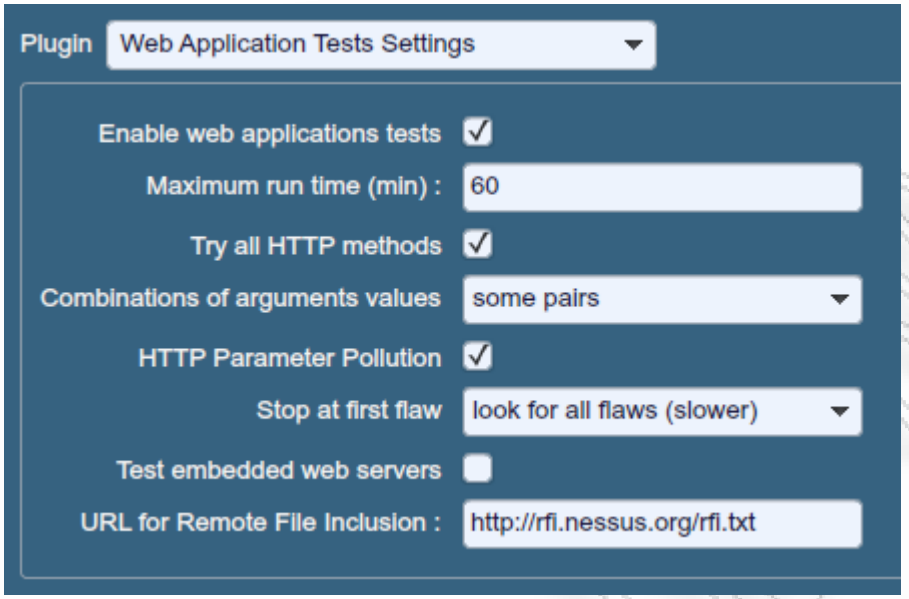


Εικόνα 60: Παραμετροποίηση Policy/Login configurations

Στην καρτέλα Web Application Tests Settings, μια από τις σημαντικότερες επιλογές ρυθμίσεων, γίνεται επιβεβαίωση ύπαρξης των παρακάτω επιλογών

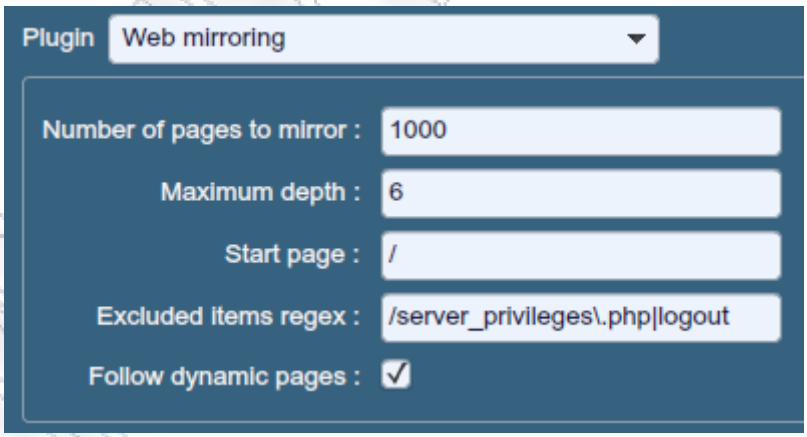
- "**Enable web application tests**" εάν αυτή η ρύθμιση δεν έχει επιλεγεί, τότε το Nessus θα πραγματοποιήσει ελέγχους μόνο για γνωστές ευπάθειες και δεν θα εκτελέσει τις δοκιμές για άγνωστες ευπάθειες της συγκεκριμένης εφαρμογή που σαρώνεται, τις λεγόμενες zero day vulnerabilities [14].
- "**Maximum run time (min)**" ορίζει το μέγιστο χρόνο εκτέλεσης, σε λεπτά, για κάθε CGI plugin, στην κάθε πόρτα που σαρώνεται και όχι το συνολικό χρόνο που θα διαρκέσει μια σάρωση. Η προεπιλογή των 60 λεπτών μπορεί να διατηρηθεί στην προκειμένη περίπτωση, εκτός και αν στο τέλος της σάρωσης παρατηρηθούν στην αναφορά χρονικές λήξεις κάποιων ελέγχων.
- "**Try all HTTP methods**" εξασφαλίζεται ο έλεγχος της εφαρμογής τόσο με τη μέθοδο GET όσο και με τη μέθοδο POST. Χωρίς την ενεργοποίησης της συγκεκριμένης επιλογής θα πραγματοποιηθούν έλεγχοι μόνο με τη μέθοδο GET.
- "**Combination of arguments values**" γίνεται επιλογή της ρύθμισης "**some pairs"**, όπου είναι η μέση κατάσταση των One value και All pairs, με την οποία πραγματοποιούνται έλεγχοι με συνδυασμό, τόσο σε αλφαριθμητικά επίθεσης όσο και σε έγκυρα δεδομένα της εφαρμογής.
- "**HTTP Parameter Pollution**" όπου πραγματοποιεί ελέγχους στα δεδομένα κάνοντας επαύξηση αυτών, με έγκυρα δεδομένα της εφαρμογής. Για παράδειγμα ένας απλός έλεγχος SQL injection μπορεί να εκτελεστεί ως εξής "/target.cgi?a='&b=2" ενώ με την συγκεκριμένη ρύθμιση θα εκτελεστεί ως "/target.cgi?a='&α=1&b=2".
- "Stop at first flaw" γίνεται ο καθορισμός της συνθήκης που αφορά το πότε θα προσχωρήσει το σάρωμα στον επόμενο στόχο. Η επιλογή "**look for all flaws**" είτε η "**per parameter**" είναι οι καταλληλότερες για την συγκεκριμένη περίπτωση, αφού θα πραγματοποιήσουν ελέγχους στη χειρότερη περίπτωση στην κάθε παράμετρο που έχει ανακαλυφθεί.
- "**Test embedded web servers**" μπορεί να προκαλέσουν προβλήματα κατά τη διάρκεια μιας σάρωσης αφού ελέγχουν ενσωματωμένους [15], ειδικού σκοπού εξυπηρετητές (π.χ. η δικτυακή διεπαφή που προσφέρουν διάφοροι δρομολογητές). Καλό είναι εάν επιθυμείται η εξέταση τέτοιου τύπου εξυπηρετητών να δημιουργείται μια πολιτική αποκλειστικά για το σκοπό αυτό.

[14] Zero-day vulnerabilities, http://en.wikipedia.org/wiki/Zero-day_attack
[15] Embedded Web Servers, http://en.wikipedia.org/wiki/Embedded_HTTP_server



Εικόνα 61: Παραμετροποίηση Policy/Web Application Tests Settings

Στην καρτέλα Web mirroring πρέπει απλά να υπάρχει η επιλογή της ρύθμισης "**Follow dynamic pages**" που εξασφαλίζει τη περιήγηση (crawl)[16] σε δυναμικούς συνδέσμους που ίσως δημιουργούνται στην εφαρμογή που σαρώνεται και κυρίως εάν αυτή είναι εφαρμογή σε Web 2.0 [17]. Η συγκεκριμένη ρύθμιση παίζει πρωταρχικό ρόλο στις στοχευμένες κυρίως σαρώσεις παρά στις γενικευμένες. Να σημειωθεί ότι με τη συγκεκριμένη επιλογή υπάρχει πιθανότητα να ξεπεραστεί το όριο που τίθεται στην παράμετρο "Number of pages to mirror".



Εικόνα 62: Παραμετροποίηση Policy/Web mirroring

[16] Web crawlers, http://en.wikipedia.org/wiki/Web_crawler
[17] Web 2.0, http://en.wikipedia.org/wiki/Web_2.0

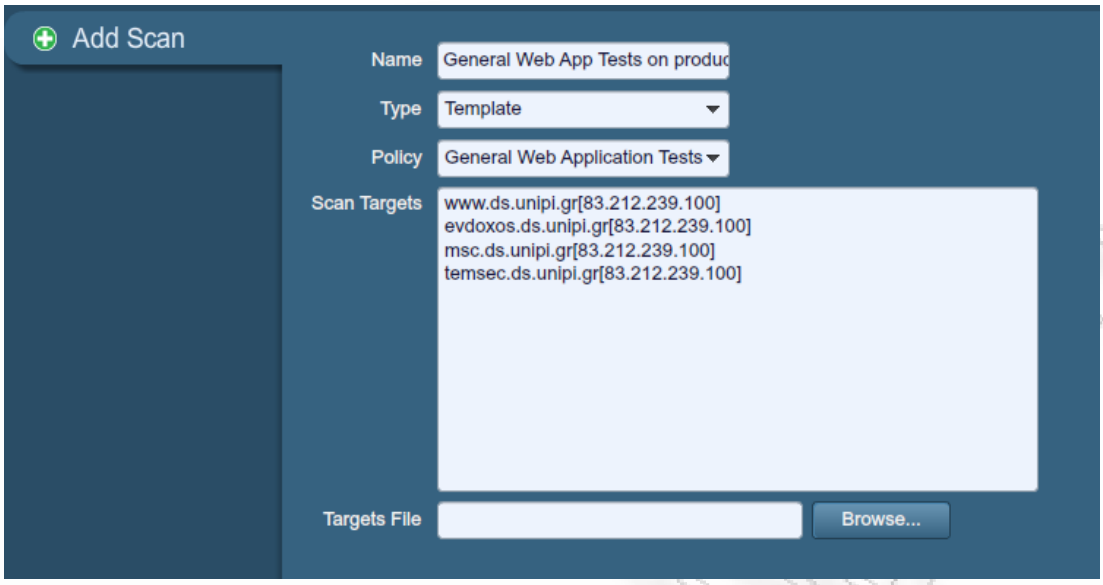
3.2.1.2. Παραμετροποίηση επιλογών καρτέλας Scan

Τέλος, αφού πραγματοποιηθεί η παραμετροποίηση της γενικευμένης πολιτικής σάρωσης πρέπει να ακολουθήσει η σύνδεση αυτής της πολιτικής με κάποιους/κάποιο στόχο προς σάρωση. Στην καρτέλα "Scan" δίνεται η δυνατότητα αυτή, και ακόμα καλύτερα η δημιουργία ενός πρότυπου σάρωσης το οποίο θα διευκολύνει τη γρήγορη εκκίνηση και παραμετροποίηση μιας σάρωσης με συγκεκριμένους στόχους. Μέσω της επιλογής "**Add**" εμφανίζεται η οθόνη παραμετροποίησης, στην οποία παρέχεται ένα χαρακτηριστικό όνομα που θα αφορά την σάρωση, για παράδειγμα "**General Web App Tests on production server**" και το οποίο κάνει γρήγορο και ξεκάθαρο το σκοπό κα το στόχο της συγκεκριμένης σάρωσης. Κατόπιν στην παράμετρο "**Type**" γίνεται επιλογή του "**template**" όπου θα αποθηκεύσει την συσχέτιση πολιτικής και στόχων για γρήγορη πρόσβαση αργότερα. Στην παράμετρο "**Policy**" γίνεται επιλογή της πολιτικής σάρωσης, στην προκειμένη η "**General Web App Test**" και στην παράμετρο "**Scan Targets**" προσδιορίζεται η λίστα των στόχων προς σάρωση. Συγκεκριμένα οι στόχοι είναι οι εξής:

- **www.ds.unipi.gr[83.212.239.100]**
- **evdoxos.ds.unipi.gr[83.212.239.100]**
- **msc.ds.unipi.gr[83.212.239.100]**
- **temsec.ds.unipi.gr[83.212.239.100]**

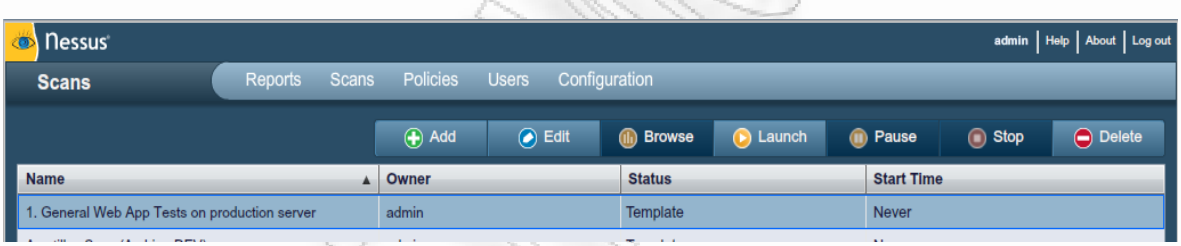
όπου αφορούν τις δικτυακές τοποθεσίες του τμήματος Ψηφιακών συστημάτων.

Σημειώνεται ότι η μορφή, όνομα τομέα σε συνδυασμό με την διεύθυνση IP (www.ds.unipi.gr[83.212.239.100]), είναι πολύ σημαντική εδώ, γιατί αυτός είναι ο μοναδικός τρόπος με τον οποίο δίνεται η δυνατότητα, στο Nessus, κατά τη σάρωση να στοχεύσει σε συγκεκριμένες εφαρμογές μέσω του ονόματος τομέα που διατηρούν (virtual hosts) και όχι μέσω της IP. Σε αντίθετη περίπτωση που δοθεί μόνο το όνομα τομέα, η σάρωση πιθανό δεν θα κατορθώσει να ανακαλύψει και να ελέγξει τη συγκεκριμένη εφαρμογή που αφορά το όνομα τομέα, αφού κατά τη διαδικασία της εκκίνησης, γίνεται η απόδοση του ονόματος σε IP διεύθυνση, με αποτέλεσμα, όταν στέλνονται τα δεδομένα των δοκιμών να καταλήγουν στην προεπιλεγμένη ιστοσελίδα-εφαρμογή που βρίσκεται πίσω από την IP διεύθυνση που ανακαλύφθηκε και που αρκετά συχνά μπορεί να είναι ακόμα και κενή.



Εικόνα 63: Εισαγωγή στόχων (Add Scan)

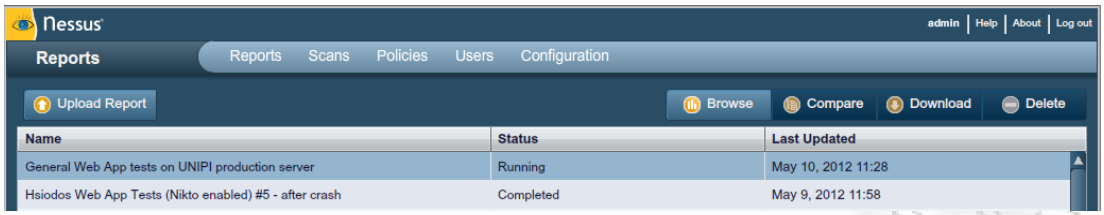
Η εκκίνηση του προτύπου σάρωσης εκκινείτε πολύ εύκολα μέσω της επιλογής "Launch" αφού έχει γίνει πρώτα επιλογή της νεοδημιουργηθείσας σάρωσης.



Εικόνα 64: Εκκίνηση σάρωσης

3.2.1.3. Ανασκόπηση αποτελεσμάτων από την καρτέλα Results

Η ανασκόπηση των αποτελεσμάτων μπορεί να πραγματοποιηθεί από την στιγμή που έχει εκκινήσει η σάρωση κάποιου στόχου. Φυσικό είναι να πρέπει να επέλθει η πάροδος κάποιου εύλογου χρονικού διαστήματος μέχρι να ξεκινήσουν να κάνουν την εμφάνιση τους τα πρώτα αποτελέσματα. Σε γενικές γραμμές οι σαρώσεις, ειδικά των εφαρμογών ιστού, είναι ιδιαίτερα χρονοβόρες, με χρόνους να κυμαίνονται από κάποιες ώρες μέχρι μερικές δεκάδες ώρες. Επίσης το φορτίο το οποίο παράγεται είναι διόλου ευκαταφρόνητο αφού γίνεται η ανασκόπηση, άρα και μεταφορά, σχεδόν ολόκληρου του περιεχομένου από κάθε δικτυακή εφαρμογή και δικτυακό ιστότοπο που σαρώνεται.

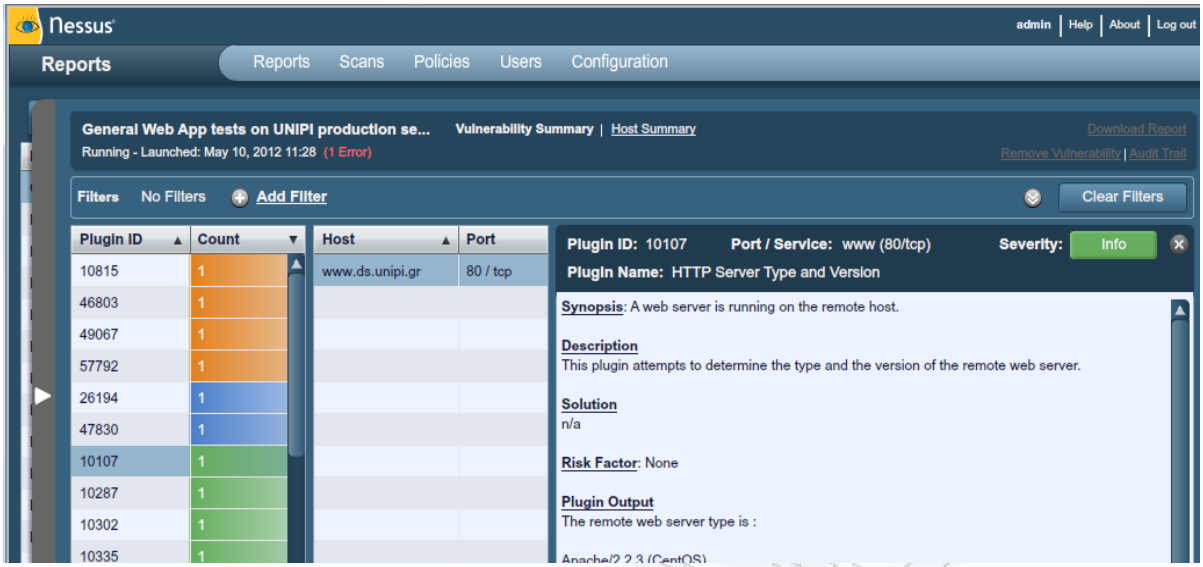


Εικόνα 65: Προβολή κατάστασης σάρωσης

Με την επιλογή της καρτέλας “Reports” εμφανίζονται σε λίστα, τόσο οι σαρώσεις όπου έχουν τελειώσει όσο και οι τρέχουσες ενεργές. Επιλέγοντας τη σάρωση που ενδιαφέρουν τα αποτελέσματα της και πατώντας την επιλογή “**Browse**” εμφανίζεται η οθόνη παρουσίασης των αποτελεσμάτων και δίνεται η δυνατότητα περιήγησης σε αυτά. Όταν μια σάρωση είναι ενεργή η οθόνη εμφάνισης των αποτελεσμάτων ανανεώνεται δυναμικά κατά τη διάρκεια της, παρουσιάζοντας κάθε καινούρια πληροφορία ή ευπάθεια που ανακαλύπτεται. Αρχικά η προβολή γίνεται σε μια κάθετη λίστα, που έχει ταξινομηθεί κατά plugin και που παρουσιάζει, τον αριθμό αναγνωριστικού του plugin (ID), τον αριθμό εμφάνισης του, τη σοβαρότητα της ευπάθειας, το όνομα του κάθε plugin και τέλος την οικογένεια στην οποία ανήκει.

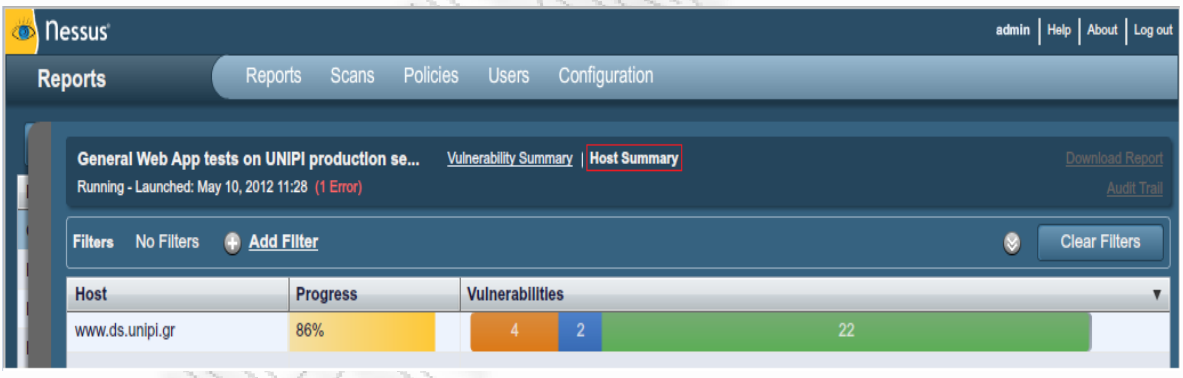
Εικόνα 66: Παρουσίαση αποτελεσμάτων με βάση τις ευπάθειες

Η επιλογή μιας από αυτές παρουσιάζει τις λεπτομέρειες που την περιγράφουν και τα σχετικά ευρήματα ως προς την εφαρμογή που σαρώθηκε σύμφωνα με τους ελέγχους όπου πραγματοποίησε.



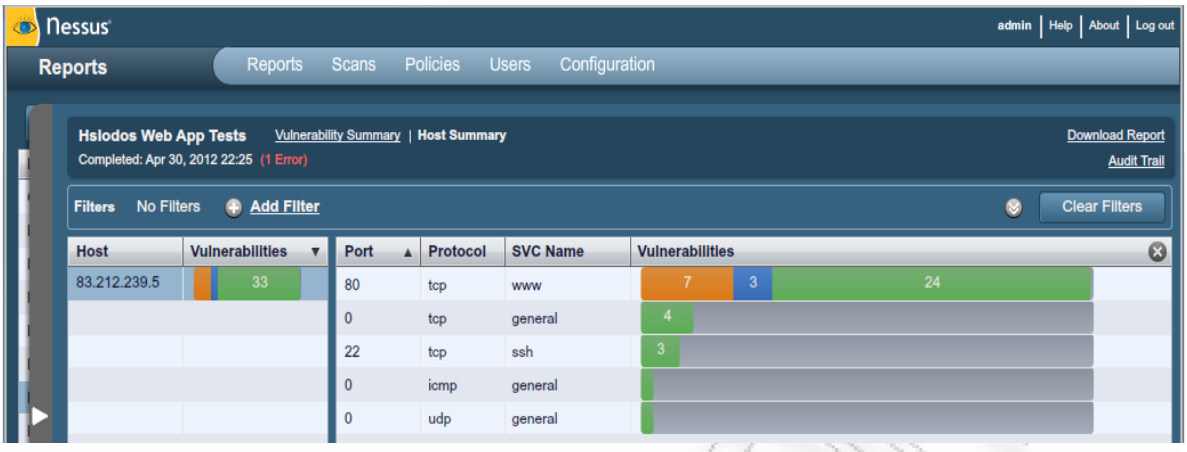
Εικόνα 67: Προβολή ευπαθειών με βάση το Host name

Η ταξινόμηση με βάση τους στόχους πραγματοποιείται πολύ εύκολα μέσω της επιλογής “**Host Summary**”. Χαρακτηριστικό της συγκεκριμένης ταξινόμησης είναι και η προβολή της μπάρας προόδου, στην περίπτωση που η σάρωση βρίσκεται σε εξέλιξη, με εμφάνιση του ποσοστού περάτωσης.



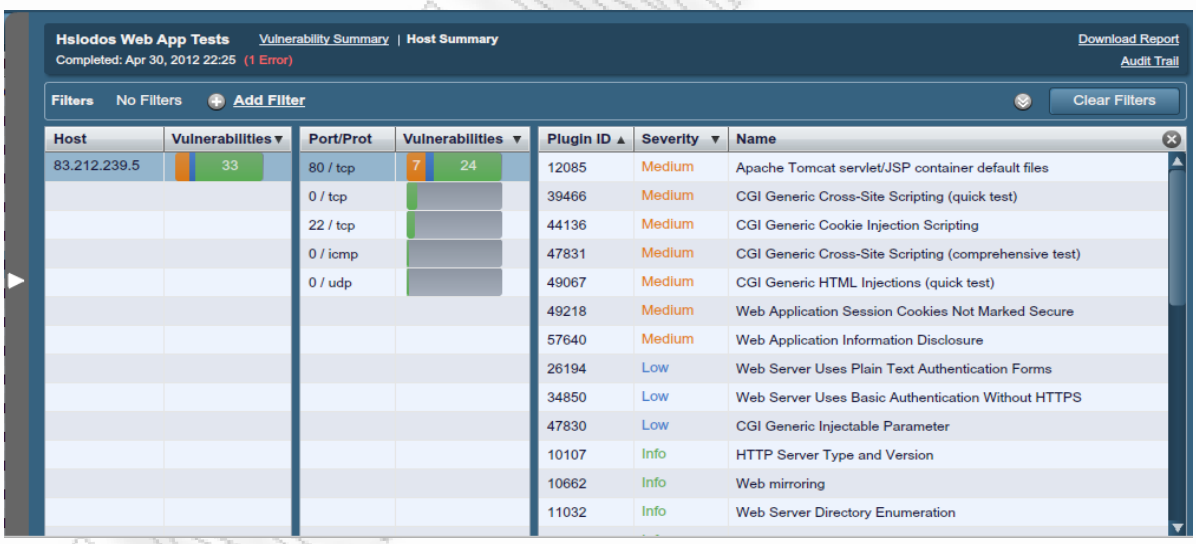
Εικόνα 68: Προβολή προόδου σάρωσης

Η επιλογή του στόχου όπου σαρώνεται εμφανίζει μια λίστα με τις πόρτες που έχουν εντοπιστεί ως ανοικτές και παρουσιάζεται συνοπτικά ο αριθμός των plugins που αφορούν την σοβαρότητα της ευπάθειας.



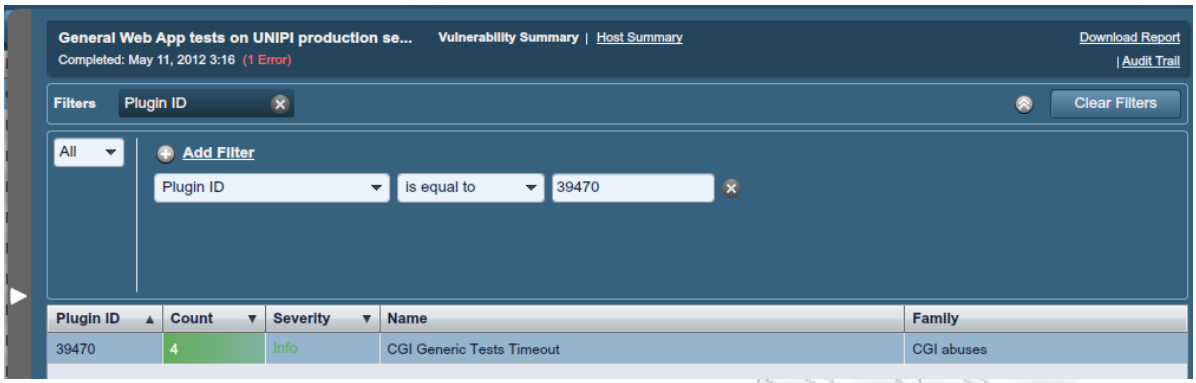
Εικόνα 69: Παρουσίαση ανοικτών Ports

Με την επιλογή μιας πόρτας ενεργοποιείται η περαιτέρω εξερεύνηση της, σε σχέση με τις ευπάθειες και τα plugins που "έτρεξαν" στην συγκεκριμένη πόρτα. Σε μια νέα οθόνη παρουσιάζονται σε κάθετη λίστα τα Plugins που εκτελέστηκαν, ο αριθμός αναγνωριστικού του plugin (ID), ο αριθμός εμφάνισης του, η σοβαρότητα της ευπάθειας και το όνομα του κάθε plugin. Η επιλογή ενός από τα plugins θα εμφανίσει αυτόματα όλες τις λεπτομέρειες που το αφορούν.



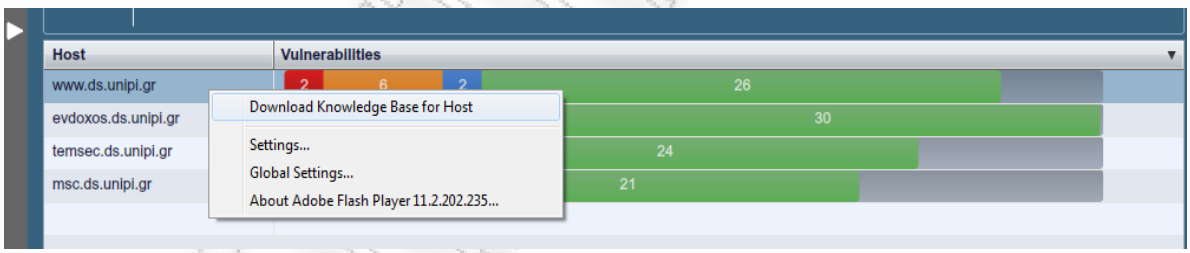
Εικόνα 70: Παρουσίαση ευπαθειών που αφορούν το port 80

Με τη χρήση των φίλτρων, μέσω της επιλογής "**Add Filter**", πραγματοποιείται η εξειδικευμένη εμφάνιση των αποτελεσμάτων. Για παράδειγμα την εμφάνιση των αποτελεσμάτων που αφορά μόνο κάποιο συγκεκριμένο plugin σύμφωνα με το όνομα του ή το ID του, όπως είναι το για παράδειγμα το plugin με αριθμό 39470, που παρουσιάζει λεπτομέρειες σχετικά με τις διακοπές εκτέλεσης των CGI Generic Tests.



Εικόνα 71: Φιλτράρισμα αποτελεσμάτων με βάση το Plugin ID

Μέσω της επιλογής "Download" μπορεί να πραγματοποιηθεί η μεταφόρτωση της τελικής αναφοράς από οποιοδήποτε σημείο της ανασκόπησης. Να σημειωθεί ότι η αναφορά θα μεταφορτωθεί σύμφωνα με το/α φίλτρα που είναι ενεργά τη συγκεκριμένη στιγμή. Σημαντικές πληροφορίες σχετικά με μια σάρωση μπορεί κανείς να βρει και από το αρχείο της σχετικής με κάποιο συγκεκριμένο στόχο, "γνωσιακής βάσης" που κρατάει το Nessus. Το συγκεκριμένο αρχείο είναι ένα απλό αρχείο κειμένου, που περιέχει τα plugins όπου εκκίνησαν και λεπτομέρειες σχετικά με το χρόνο εκτέλεσης τους ή μη εκτέλεσης τους, ενώ η μεταφόρτωση του γίνεται με την επιλογή "**Download Knowledge Base for Host**" κάνοντας δεξί κλικ στον στόχο.



Εικόνα 72: Επιλογή μεταφόρτωσης "Knowledge Base for Host"

3.2.2. Δημιουργία εξειδικευμένης πολιτικής σάρωσης εφαρμογών ιστού

Σκοπός μια εξειδικευμένης ή στοχευμένης πολιτικής σάρωσης είναι η χρήση της σε μια και μόνο συγκεκριμένη εφαρμογή, στην οποία απαιτείται προηγουμένως η επιτυχής αυθεντικοποίηση σε αυτή, για την σάρωση της και την ανακάλυψη των λειτουργιών της. Η εξειδικευμένη σάρωση δηλαδή, χρησιμοποιείται κυρίως για την εκτέλεση αυθεντικοποιημένων ελέγχων. Αυτός είναι και ο λόγος που καμία εξειδικευμένη πολιτική σάρωσης δεν μπορεί να χρησιμοποιηθεί σε περισσότερες από μια εφαρμογές ιστού, αφού είναι παραμετροποιημένη με τέτοιο τρόπο ώστε να χρησιμοποιεί τα στοιχεία της, για την εξασφάλιση πρόσβασης σε μια συγκεκριμένη εφαρμογή που επιθυμείται ο έλεγχος.

Προϋπόθεση της πολιτικής αυτής είναι η πρότερη γνώση ορισμένων παραμέτρων και στοιχείων για την επιτυχή σάρωση. Η πολιτική αυτή, βασίζεται στην γενική πολιτική, που αναφέρεται στο προηγούμενο υποκεφάλαιο.

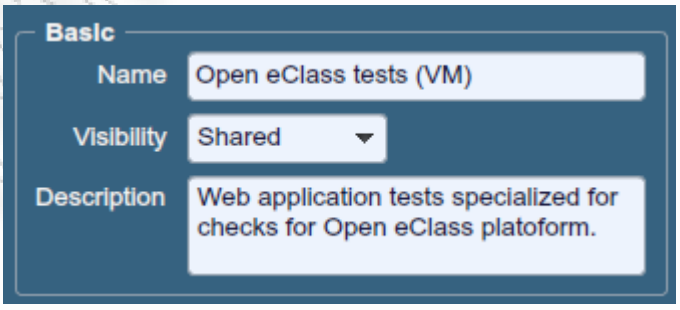
Για την δημιουργία αυτής της πολιτικής, θα χρησιμοποιηθεί μια πραγματική εφαρμογή η οποία βρίσκεται εγκατεστημένη, σε ελεγχόμενο περιβάλλον για τη διεξαγωγή των σαρώσεων. Η εφαρμογή αυτή, διανέμεται ελεύθερα ως Λογισμικό Ανοικτού Κώδικα σύμφωνα με τη γενική δημόσια άδεια GNU General Public License (GNU GPL). Συγκεκριμένα, ελέγχεται η πλατφόρμα **Open eClass** [18], ένα σύστημα διαχείρισης ηλεκτρονικών μαθημάτων για την ηλεκτρονική οργάνωση, αποθήκευση και παρουσίαση του εκπαιδευτικού υλικού. Αποτελεί πρόταση του Ακαδημαϊκού Διαδικτύου GUnet για την υποστήριξη των Υπηρεσιών Ασύγχρονης Τηλεκπαίδευσης.

Να σημειωθεί ότι, εξαιρετικά σημαντικό ρόλο, στην δημιουργία και πραγματοποίηση μιας στοχευμένης πολιτικής, κατέχει η πρότερη χειροκίνητη συλλογή πληροφοριών για την αποτελεσματική κατάρτιση της και η οποία θα περιγράφεται καθ' όλη τη διάρκεια, στη συνέχεια του κεφαλαίου.

3.2.2.1. Παραμετροποίηση επιλογών καρτέλας Policies

Μέσω της καρτέλας Policies γίνεται επιλογή της προϋπάρχουσας πολιτικής "General Web Application tests", που δημιουργήθηκε νωρίτερα κατά την κατάρτιση της γενικής πολιτικής σάρωσης και επιλέγεται η αντιγραφή της με αποτέλεσμα τη δημιουργία μιας νέας πολιτικής με όνομα "Copy of General Web Application tests".

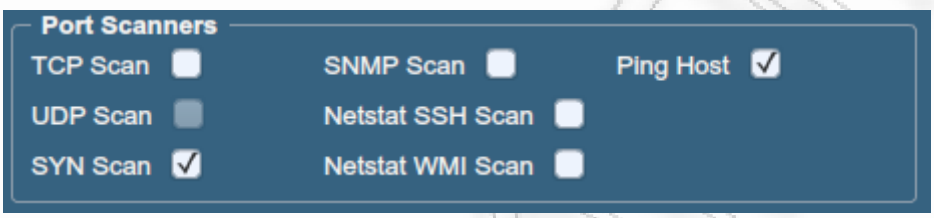
Κατόπιν ξεκινάει η παραμετροποίηση της καινούριας πολιτικής μέσω της επιλογής "**Edit**". Στο τμήμα "**Basic**" δίνεται ένα καινούριο όνομα σε αυτή το οποίο θα υποδηλώνει το στόχο της ώστε με μια ματιά να γίνεται κατανοητό τι αντιπροσωπεύει, μαζί με μια σύντομη περιγραφή που ξεκαθαρίζει το σκοπό της συγκεκριμένης πολιτικής.



Εικόνα 73: Επιλογές Edit Policy/General/Basic

[18] Open eClass, http://www.openeclass.org/

Στο τμήμα "**Port Scanners**" που αντιπροσωπεύει τη σάρωση για "ανοικτές" πόρτες στον απομακρυσμένο στόχο παραμένουν οι επιλογές "**Ping Host**" και "**SYN Scan**" ώστε να δοθεί η δυνατότητα στον σαρωτή να κρατάει δεδομένα και στατιστικά σχετικά με τη σύνδεση στον απομακρυσμένο στόχο. Επίσης η επιλογή του SYN Scan επιταχύνει την εμφάνιση αποτελεσμάτων σε μια ενεργή σάρωση. Συνήθως στην στοχευμένη πολιτική, δεν έχει πολύ νόημα να παραμείνουν ενεργά όλα τα port scanners αφού η μόνη πόρτα και υπηρεσία που ενδιαφέρει είναι αυτή του εξυπηρετητή δικτύου. Παρόλα αυτά μπορεί να κατευθυνθεί η σάρωση σε αυτή/ες τις πόρτες που έχουν μεγαλύτερο ενδιαφέρον.

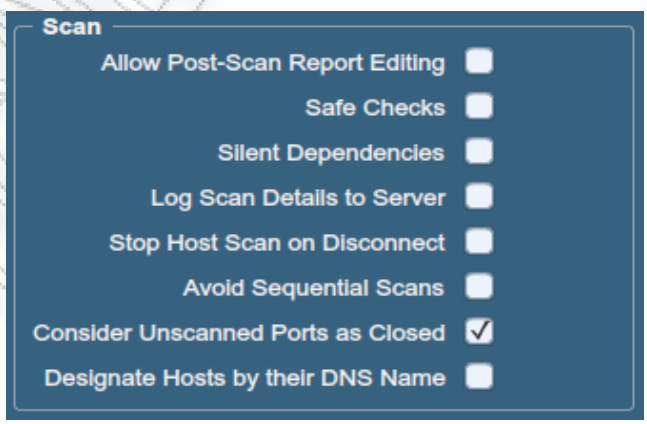


Εικόνα 74: Επιλογές Edit Policy/General/Port Scanners

Για τον ίδιο λόγο που αναφέρθηκε στην προηγούμενη παράγραφο, στο τμήμα "Port Scan Options" παραμένει ως μόνη επιλογή προς σάρωση, η πόρτα 80.

Εικόνα 75: Επιλογές Edit Policy/General/Port Scan Options

Στο τμήμα "**Scan**" παραμένει ως μόνη επιλογή η "**Consider Unscanned Ports as Closed**" ώστε να αποφευχθεί η άσκοπη σάρωση για ανακάλυψη υπηρεσιών πίσω από πόρτες που δεν έχουν κάποιο ενδιαφέρον στη σάρωση και στην επιτάχυνση της όλης διαδικασίας.



Εικόνα 76: Επιλογές Edit Policy/General/Scan

Η επιλογή των **plugins** έχει ήδη πραγματοποιηθεί κατά τη δημιουργία της γενικής πολιτικής σάρωση και άρα δεν χρειάζεται καμία παρέμβαση στην συγκεκριμένη καρτέλα των plugins.

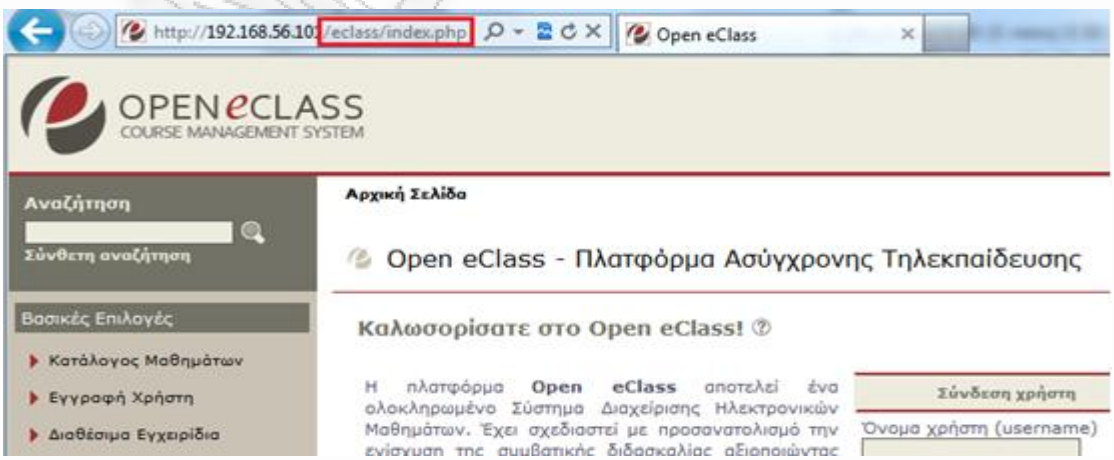
Η καρτέλα **Preferences**, οδηγεί στη ρύθμιση των προηγμένων επιλογών που αφορούν μια στοχευμένη πολιτική σάρωσης. Οι ρυθμίσεις που χρειάζονται παραμετροποίηση ή/και επισκόπηση, βρίσκονται στις ακόλουθες καρτέλες από το μενού αναδίπλωσης:

1. Global variable settings
2. HTTP login page
3. Login configuration
4. Web Application Tests Settings
5. Web mirroring

Στην καρτέλα Global variable settings θα πρέπει απλά να βεβαιωθούμε ότι διατηρούνται οι προηγούμενες ρυθμίσεις αφού δεν είναι απαραίτητη καμία παρέμβαση για νέες ρυθμίσεις.

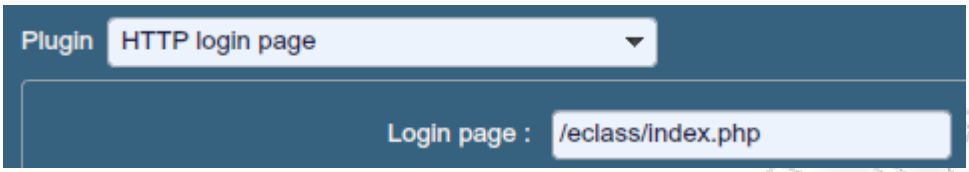
Οι ρυθμίσεις στην καρτέλα HTTP login page είναι πολύ σημαντικές ώστε να εξασφαλιστεί η επιτυχής είσοδος στην εφαρμογή υπό εξέταση. Κατά τη διάρκεια της ρύθμισης των επιλογών της καρτέλας αυτής σίγουρα θα χρειαστεί η περαιτέρω μελέτη της εφαρμογής, για την συλλογή των κατάλληλων πληροφοριών που αφορούν τις συγκεκριμένες ρυθμίσεις ή και ακόμα η συμβολή εργαλείων, όπως είναι για παράδειγμα κάποιο εργαλείο επίβλεψης πακέτων δικτύου ή κάποιος διαμεσολαβητής συνεδριών.

Αρχικά γίνεται πλοήγηση στην οθόνη αυθεντικοποίησης της εφαρμογής ώστε να παρουσιαστεί το πραγματικό URI που οδηγεί σε αυτή την οθόνη.



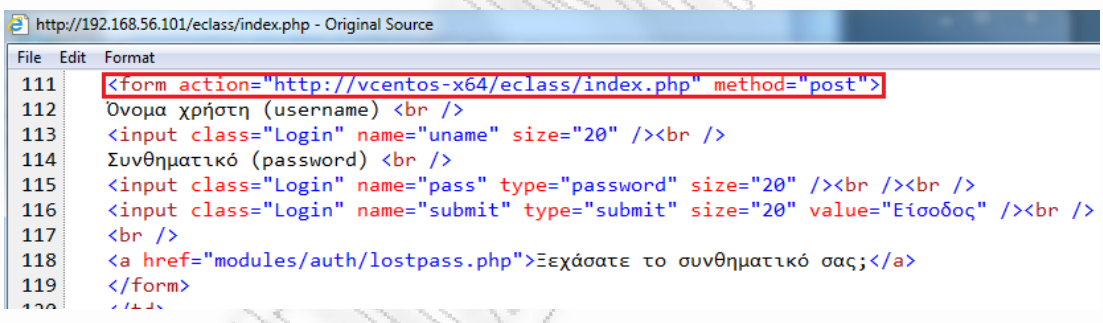
Εικόνα 77: Αρχική σελίδα εφαρμογής Open eClass

Και το οποίο είναι η παράμετρος που εισάγεται στο πεδίο "**Login page**".



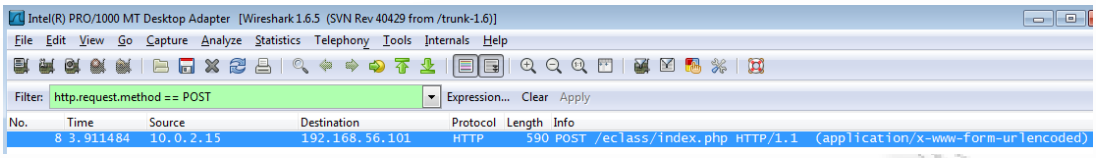
Εικόνα 78: HTTP Login page μέρος 1

Ακολούθως στο πεδίο "**Login form"** εισάγεται το πεδίο "action" που βρίσκεται μέσα στη μέθοδο **<form>** της HTML. Αυτός είναι στην ουσία ο προορισμός που θα μεταφερθεί ο χρήστης μετά την αυθεντικοποίηση. Για την ανακάλυψη της συγκεκριμένης παραμέτρου πρέπει να γίνει αναδρομή στον κώδικα της εφαρμογής μέσω του φυλλομετρητή ιστού. Συγκεκριμένα στον κώδικα που παρουσιάζεται στην παρακάτω εικόνα το URI που θα χρησιμοποιηθεί είναι το "**/eclass/index.php**" χωρίς το "http://vcentos-x64" αφού η συγκεκριμένη συμβολοσειρά αποτελεί το στόχο, όπου επικολλάτε αυτόματα από το εργαλείο κατά τη σάρωση και ο οποίος εισάγεται κατά τη δημιουργία της σάρωσης, μέσω της καρτέλας "Scans".

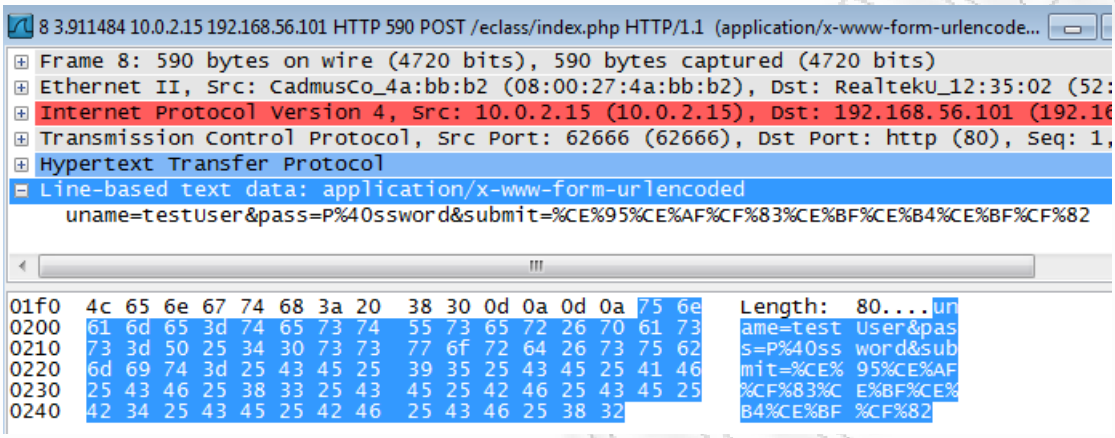


Εικόνα 79: Κώδικας σελίδας αυθεντικοποίησης

Επίσης στο συγκεκριμένο παράδειγμα από το ίδιο κομμάτι κώδικα παρουσιάζονται τόσο οι παράμετροι αυθεντικοποίησης "unane" και "pass" που χρησιμοποιούνται, όσο και ο τύπος της μεθόδου αυθεντικοποίησης, που δεν είναι άλλος από την "POST". Για την ορθή και πιο σίγουρη παραμετροποίηση του πεδίου "**Login form fields**" γίνεται η χρήση του εργαλείου Wireshark ώστε να είναι δυνατή η παρατήρηση της ακριβής συμβολοσειράς που αποστέλλεται στον διακομιστή για την πραγματοποίηση της αυθεντικοποίησης. Αφού πραγματοποιηθεί η καταγραφή της κίνησης μεταξύ του web browser και του διακομιστή, τότε γίνεται φιλτράρισμα των αποτελεσμάτων με τη χρήση της συμβολοσειράς "*http.request.method == POST*" ώστε να παρουσιαστούν όσα πακέτα έκαναν χρήση της μεθόδου POST. Μέσα στα bytes μεταφοράς αυτού του πακέτου φαίνεται ξεκάθαρα η συμβολοσειρά που χρησιμοποιήθηκε για την πραγματοποίηση της αυθεντικοποίησης στην εφαρμογή, η οποία είναι η



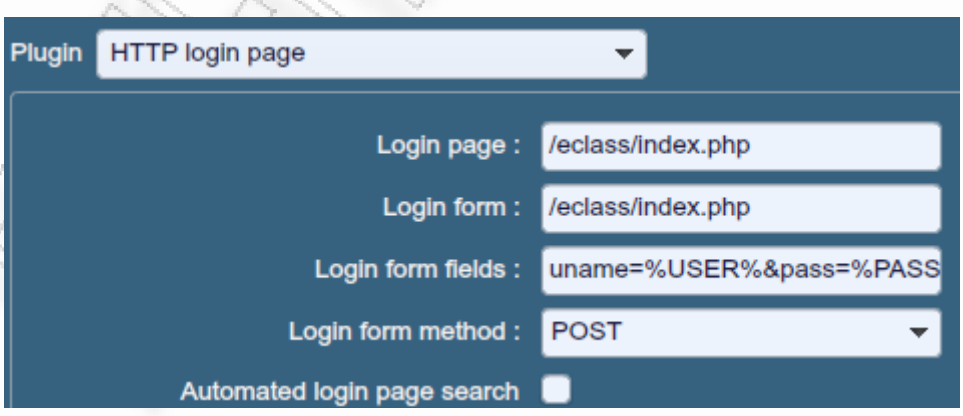
Εικόνα 80: Φίλτρο εύρεσης πακέτου αυθεντικοποίησης στο Wireshark



Εικόνα 81: Δικτυακό πακέτο από Wireshark

"**uname=testUser&pass=P%40ssword&submit=%CE%95%CE%AF%CF%83%CE%BF%CE%B4%CE%BF%CF%82**"

Στο πεδίο "**Login form fields**" θα εισαχθεί η ακόλουθη συμβολοσειρά "**uname=%USER%&pass=%PASS%&submit=%CE%95%CE%AF%CF%83%CE%BF%CE%B4%CE%BF%CF%82**" στην οποία αντικαθίστανται το πραγματικό username και password με τις παραμέτρους %USER% και %PASS%, οι οποίες θα αντικατασταθούν με τη σειρά τους αυτόματα κατά τη διάρκεια της σάρωσης από τις παραμέτρους που θα εισαχθούν στην καρτέλα "**Login configurations**" αμέσως μετά.



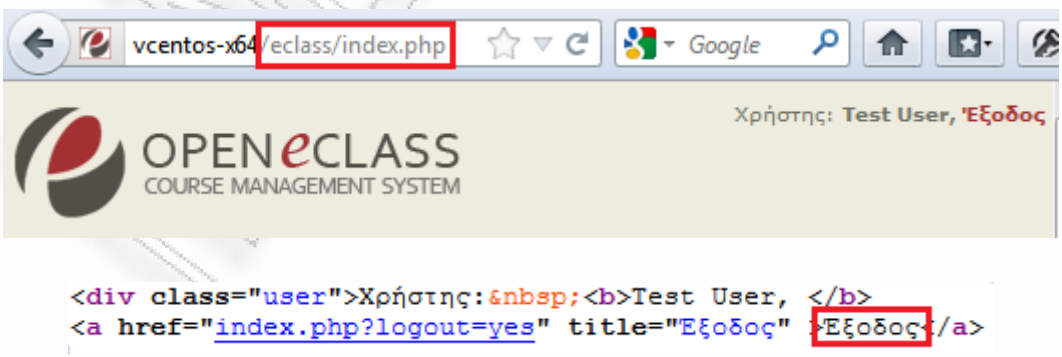
Εικόνα 82: HTTP Login page μέρος 2

Αφού πλέον υπάρχουν όλα τα απαραίτητα πεδία για την επιτυχή αυθεντικοποίηση η επιλογή "Automated login page search" δεν είναι απαραίτητη και άρα δεν χρειάζεται να επιλεγεί.

Το πεδίο "**Re-authenticate delay (seconds)**" ορίζει το χρόνο αναμονής που θα έχει το Nessus για την επανάληψη της διαδικασίας αυθεντικοποίησης, σε περίπτωση που για οποιονδήποτε λόγο, γίνει αποσύνδεση του Nessus κατά τη διάρκεια της σάρωσης.

Στο πεδίο "**Check authentications on page**" ορίζεται το URI μονοπάτι της σελίδας, μέσω της οποία το Nessus, ανά τακτά χρονικά διαστήματα ελέγχει κατά πόσο εξακολουθεί να παραμένει αυθεντικοποιημένο στην εφαρμογή και αν διαπιστωθεί το αντίθετο τότε επαναλαμβάνεται η διαδικασία της αυθεντικοποίησης, λαμβάνοντας υπόψη πάντοτε το προηγούμενο πεδίο του "Re-authenticate delay (seconds)". Για τον ορισμό φυσικά του συγκεκριμένου URI θα πρέπει να γίνει χειροκίνητα η αυθεντικοποίηση από τον ελεγκτή και μετά την επιτυχή είσοδο στην εφαρμογή να χρησιμοποιηθούν οι πληροφορίες που παρέχονται στο URL του φυλλομετρητή.

Στην παράμετρο "**Authentication regex**" εισάγεται η συμβολοσειρά μέσω της οποία το Nessus θα μπορεί να διαπιστώσει κατά τη διάρκεια της σάρωσης κατά πόσο η αυθεντικοποίηση στην εφαρμογή πραγματοποιήθηκε με επιτυχία ή όχι. Συνήθως μια λέξη η οποία συναντάται μόνο στη περίπτωση που κάποιος είναι αυθεντικοποιημένος στην εφαρμογή όπως για παράδειγμα οι "**logout**" ή "**Έξοδος**". Συγκεκριμένα, η βέλτιστη πρακτική είναι η ανασκόπηση του κώδικα στην σελίδα που παρουσιάζεται μετά την επιτυχή είσοδο στην εφαρμογή. Άρα στο προηγούμενο βήμα που έγινε ο έλεγχος για τη σελίδα μετά την αυθεντικοποίηση πραγματοποιείται και η ανασκόπηση του κώδικα για την λέξη "κλειδί" όπως παρουσιάζεται στην ακόλουθη εικόνα.

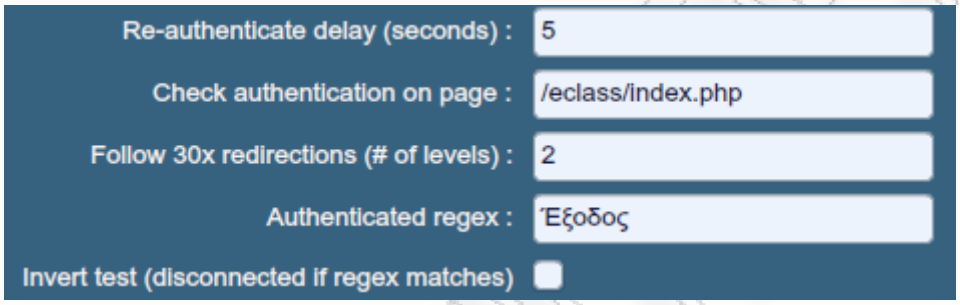


Εικόνα 83: Σελίδα μετά από επιτυχή αυθεντικοποίηση και κώδικας αυτής

Το πεδίο "**Follow 30x redirections (# of levels)**" ορίζει το πλήθος των ανακατευθύνσεων, τύπου 30x που ίσως συναντήσει στην εφαρμογή κατά τη διάρκεια των

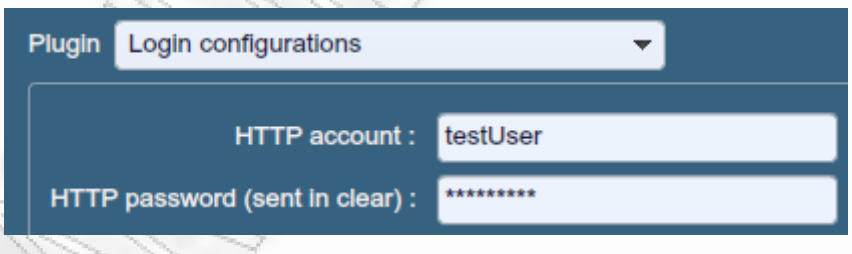
ελέγχων το Nessus και πόσες από αυτές θα ακολουθήσει ώστε να μην πέσει σε μεγάλο αριθμό ανακατευθύνσεων και τελικά χάσει το στόχο του. Ο αριθμός δύο (2) που είναι και η προεπιλογή δεν χρειάζεται να αλλαχτεί.

Μέσω της επιλογής "**Invert test (disconnected if regex matches)**" αντιστρέφει τη λειτουργία της προηγούμενης παραμέτρου που ως αποτέλεσμα έχει να κάνει το Nessus να αντιλαμβάνεται ότι με την παρουσία της συγκεκριμένης συμβολοσειράς σημαίνει την αποτυχία της αυθεντικοποίησης ή της αποσύνδεσης από την εφαρμογή.



Εικόνα 84: HTTP Login page μέρος 3

Στην καρτέλα Login configuration θα γίνει παραμετροποίηση των πεδίων "HTTP account" στο οποίο εισάγεται το όνομα χρήστη με το οποίο θα πραγματοποιηθεί η αυθεντικοποίηση στην εφαρμογή υπό εξέταση και το πεδίο "HTTP password (sent in clear)" εισάγεται το συνθηματικό. Αυτά τα πεδία αντικαθιστούν τις προαναφερθείσες παραμέτρους **%USER%** και **%PASS%** αντιστοίχως όπου εισήχθησαν στο πεδίο "Login form fields" νωρίτερα.



Εικόνα 85: Login configuration

Στην καρτέλα Web Application Test Settings ορίζονται οι έλεγχοι όπου θα πραγματοποιηθούν στην εφαρμογή προς εξέταση. Η επιλογή "**Enable web applications tests**" εξασφαλίζει την εκτέλεση των απαραίτητων plugins για την ανακάλυψη αδυναμιών σχετικά με τις εφαρμογές ιστού. Αυτό επιτυγχάνεται μέσω μιας ενσωματωμένης λειτουργίας του Nessus, η οποία κάνει χρήση της τεχνικής black box δοκιμής διείσδυσης, εισάγοντας στην εφαρμογή τυχαία δεδομένα εισόδου στα entry points που ανακαλύπτει, μέσω της διαδικασία mirroring που περιγράφεται παρακάτω και παρακολουθώντας την

έξοδο που παράγει αυτό, για τυχόν παράξενη συμπεριφορά ή αποτυχίας απόκρισης της εφαρμογής. Η λειτουργία αυτή ονομάζεται **fuzzing** [19].

Η επιλογή "**Maximum run time (min)**" καθορίζει το χρόνο εκτέλεσης του κάθε CGI plugin και όχι το συνολικό χρόνο της σάρωσης. Αυτό πρακτικά σημαίνει ότι, για παράδειγμα αν το plugin που είναι υπεύθυνο για την ανακάλυψη ευπαθειών SQL injection εκκινήσει την εκτέλεση του, αυτό θα συμβαίνει για τα επόμενα 120 λεπτά, εφαρμόζοντας τους αλγοριθμικούς ελέγχους του, στις παραμέτρους όπου ανακαλύφθηκαν από τη διαδικασία του web mirroring, προτού διακοπεί η λειτουργία του, εκτός και αν προλάβει να τελειώσει η εκτέλεση του πριν από τον προκαθορισμένο χρόνο. Τα 120 λεπτά είναι αρκετός χρόνος σάρωσης για μια μεσαίου μεγέθους εφαρμογή για την πραγματοποίηση όλων των ελέγχων. Αργότερα θα γίνει αναφορά στον τρόπο πληροφόρησης, σχετικά με το ποια plugins δεν πρόλαβαν να εκτελεστούν στον προκαθορισμένο χρόνο και διακόπηκε η λειτουργία τους.

Η επιλογή "**Try all HTTP methods**" εξασφαλίζει στην σάρωση τη χρήση τόσο της μεθόδου GET, που αποτελεί την προεπιλογή, στην αποστολή των ελέγχων όσο και της μεθόδου POST. Στην επιλογή "**Combinations of arguments values**" καθορίζεται η διαχείριση από το Nessus των μεταβλητών που αποστέλλονται στην εφαρμογή για την πραγματοποίηση των ελέγχων. Η τιμή "**some combinations**" αποτελεί μια μέση κατάσταση των επιλογών "all pairs" και "all combinations". Να σημειωθεί ότι η συγκεκριμένη επιλογή σε συνδυασμό με την επιλογή της "**Stop at first flaw**" είναι αυτές οι οποίες έχουν το μεγαλύτερο αντίκτυπο στην διάρκεια πραγματοποίησης των ελέγχων και το κατά πόσο θα φτάσει ο χρόνος που ορίστηκε προηγουμένως για την εκτέλεση αυτών.

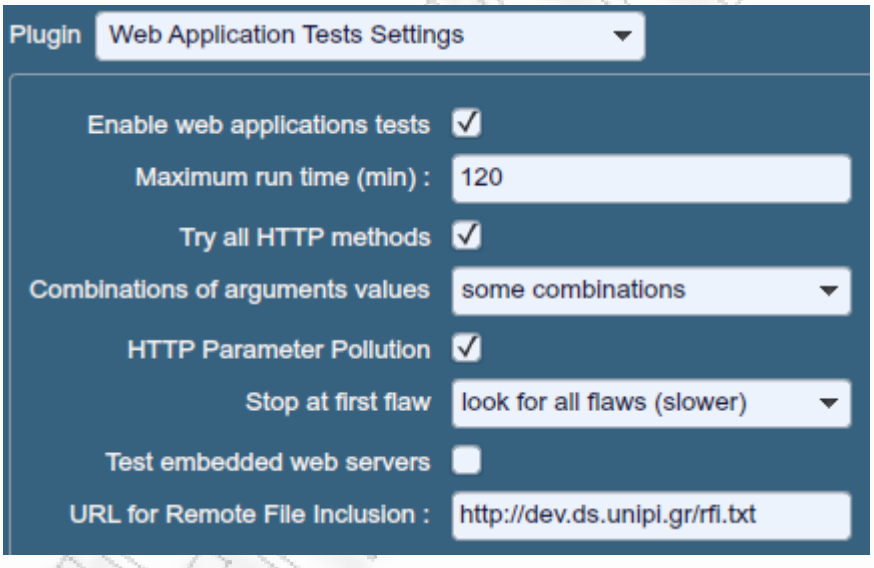
Με την επιλογή "**HTTP Parameter Pollution**" πραγματοποιείται η εισαγωγή της κανονικής μεταβλητής παράλληλα με την μεταβλητή ελέγχου, αυξάνοντας έτσι τους συνδυασμούς των παραμέτρων ελέγχου.

Όπως ήδη αναφέρθηκε η παράμετρος "**Stop at first flaw**" είναι υπεύθυνη για τον καθορισμό της συνθήκης τερματισμού κάποιου plugin ελέγχου. Ορίζοντας την τιμή στο "**look for all flaws (slower)**" εξασφαλίζεται ο πλήρης έλεγχος όλων των μεταβλητών που ανακαλύφθηκαν μέσω της διαδικασίας του web mirroring και αυτός ακριβώς είναι και ο λόγος της μεγάλης χρονικής καθυστέρησης της εκτέλεσης των ελέγχων και της αύξησης της πιθανότητας αυτοί να τερματίσουν λόγω χρόνου πριν ολοκληρώσουν το έργο τους.

[19] Fuzzing, https://www.owasp.org/index.php/Fuzzing

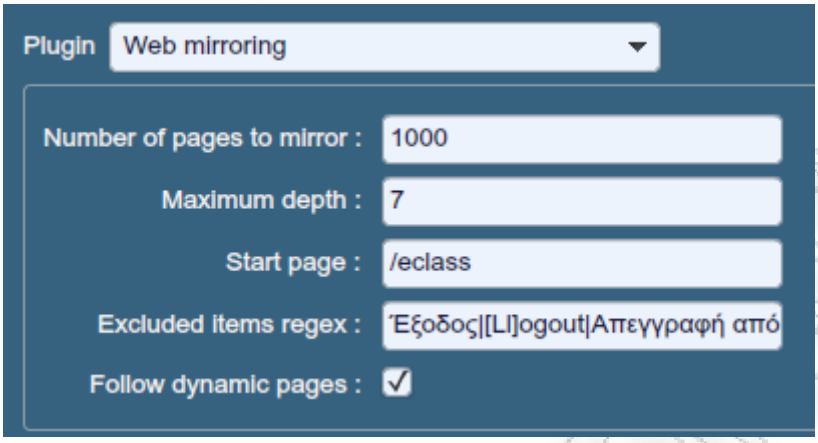
Η επιλογή "**Test embedded web servers**" πραγματοποιεί σάρωση σε ενσωματωμένους Web Servers που "τρέχουν" στους απομακρυσμένους στόχους και για αυτό το λόγο, αφού ο έλεγχος πραγματοποιείται στοχευμένα σε συγκεκριμένη εφαρμογή στον απομακρυσμένο στόχο, δεν έχει νόημα η επιλογή της. Να σημειωθεί ότι αν επιθυμείται ο έλεγχος των ενσωματωμένων web servers, η βέλτιστη πρακτική είναι η δημιουργία μιας στοχευμένης πολιτικής με μόνο στόχο και επιλογή αυτούς.

Στην επιλογή "**URL for Remote File Inclusion**" παρέχεται η διεύθυνση η οποία περιέχει ένα ειδικά διαμορφωμένο αρχείο που πραγματοποιεί τους ελέγχους για την απομακρυσμένη ένταξη αρχείων. Η προεπιλογή είναι το URL που περιέχει αυτό το αρχείο και φιλοξενείται στους εξυπηρετητές της Tenable. Μπορεί όμως να γίνει η παροχή αυτού του αρχείου και μέσω ενός εξυπηρετητή που βρίσκεται στο ίσιο δίκτυο με αυτό της εφαρμογής που ελέγχεται.



Εικόνα 86: Web application tests settings

Η καρτέλα Web Mirroring προσφέρει τη σημαντικότερη λειτουργία του Nessus, κατά την οποία πραγματοποιείται η ανακάλυψη των σελίδων, λειτουργιών και παραμέτρων που αποτελούν την υπό εξέταση εφαρμογή. Αυτό εξασφαλίζει την ανακάλυψη των άγνωστων ευπαθειών που ίσως περιέχει η εφαρμογή προς εξέταση, οι λεγόμενες zero day vulnerabilities, και που είναι ο μοναδικός τρόπος εξέτασης των προσαρμοσμένων και κατά παραγγελία εφαρμογών που αναπτύσσονται εσωτερικά σε ένα οργανισμό.



Εικόνα 87: Web mirroring (crawl)

Στη μεταβλητή "**Number of pages to mirror**" εισάγεται ο μέγιστος αριθμός σελίδων όπου επιτρέπεται να ανακαλυφθούν. Η προεπιλογή είναι 1000 σελίδες και η οποία σε φυσιολογικές συνθήκες δεν πρόκειται να προσεγγιστεί.

Με τη μεταβλητή "**Maximum depth**" ορίζεται το μέγιστο "βάθος" όπου επιτρέπει στη μέθοδο ανακάλυψης να φτάσει ακολουθώντας τους συνδέσμους της εφαρμογής για κάθε αρχική σελίδα που ανακαλύπτεται.

Στη μεταβλητή "**Start page**" ορίζεται το URI σο οποίο συνήθως αντιπροσωπεύει το φάκελο στον οποίο βρίσκεται μέσα η εφαρμογή προς σάρωση. Στη συγκεκριμένη περίπτωση η εφαρμογή βρίσκεται στο φάκελο "**/eclass**" από τον οποίο και θα ξεκινήσει η διαδικασία της σάρωσης.

Σημαντική μεταβλητή είναι η "**Excluded items regex**" μέσω της οποία δηλώνονται οι συμβολοσειρές στις οποίες όταν το Nessus συναντήσει κατά τη διάρκεια της μεθόδου ανακάλυψης συνδέσμων αποφεύγει την επιλογή τους γιατί μπορεί να έχουν ως αποτέλεσμα είτε την έξοδο από την εφαρμογή, για παράδειγμα με την επιλογή του "logout" είτε την πρόκληση κάποιας "ζημιάς" μη εσκεμμένα. Η συμπλήρωση φυσικά της συγκεκριμένης παραμέτρου προϋποθέτει την χειροκίνητη περιήγηση στην εφαρμογή και την συλλογή των απαραίτητων πληροφοριών από τον ελεγκτή. Για παράδειγμα στην συγκεκριμένη περίπτωση έχουν τεθεί ως συμβολοσειρές προς αποφυγή οι λέξεις "**Έξοδος|[Ll]ogout|Απεγγραφή από μάθημα|Αποστολή μηνύματος|English|Español**". Κάθε λέξη-συμβολοσειρά πρέπει να διαχωρίζεται από το σύμβολο "|" και να αποφεύγονται οι ειδικοί χαρακτήρες σύμφωνα με τους κανόνες συγγραφής των regular expressions [20] Οι λέξεις "Έξοδος" και "logout" ορίζονται προφανών για την αποφυγή επιλογής τους που θα

[20] RegEx, http://en.wikipedia.org/wiki/Regular_expression

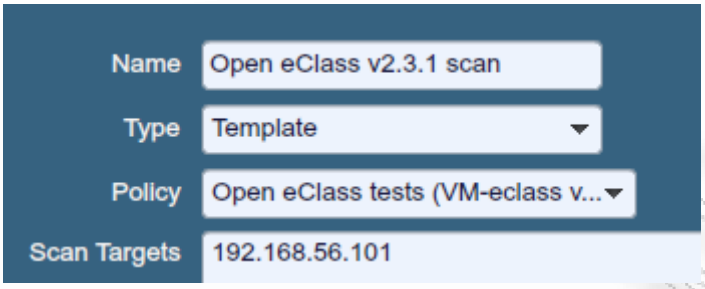
έχει ως αποτέλεσμα ην έξοδο από την εφαρμογή κατά τη διάρκεια της διαδικασίας ανακάλυψης των λειτουργιών και σελίδων που αποτελούν την εφαρμογή. Με την αποφυγή της επιλογής των λέξεων "Απεγγραφή από το μάθημα" εξασφαλίζεται η μη διαγραφή του χρήστη αφού η εφαρμογή απαγορεύει την διαγραφή του σε περίπτωση που αυτός είναι εγγεγραμμένος έστω και σε ένα μάθημα. Οι λέξεις "Αποστολή μηνύματος" εξασφαλίζουν τη μη αποστολή μηνυμάτων σε όσους είναι εγγεγραμμένοι στο μάθημα ή στον εκπαιδευτή του μαθήματος. Με την χρήση των λέξεων "English" και "Español" αποφεύγεται η αλλαγή της γλώσσας στην πλατφόρμα και άρα η αλλαγή της λέξης Έξοδος που αν επιλεγεί έχει ως αποτέλεσμα την αποσύνδεση από την εφαρμογή.

3.2.2.2. Παραμετροποίηση επιλογών καρτέλας Scan

Για την εκτέλεση της σάρωσης προς το στόχο, θα πρέπει να δημιουργηθεί ένα πρότυπο σάρωσης μέσω της καρτέλας "**Scan**" του Nessus. Επιλέγοντας το "**Add**" στην καρτέλα Scan δημιουργείται μια καινούρια σάρωση όπου θα σωθεί ως πρότυπο. Δίνεται το όνομα "**Open eClass v2.3.1 scan**" από το οποίο στο τέλος της σάρωσης και μέσω της επιλογής "Reports" θα εμφανιστούν τα αποτελέσματα. Στο πεδίο "**Type**" γίνεται επιλογή του "**Template**" ώστε να αποθηκευτεί ως πρότυπο η συγκεκριμένη σάρωση. Κατόπιν γίνεται η επιλογή της πολιτικής που δημιουργήθηκε προηγουμένως και τέλος στη μεταβλητή, "**Scan Targets**" εισάγεται ο στόχος με την μορφή της IP διεύθυνσης.

Να σημειωθεί ότι στο πεδίο "**Scan Targets**" και κυρίως όταν η σάρωση αφορά δικτυακές εφαρμογές, η βέλτιστη πρακτική εισαγωγής του στόχου, είναι υπό τη μορφή του συνδυασμού ονόματος και IP διεύθυνσης, όπως για παράδειγμα **www.example.com[192.168.1.1]**. Με τον τρόπο αυτό, εξασφαλίζεται η ορθή λειτουργία της διαδικασίας ανακάλυψης συνδέσμων που περιέχονται στην εφαρμογή (crawl), αφού αποκλείεται έτσι η λανθασμένη λήψη λαθών τύπου 404, για την μη ύπαρξη συνδέσμου. Το σφάλμα αυτό μπορεί να οφείλεται, είτε σε μόνιμους συνδέσμους (permalinks)[21] στην εφαρμογή που κάνουν χρήση του ονόματος τομέα, είτε σε συνδέσμους που κάνουν χρήση κάποιου DNS ονόματος το οποίο το Nessus δεν μπορεί να συσχετίσει με την IP διεύθυνση εξετάζει. Αυτό έχει ως αποτέλεσμα τη παράληψη ελέγχου σε κάποιους συνδέσμων, αφού κατά τη διαδικασία ανακάλυψης δεν είχε τη δυνατότητα να τους καταγράψει.

[21] Permalink, http://en.wikipedia.org/wiki/Permalink



Εικόνα 88: Στόχοι σάρωσης

Τέλος, αφού πραγματοποιηθεί η αποθήκευση του προτύπου, γίνεται η εκκίνηση του μέσω της επιλογής “Launch” και έτσι ενεργοποιείται η σάρωση του στόχου.

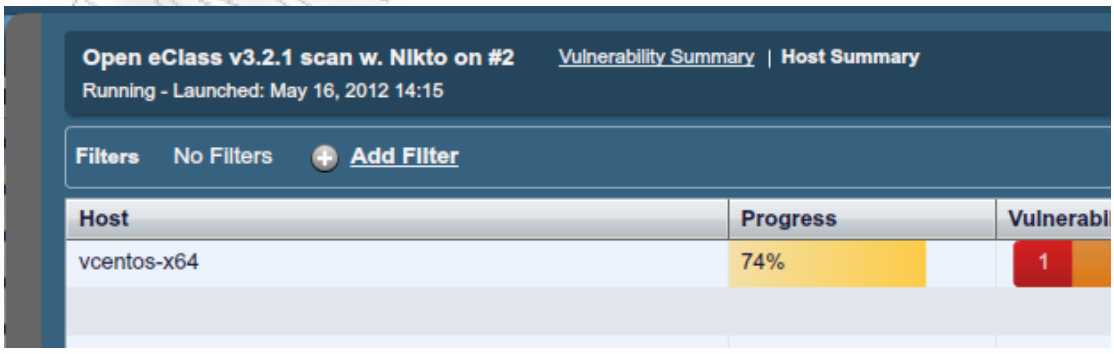
Κεφαλαίο 4 (Αποτελέσματα)

4.1. Εισαγωγή

Στόχος του παρόντος κεφαλαίου είναι η λεπτομερής ανάλυση των τριών κύριων αξόνων που ακολουθούνται για την διενέργεια μιας αποτελεσματικής, αυτοματοποιημένης σάρωσης δικτυακών εφαρμογών. Ο πρώτος από αυτούς τους άξονες είναι , η αντιστοίχηση των αποτελεσμάτων μιας σάρωσης με τις κατηγορίες ευπαθειών που παρουσιάζονται στην λίστα Top 10 OWASP. Ο δεύτερος άξονας περιέχει την ανάλυση των αποτελούν μιας σάρωσης, μέσω της οποίας ο ελεγκτής θα καταρτίσει μια λίστα με τις ευπάθειες που ανακαλύφθηκαν και τις παραμέτρους προγραμματισμού που αυτές επηρεάζουν. Τέλος ο τρίτος άξονας συμπεριλαμβάνει τα συμπεράσματα και τα κρίσιμα βήματα που πρέπει να ακολουθηθούν για την δημιουργία μιας, όσο το δυνατό αποδοτικότερης πολιτικής.

4.2. Ανασκόπηση αποτελεσμάτων από την καρτέλα Results σύμφωνα με τη λίστα Top 10 του OWASP

Όπως ήδη έχει αναγερθεί στο προηγούμενο κεφάλαιο, με την έναρξη μιας σάρωσης μέσω της καρτέλας "Scans" αυτόματα ξεκινά η καταγραφή της, στην καρτέλα "**Reports**". Τα αποτελέσματα ξεκινούν να παρουσιάζονται μερικά δευτερόλεπτα μετά την έναρξη της σάρωσης και τα πρώτα αποτελέσματα είναι αυτά που επιβεβαιώνουν την λειτουργία του στόχου ως web server και την πόρτα 80 ως "ανοικτή". Με την επιλογή του "**Browse**" στην καρτέλα "**Reports**" εμφανίζεται η οθόνη παρουσίασης των αποτελεσμάτων. Η προεπιλογή του τρόπου εμφάνισης των αποτελεσμάτων είναι η "**Vulnerability Summary**" στην οποία γίνεται η κάθετη παρουσίαση των plugins που επέστρεψαν κάποια αποτελέσματα. Η επιλογή του "**Host Summary**", έχει ως αποτέλεσμα την αλλαγή εμφάνισης των αποτελεσμάτων και τη παρουσίαση αυτών μαζί με την μπάρα προόδου της σάρωσης, ενώ βρίσκεται σε εξέλιξη, όπως φαίνεται στην ακόλουθη εικόνα.



Εικόνα 89: Μπάρα εξέλιξης σάρωσης

Σε αυτό το σημείο θα γίνει προσπάθεια αντιστοίχησης των plugins και των αριθμών αναγνωριστικού αυτών, με τη λίστα που παρέχει ο OWASP για τους Top 10 κατηγορίες ευπαθειών που πλήττουν τις εφαρμογές ιστού. Στον παρακάτω πίνακα εμφανίζονται οι αντιστοιχίες αυτές.

Κατηγορία	Περιγραφή
A1 - Injections	• SQL Injection (*CGI abuses*) ❖ **11139, 42424, 42426, 42427, 42479, 43160, 51973** • XML Injection (*CGI abuses*) ❖ **46196** • HTTP Header Injection (*CGI abuses: XSS*) ❖ **39468, 49067** • Cookie Injection (*CGI abuses*) ❖ **44135**
A2 – Cross-Site Scripting (XSS)	• Cross-Site Scripting (*CGI abuses: XSS*) ❖ **10815, 39466, 42425, 47831, 46193, 49067, 51972**
A3 – Broken Authentication and Session Management	• Μη πραγματοποίηση αυθεντικοποίησης μέσω SSL πρωτοκόλλου ❖ **26194, 34850** • Δέουσα εφαρμογή του πρωτοκόλλου SSL ❖ **15901, 20007, 26928, 35291, 42053, 42873 , 42880, 53491, 53360, 56043, 56284, 56984, 57041**
A4 – Insecure Direct Object References	• Δυνατότητα περιήγησης σε καταλόγους ιστού (web catalogs) ❖ **40984** • Path Transversal (*CGI abuses*) ❖ **50494** • Παράμετροι που προσδιορίζονται για χειροκίνητες δοκιμές ❖ **40773, 44134, 47830 ***
A5 – Cross-Site Request Forgery (CSRF)	• On Site Request Forgery (*CGI Generic*) ❖ **47832** • Έλεγχοι για γνωστές ευπάθειες CSRF σε συγκεκριμένα προϊόντα (π.χ. Wordpress, Drupal κτλ).
A6 – Security Misconfiguration	• Καλύπτεται μόνο από ελέγχους που περιέχονται στην εμπορική συνδρομή του Nessus (Professional Feed) • Αναγνώριση των ανοικτών πορτών με δυνατότητα για χειροκίνητους ελέγχους (ports scanners).

	• Έλεγχοι για ανακάλυψη προεπιλεγμένων λογαριασμών και κωδικών πρόσβασης.
A7 – Insecure Cryptographic Storage	*Δεν καλύπτεται μέσω των αυτοματοποιημένων ελέγχων που προσφέρει το Nessus*
A8 – Failure to Restrict URL Access	*Δεν καλύπτεται μέσω των αυτοματοποιημένων ελέγχων που προσφέρει το Nessus*
A9 – Insufficient Transport Layer Protection	• Μη πραγματοποίηση αυθεντικοποίησης μέσω του πρωτοκόλλου SSL ❖ **26194, 34850** • Δέουσα εφαρμογή του πρωτοκόλλου SSL ❖ **15901, 20007, 26928, 35291, 42053, 42873 , 42880, 53491, 53360, 56043, 56284, 56984, 57041** • Χρήση ασφαλούς Cookie ❖ **49218, 84832**
A10 – Unvalidated Redirects and Forwards	• Ανοιχτή ανακατεύθυνση χωρίς έλεγχο (*CGI Generic*) ❖ **47834**

Πίνακας 4: Πίνακας αντιστοίχησης αναγνωριστικών των Nessus plugins με το Top 10 του OWASP 2010

Το Nessus εκτός από το μεγάλο βαθμό κάλυψης στις κατηγορίες ευπαθειών OWASP Top 10 – 2010, περιλαμβάνει και δύο ακόμα κατηγορίες, στις οποίες μειώθηκε η σημαντικότητα τους, και που περιέχονταν στη λίστα **OWASP Top 10 – 2007.** Αυτές παρουσιάζονται στον ακόλουθο πίνακα.

Κατηγορία	Περιγραφή
A3 – Malicious File Execution	• Εκτέλεση εντολών (*CGI abuses*) ❖ **39465,44967**
A6 – Information Leakage and Improper Error Handling	• Directory Traversal (*CGI abuses*) ❖ **39467, 46195, 46194** • Ένταξη αρχείων (File Inclusion) (*CGI abuses*) ❖ **39469, 42056, 42872** • Server Side Includes (*CGI abuses*) ❖ **42423, 42054** • Μηνύματα λαθών ❖ **40406, 48926, 48927**

Πίνακας 5: Πίνακας αντιστοίχησης αναγνωριστικών των Nessus plugins με το Top 10 του OWASP 2007

Συμπληρωματικά στα ανωτέρω plugins, ιδιαίτερο ενδιαφέρον, σχετικά με τις πληροφορίες που παρέχουν, έχουν και τα ακόλουθα:

- Μορφοποίηση συμβολοσειρών (*CGI abuses*)
  - **42055**
- Χειραγώγηση των cookies (*CGI abuses*)
  - **44136**
- Επιπλέον επιθέσεις (*CGI abuses*)
  - **44134, 47830, 47832, 47834**

Να σημειωθεί εδώ ότι, προφανώς στην κάθε σάρωση που πραγματοποιείται μέσω του Nessus δεν εμφανίζονται στην τελική αναφορά όλα τα παραπάνω plugins, αφού μέσα στις αναφορές συμπεριλαμβάνονται μόνο αυτά τα Plugins τα οποία εκτελέστηκαν και είχαν στην έξοδο τους κάποιο αποτέλεσμα.

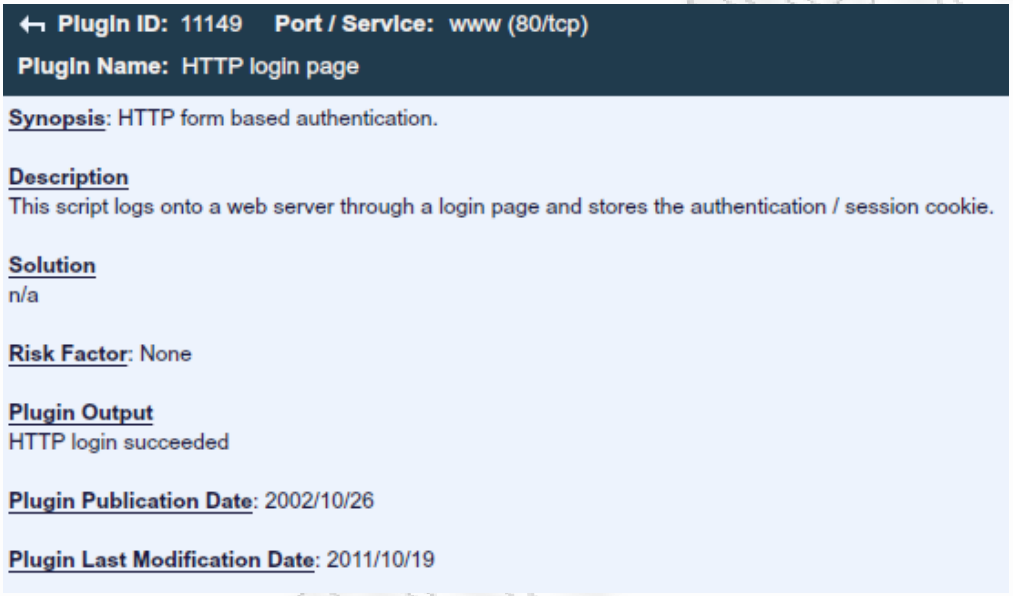
Λαμβάνοντας υπόψη τα προαναφερθέντα και κάνοντας χρήση του φιλτραρίσματος των αποτελεσμάτων με βάση τον αριθμό αναγνωριστικού (plugin ID) ή στην οικογένεια την οποία ανήκουν (CGI abuses), μπορούν να δημιουργηθούν αναφορές με πολύ στοχευμένα αποτελέσματα ανάλογα με τις ευπάθειες που ανακαλύφθηκαν και τη συσχέτιση αυτών με την κατηγορία που ανήκουν στη λίστα Top 10 OWASP. Επιπρόσθετα δίνεται η δυνατότητα μέσω της δικτυακής διεπαφής της εφαρμογής, να πραγματοποιηθεί μια άμεση ταξινόμηση των αποτελεσμάτων ανά πάσα στιγμή, δηλαδή είτε η σάρωση βρίσκεται σε εξέλιξη είτε τελείωσε, με βάση το plugin ID και έτσι με μια γρήγορη ματιά να εντοπιστούν τα συγκεκριμένα plugins που έχουν μεγαλύτερο ενδιαφέρον τα αποτελέσματα τους.

4.3. Ανάλυση αποτελεσμάτων σάρωσης

Τα αποτελέσματα που εμφανίζονται στην τελική αναφορά μιας σάρωσης, είναι αυτά όπου εκτός από το ότι παρουσιάζουν μια γενική εικόνα του επιπέδου ασφάλειας που κατέχει η εφαρμογή, τελικά θα δώσουν τις κατευθυντήριες γραμμές προς τους developers και διαχειριστές, ώστε να προβούν σε διορθωτικές κινήσεις για την αύξηση της ασφάλειας που τελικά είναι και το ζητούμενο από μια τέτοια διαδικασία.

Σχετικά με τα αποτελέσματα του Nessus Θα παρουσιαστούν τα σημαντικότερα plugins καθώς και οι οικογένειες τους, από όπου μπορεί ο ελεγκτής να πάρει τις πλέον πολύτιμες πληροφορίες σχετικά με τη σάρωση. Στην προκειμένη περίπτωση πραγματοποιείται επιθετικοποιημένος έλεγχος στην εφαρμογή Open eClass.

Αφού ο έλεγχος ο οποίος πραγματοποιείται είναι έλεγχος μετά από αυθεντικοποίηση, ένα από τα σημαντικότερα και από τα πρώτα plugins που ψάχνει ο ελεγκτής είναι αυτό της επιβεβαίωσης της επιτυχούς αυθεντικοποίησης στην εφαρμογή, το plugin με αναγνωριστικό **11149 (HTTP Login page)** από το αποτέλεσμα του οποίου θα κριθεί η συνέχιση της σάρωσης ή η διακοπή της για την παρέμβαση στην πολιτική με διορθωτικές κινήσεις στην καρτέλα "**HTTP Login page**" μέχρι την πραγματοποίηση επιτυχούς αυθεντικοποίησης όπως φαίνεται παρακάτω.



Εικόνα 90: Επιβεβαίωση επιτυχούς αυθεντικοποίησης

Μετά τη διαπίστωση της επιτυχούς ή ανεπιτυχούς αυθεντικοποίησης το επόμενο plugin με το μεγαλύτερο ενδιαφέρον παρουσιάζει το plugin με αναγνωριστικό **10662 (Web mirroring)** στα αποτελέσματα του οποίου φαίνεται η ανακάλυψη των λειτουργιών και σημείων εισόδου (entry points) στα οποία, θα πραγματοποιηθούν οι έλεγχοι ασφάλειας. Στην παρακάτω εικόνα φαίνεται ότι η διαδικασία ανακάλυψης (crawl) κατάφερε να βρει έντεκα λειτουργίες που περιέχουν αρκετά σημεία εισόδου για την πραγματοποίηση ελέγχων. Με τον τρόπο αυτό γνωρίζει ο ελεγκτής ότι αυτές είναι οι παράμετροι (σημεία εισόδου) όπου θα ελεγχτούν με τις fuzzing δυνατότητες του Nessus, σε σχέση με όλες τις κατηγορίες ευπαθειών που καλύπτονται στη λίστα OWASP Top 10 και που αναφέρθηκαν στο προηγούμενο υποκεφάλαιο.

Plugin ID: 10662 **Port / Service:** www (80/tcp)
Plugin Name: Web mirroring

Synopsis: Nessus crawled the remote web site.

Description
This script makes a mirror of the remote web site(s) and extracts the list of CGIs that are used by the remote host.

It is suggested that you change the number of pages to mirror in the 'Options' section of the client.

Solution
n/a

Risk Factor: None

Plugin Output
The following CGI have been discovered :

Syntax : cginame (arguments [default value])

/eclass/modules/link/link_goto.php (link_id [1] link_url [http://www.google.com])
/eclass/modules/profile/profile.php (prenom_form [test] nom_form [user] username_form [testUser] submit [ye...)
/eclass/index.php (logout [yes])
/eclass/modules/search/search_incourse.php (search_terms [] subsystems[3] [3] subsystems[5] [10] subsystems[6] [4]...)
/eclass/modules/auth/courses.php (selectCourse[1] [2] changeCourse[1] [2] fc [] selectCourse[0] [1] chan...)
/eclass/modules/auth/opencourses.php (fc [1])
/eclass/modules/search/search.php (reset [Νέα Αναζήτηση] search_terms [] search_terms_title [...)
/eclass/modules/agenda/myagenda.php (month [4] year [2012])
/eclass/modules/help/help.php (topic [Portfolio_student] language [greek])
/eclass/modules/profile/password.php (password_form [] changePass [do] old_pass [] password_form1 [] submit ...)
/eclass/modules/unreguser/unregcours.php (cid [TMA100] u [7] doit [yes])

145 requests were sent in 3.419 s = 42 req/s = 23 ms/req

Plugin Publication Date: 2001/05/04

Plugin Last Modification Date: 2012/04/24

Εικόνα 91: Αποτελέσματα διαδικασίας crawl

Μεγάλη σημασία έχει και η διερεύνηση του plugin με αριθμό αναγνωριστικού **39470 (CGI Generic Tests Timeouts)** μέσω του οποίου δίνεται αναφορά, σχετικά με το ποια plugins-έλεγχοι έχουν τελειώσει χωρίς να ανακαλύψουν κάποια ευπάθεια και σε ποια plugins έχει διακοπεί η λειτουργία τους με αποτέλεσμα να μην προλάβουν να εκτελέσουν όλους τους ελέγχους, σε όλες τις μεταβλητές που ανακαλύφθηκαν. Να σημειωθεί ότι το συγκεκριμένο plugin εμφανίζεται μόνο αφού έχει τελειώσει η σάρωση του στόχου αφού σε αντίθετη περίπτωση δεν μπορεί να γνωρίζει την κατάσταση του κάθε plugin. Σύμφωνα με τα αποτελέσματα του συγκεκριμένου plugin ο ελεγκτής μπορεί να αποφασίσει να επαναλάβει μια σάρωση προβαίνοντας προηγουμένως σε αλλαγές, στην καρτέλα "**Web Application Tests Settings**", που έχουν να κάνουν με το χρόνο εκτέλεσης των ελέγχων, για την περίπτωση που υπήρχε αυξημένος αριθμός timeouts και στους συνδυασμούς ελέγχου των παραμέτρων όπως το "all combinations" και "all flaws".

Plugin ID: 39470 **Port / Service:** www (80/tcp)

Plugin Name: CGI Generic Tests Timeout

Synopsis: Some generic CGI attacks ran out of time.

Description

Some generic CGI tests ran out of time during the scan. The results may be incomplete.

Solution

Run your run scan again with a longer timeout or less ambitious options :

- Combinations of arguments values = 'all combinations' is much slower than 'two pairs' or 'single'.

- Stop at first flaw = 'per port' is quicker.

- In 'some pairs' or 'some combinations' mode, try reducing web_app_tests.tested_values_for_each_parameter in nessusd.conf

Risk Factor: None

Plugin Output

The following tests timed out without finding any flaw :
- SQL injection (2nd order)
- SQL injection
- directory traversal (write access)
- directory traversal
- arbitrary command execution
- directory traversal (extended test)
- XML injection
- web code injection
- persistent XSS
- local file inclusion
- blind SQL injection (time based)
- blind SQL injection

The following tests were interrupted and did not report all possible flaws :
- cookie manipulation
- cross-site scripting (quick test)
- cross-site scripting (comprehensive test)

Plugin Publication Date: 2009/06/19

Plugin Last Modification Date: 2011/03/06

Εικόνα 92: Plugins που έληξαν ή δεν είχαν αποτελέσματα

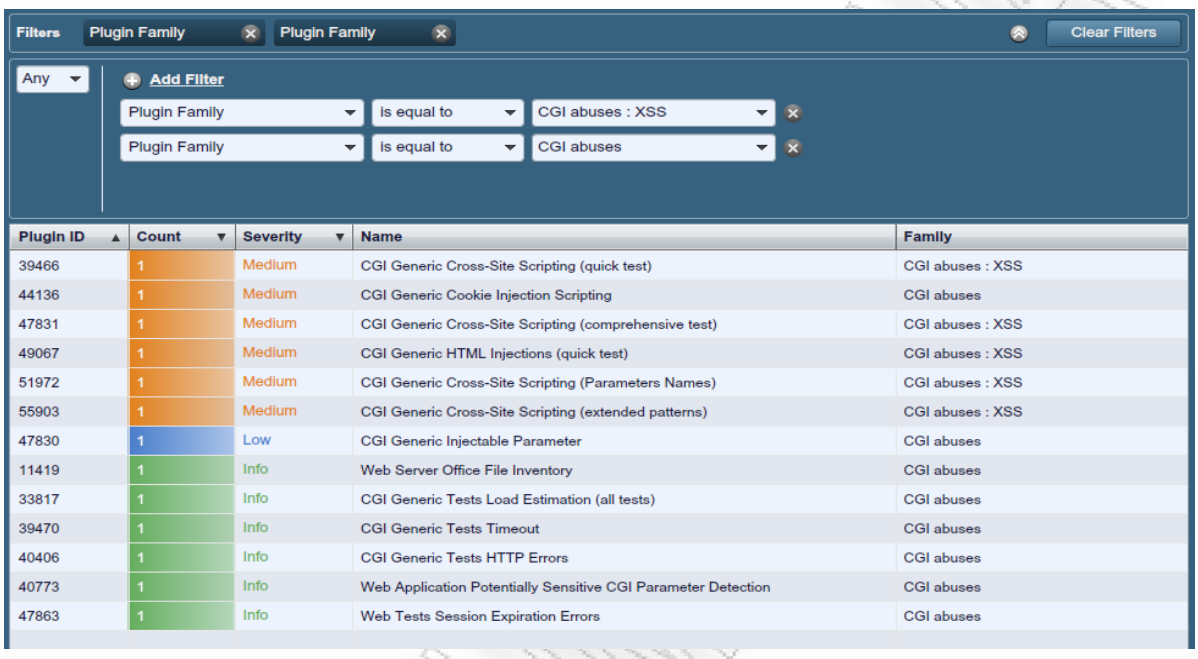
Ένα ακόμη σημαντικό plugin είναι το **47830 (CGI Generic Injectable Parameter)** μέσω του οποίου παρουσιάζονται όλες εκείνες οι μεταβλητές που επιδέχονται την γενική και αδόμητη εισαγωγή δεδομένων. Αυτές οι παράμετροι συνήθως είναι οι ίδιες που δημιουργούν και τα επιπλέον προβλήματα σε ότι έχει να κάνει με τις ευπάθειες τύπου SQL Injections και Cross Site Scripting, αφού μέσω αυτού του plugin διαπιστώνεται ότι δεν προβαίνει η εφαρμογή σε επικύρωση δεδομένων πριν την παραχώρηση τους στην εκάστοτε διεργασία. Να σημειωθεί ότι τα αποτελέσματα αυτού του plugin μπορεί να έχουν πάρα πολύ μεγάλη έκταση αρκετές φορές και επίσης δεν είναι απαραίτητο ότι αυτές οι παράμετροι τελικά παρουσιάζουν πρόβλημα. Βέλτιστη πρακτική, είναι όμως οι προγραμματιστές να προβούν σε έλεγχο των συγκεκριμένων παραμέτρων, ώστε να

διαπιστωθεί η "αθωότητα" τους ή ακόμη καλύτερα να προστεθεί μια διαδικασία επικύρωσης δεδομένων πριν την παραχώρηση τους προς τα έσω της εφαρμογής. Ένα δείγμα της εξόδου αυτού του plugin παρουσιάζεται στην ακόλουθη εικόνα.

Εικόνα 93: Παράμετροι χωρίς επικύρωση δεδομένων

Η χρήση των φίλτρων πάνω στα αποτελέσματα μπορεί να φανεί ιδιαίτερα χρήσιμη και ειδικά όταν αυτά παρουσιάζουν μεγάλο πλήθος, τότε είναι που κρίνεται απαραίτητη η χρήση τους έτσι ώστε να παρουσιαστεί μια ποιο συγκροτημένη και ξεκάθαρη εικόνα των ευρημάτων. Εδώ με τη χρήση δύο φίλτρων που προσδιορίζουν την εμφάνιση αποτελεσμάτων από συγκεκριμένες οικογένειες plugins, μπορεί να γίνει παρουσίαση των αποτελεσμάτων αυτών, που έχουν να κάνουν καθαρά και μόνο με τις αδυναμίες στο επίπεδο εφαρμογής και αυτά είναι τα **CGI Abuses plugin family** και **CGI Abuses: XSS plugin**

Family όπως φαίνονται στην παρακάτω εικόνα. Σε αυτές τις αδυναμίες είναι που παρουσιάζονται και οι λεγόμενες zero-day vulnerabilities, που δεν είναι γνωστές εκ των προτέρων και που οφείλονται συνήθως στα προγραμματιστικά λάθη κατά την ανάπτυξη της εφαρμογής.



Plugin ID	Count	Severity	Name	Family
39466	1	Medium	CGI Generic Cross-Site Scripting (quick test)	CGI abuses : XSS
44136	1	Medium	CGI Generic Cookie Injection Scripting	CGI abuses
47831	1	Medium	CGI Generic Cross-Site Scripting (comprehensive test)	CGI abuses : XSS
49067	1	Medium	CGI Generic HTML Injections (quick test)	CGI abuses : XSS
51972	1	Medium	CGI Generic Cross-Site Scripting (Parameters Names)	CGI abuses : XSS
55903	1	Medium	CGI Generic Cross-Site Scripting (extended patterns)	CGI abuses : XSS
47830	1	Low	CGI Generic Injectable Parameter	CGI abuses
11419	1	Info	Web Server Office File Inventory	CGI abuses
33817	1	Info	CGI Generic Tests Load Estimation (all tests)	CGI abuses
39470	1	Info	CGI Generic Tests Timeout	CGI abuses
40406	1	Info	CGI Generic Tests HTTP Errors	CGI abuses
40773	1	Info	Web Application Potentially Sensitive CGI Parameter Detection	CGI abuses
47863	1	Info	Web Tests Session Expiration Errors	CGI abuses

Εικόνα 94: Αποτελέσματα των CGI Abuses/XSS Plugins

Με τη χρήση και πάλι των φίλτρων και επιλέγοντας την οικογένεια plugin **Web Servers** γίνεται η παρουσίαση όλων των ευπαθειών και πληροφοριών που ανακαλύφθηκαν στον στόχο και που αφορούν το επίπεδο εξυπηρετητή, δηλαδή την υπηρεσία που "σερβίρει" την εφαρμογή, και όχι την εφαρμογή αυτή καθ' αυτή. Οι πληροφορίες εδώ είναι επίσης πολύ σημαντικές γιατί τελικά, μπορεί αυτές οι ευπάθειες να αποτελούν απειλή για την εφαρμογή αλλά και να παρουσιάζουν την λανθασμένη διαμόρφωση του εξυπηρετητή που φιλοξενεί την εφαρμογή. Μέσω αυτών των plugins και των αποτελεσμάτων τους, πολλές φορές προτείνονται λύσεις που έχουν άμεση σχέση με την διαμόρφωση του εξυπηρετητή και δίνουν αρκετά καλές κατευθυντήριες γραμμές ώστε να ξεπεραστούν τα συγκεκριμένα προβλήματα. Τα ευρήματα από αυτή τη σάρωση παρουσιάζονται στην παρακάτω εικόνα.

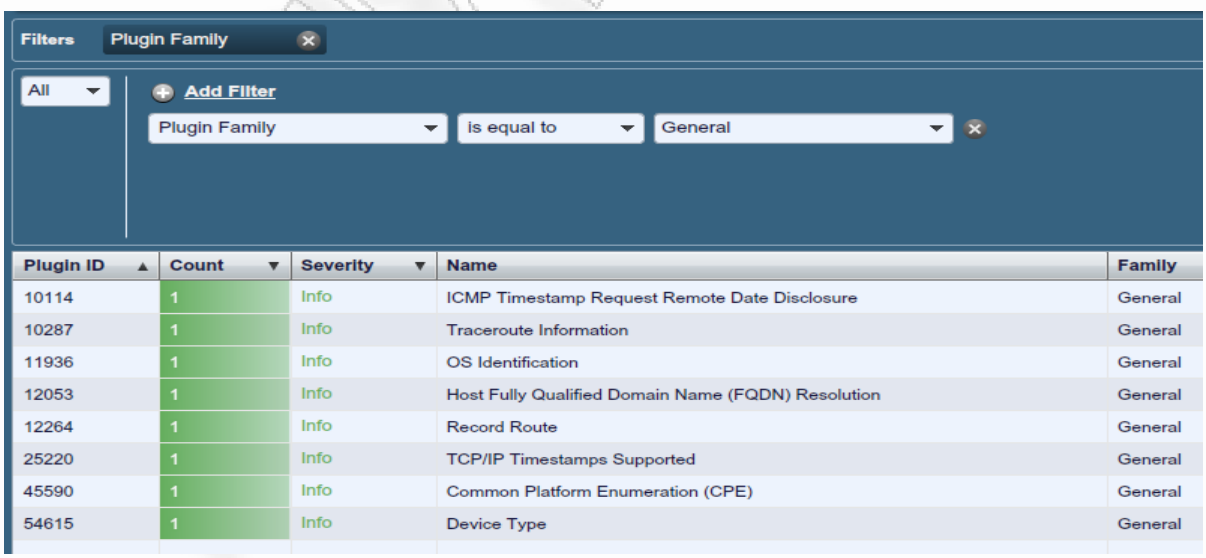
Filters: Plugin Family

Any | Add Filter | Plugin Family | is equal to | Web Servers

Plugin ID	Count	Severity	Name	Family
45084	1	High	Session Fixation Attack on HTTP Cookies	Web Servers
11213	1	Medium	HTTP TRACE / TRACK Methods Allowed	Web Servers
46201	1	Medium	Fixed HTTP Session Cookies	Web Servers
46803	1	Medium	PHP expose_php Information Disclosure	Web Servers
48432	1	Medium	Web Application Session Cookies Not Marked HttpOnly	Web Servers
49218	1	Medium	Web Application Session Cookies Not Marked Secure	Web Servers
26194	1	Low	Web Server Uses Plain Text Authentication Forms	Web Servers
10107	1	Info	HTTP Server Type and Version	Web Servers
10662	1	Info	Web mirroring	Web Servers
11032	1	Info	Web Server Directory Enumeration	Web Servers
11919	1	Info	HMAP Web Server Fingerprinting	Web Servers
24260	1	Info	HyperText Transfer Protocol (HTTP) Information	Web Servers
39463	1	Info	HTTP Server Cookies Set	Web Servers
42057	1	Info	Web Server Allows Password Auto-Completion	Web Servers
43111	1	Info	HTTP Methods Allowed (per directory)	Web Servers
44987	1	Info	HTTP Session Cookies	Web Servers
49704	1	Info	External URLs	Web Servers
52973	1	Info	Restricted Web Pages Detection	Web Servers

Εικόνα 95: Αποτελέσματα των Web Servers Plugins

Επιπλέον, μια ακόμη οικογένεια plugin που παρουσιάζει ιδιαίτερο ενδιαφέρον ως προς τις πληροφορίες που προσφέρει και οι οποίες βρίσκονται σε ακόμη χαμηλότερο επίπεδο είναι η **General**. Εδώ παρουσιάζονται οι πληροφορίες που αφορούν το επίπεδο δικτύου και λειτουργικού συστήματος του εξυπηρετητή που φιλοξενεί την εφαρμογή.



Filters: Plugin Family

All | Add Filter | Plugin Family | is equal to | General

Plugin ID	Count	Severity	Name	Family
10114	1	Info	ICMP Timestamp Request Remote Date Disclosure	General
10287	1	Info	Traceroute Information	General
11936	1	Info	OS Identification	General
12053	1	Info	Host Fully Qualified Domain Name (FQDN) Resolution	General
12264	1	Info	Record Route	General
25220	1	Info	TCP/IP Timestamps Supported	General
45590	1	Info	Common Platform Enumeration (CPE)	General
54615	1	Info	Device Type	General

Εικόνα 96: Αποτελέσματα των General Plugins

Κεφάλαιο 5 (Οδηγίες κατάρτισης πολιτικών)

Η κατάρτιση των πολιτικών σάρωσης είναι μια πολύ σημαντική διαδικασία, κατά την οποία απαιτείται αρκετή χειρονακτική εργασία από τον ελεγκτή, για την ορθή δόμηση της. Αυτό θα έχει όμως ως αποτέλεσμα την βέλτιστη απόδοση μιας σάρωσης, ως προς τα παραγόμενα αποτελέσματα που θα παρουσιαστούν στην τελική αναφορά. Τα αποτελέσματα θα πρέπει να προσεγγίζουν όσο το δυνατό την πραγματική κατάσταση ασφάλειας της εφαρμογής τόσο σε γνωστές, όσο και σε άγνωστες ευπάθειες και αυτό εξασφαλίζεται μόνο με την τήρηση κάποιων κανόνων. Για το λόγο αυτό είναι πολύ σημαντικό να ακολουθηθούν προσεχτικά ορισμένες κατευθυντήριες γραμμές κατά τη δημιουργία μιας πολιτικής, οι οποίες παρουσιάζονται πιο κάτω:

1. Καταγραφή πακέτου αυθεντικοποίησης με τη βοήθεια διαμεσολαβητών. Αυτή είναι μια καλή πρακτική κατά την οποία εξασφαλίζεται η σύλληψη του ακριβές αλφαριθμητικού (string) που χρησιμοποιείται κατά την διαδικασία της αυθεντικοποίησης για τον ορθό καθορισμό των απαραίτητων πεδίων στην δημιουργία της πολιτικής.

2. Χειροκίνητη περιήγηση στην εφαρμογή για την ανακάλυψη συνδέσμων και λειτουργιών, μέσω της οποίας γίνεται η καταγραφή του αριθμού-πλήθους των συνδέσμων που αποτελούν την εφαρμογή. Αυτό είναι χρήσιμο κατά την διάρκεια της σάρωσης, ώστε να διαπιστωθεί, κατά πόσο κατόρθωσε το αυτοματοποιημένο εργαλείο να ανακαλύψει όλες τις λειτουργίες αλλά και σημεία εισόδου της εφαρμογής συγκρίνοντας τους δύο αριθμούς.

3. Καταγραφή των αλφαριθμητικών (strings) αποφυγής. Αυτά είναι οι λέξεις κλειδιά, πίσω από τις οποίες κρύβονται οι λειτουργίες που πρέπει να αποφευχθούν κατά τη διάρκεια της διαδικασίας αυτόματης ανακάλυψης λειτουργιών και συνδέσμων (crawl), αφού η επιλογή τους προκαλεί είτε την αποσύνδεση του χρήστη από την εφαρμογή, είτε ακόμη και τη διαγραφή αυτού με αποτέλεσμα την αδυναμία αυθεντικοποιημένων ελέγχων.

4. Επισκόπηση του κώδικα της εφαρμογής μέσω του φυλλομετρητή. Απαραίτητη διαδικασία η οποία συνδυάζεται με τις δύο προηγούμενες αφού μέσα στον κώδικα της ιστοσελίδας μπορεί να γίνει συλλογή από πολλές και πολύτιμες

πληροφορίες, σχετικά με τις λειτουργίες της εφαρμογής αλλά και των αλφαριθμητικών αποφυγής.

5. Χρήση διαφορετικών use-cases, που προκαλούν την διαφοροποίηση στην συμπεριφορά κάποιων λειτουργιών της εφαρμογής, με σκοπό την κατανόηση της γενικής συμπεριφοράς αυτής, ώστε να μπορούν οι συγκεκριμένες λειτουργίες να επαληθευτούν ή αποφευχθούν κατά τη διάρκεια της σάρωσης.

6. Τέλος η συνεχής παρακολούθηση της εφαρμογής κατά τη διάρκεια της σάρωσης μέσω κάποιου λογαριασμού εποπτείας, συνήθως αυτός του διαχειριστή, είναι απαραίτητη για την εξασφάλιση της ορθής αδιάκοπης λειτουργία της εφαρμογής αλλά και της σάρωσης.

Όση πείρα και αν διαθέτει ένας ελεγκτής, είναι γενικά αποδεχτό, ότι η επανάληψη των σαρώσεων και τα επαναλαμβανόμενα βήματα διόρθωσης μιας πολιτικής είναι αναπόφευκτα, ειδικά κατά την πραγματοποίηση των αρχικών σαρώσεων ελέγχων σε καινούριες εφαρμογές όπου δεν έχει ξανασυναντήσει. Σημαντικό παράγοντα επίσης, για την επιτυχή διενέργεια των ελέγχων, από αυτοματοποιημένα εργαλεία που κάνουν χρήση της μεθόδου black box, κατέχει η συνεργασία και συνεχής ανατροφοδότηση από τους developers και διαχειριστές της εφαρμογής σχετικά με το πώς πρέπει να διατηρηθούν κάποια στοιχεία αυτής και πως πρέπει να διενεργηθούν ορισμένοι έλεγχοι λαμβάνοντας πάντα σοβαρά υπόψη την κρισιμότητα της εφαρμογής προς έλεγχο.

Κεφάλαιο 6 (Σύνοψη και Συμπεράσματα)

Υπάρχουν πολλοί μέθοδοι ελέγχου ασφάλειας των εφαρμογών ιστού. Ένας από αυτούς και ίσως ο πιο αποτελεσματικός είναι η επιθεώρηση ασφάλειας του κώδικα της εφαρμογής. Αυτός ο έλεγχος, αποτελείτε από μια λεπτομερή και μεθοδική διαδικασία, κατά την οποία απαιτείτε μεγάλη χειρωνακτική προσπάθεια, σημαντικό υπόβαθρο γνώσεων και άφθονο χρόνο στη διάθεση του προσωπικού ασφάλειας, που θα πραγματοποιήσει το συγκεκριμένο έλεγχο. Για τους λόγους αυτούς, όπου αποτελούν και τα μειονεκτήματα της μεθόδου, συνήθως αναλαμβάνεται από εξειδικευμένο οργανισμό. Το γεγονός αυτό, όμως, αυξάνει το κόστος ελέγχου, που ίσως είναι και το πιο σημαντικό μειονέκτημα, το οποίο και καθιστά την επιλογή της μεθόδου αυτής απαγορευτική. Κατά συνέπεια οι περισσότεροι οργανισμοί-επιχειρήσεις καταφεύγουν στη λύση του ελέγχου ασφάλειας με τη μέθοδο της black box δοκιμής διείσδυσης, κάτι το οποίο μπορεί να πραγματοποιηθεί ακόμα και από το ίδιο το προσωπικό του οργανισμού. Διαπιστώνεται η γενική παραδοχή ότι η μέθοδος της δοκιμής διείσδυσης με την black box τεχνική είναι η βέλτιστη μέθοδος για τον έλεγχο των εφαρμογών ιστού. Η μέθοδος αυτή αφορά τον έλεγχο ασφάλειας της εφαρμογής μέσω της εξωτερικής της διεπαφής, δηλαδή αυτή που χρησιμοποιείται από τους τελικούς χρήστες και που δεν λαμβάνει καθόλου υπόψη την εσωτερική δομή της εφαρμογής καθώς επίσης και το γεγονός ότι την ίδια διεπαφή χρησιμοποιούν και οι επίδοξοι εισβολείς κατά της εφαρμογής ενώ, πολλές από τις μεθόδους σάρωσης είναι πανομοιότυπες με αυτές των επιτιθέμενων. Για να πραγματοποιηθεί ένας αποτελεσματικός έλεγχος μιας εφαρμογής δικτύου με τη μέθοδο αυτή, θα πρέπει ο ελεγκτής να “παίξει” πολύ με την εφαρμογή, να προσδιορίσει τι αποτελεί “αντικανονική εισαγωγή” δεδομένων για την εφαρμογή παρατηρώντας καλά κάθε λειτουργία της και να προσπαθήσει να εισάγει τα αντικανονικά δεδομένα. Αυτό βέβαια, είναι μια πολύ χρονοβόρα διαδικασία και απαιτεί βαθιά γνώση σε θέματα ασφάλειας, ώστε να είναι εφικτός ο προσδιορισμός των ευπαθειών, που πιθανόν να υπάρξουν. Το όφελος βέβαια είναι, ότι κατά αυτό τον τρόπο, εξετάζεται και η λειτουργικότητα της εφαρμογής ιστού, αφού ο ελεγκτής συμπεριφέρεται σαν ένας απλός εξωτερικός χρήστης της.

Επειδή, όπως αναφέρθηκε ήδη, είναι δύσκολη η χειροκίνητη μέθοδος της δοκιμής διείσδυσης με την black box τεχνική, γίνεται χρήση αυτοματοποιημένων εργαλείων για το σκοπό αυτό. Τα εργαλεία αυτά ενδέχεται να είναι εργαλεία, που αυτοματοποιούν το σύνολο των ελέγχων ή να αυτοματοποιούν ορισμένες μόνο από τις λειτουργίες ελέγχων.

Υπάρχουν και εργαλεία, που είναι πολύ εξειδικευμένα και χρησιμοποιούνται μόνο από ειδικούς ασφάλειας σε χειροκίνητους ελέγχους. Όλα αυτά τα εργαλεία, συνήθως, διατηρούν βάσεις δεδομένων με γνωστές ευπάθειες και σύμφωνα με αυτές κάνουν τους ελέγχους. Το ζήτημα, όμως, είναι κατά πόσο αυτά τα εργαλεία έχουν ενημερωμένες βάσεις δεδομένων, κατά πόσο κάνουν σωστά τους ελέγχους που επιθυμούμε και συνεπώς σε ποιο βαθμό μπορούμε να τα εμπιστευόμαστε. Χρησιμοποιώντας διάφορα εργαλεία και με τους ελέγχους που διενεργήθηκαν, διαπιστώθηκε ότι δεν εξάγονται τα ίδια αποτελέσματα. Αυτό αποτελεί μια ένδειξη του ότι δεν είναι όλα τα εργαλεία αποτελεσματικά, καθώς επίσης ότι δεν έχουν όλα τις ίδιες δυνατότητες ελέγχου ασχέτως, αν στις δυνατότητες, που διαφημίζουν, αναφέρονται. Αυτό το φαινόμενο πιθανότατα είναι πιο έντονο στα εμπορικά εργαλεία, όπου ο κώδικας τους είναι κλειστός και δεν είναι εφικτό κάποιος να ελέγξει την αποτελεσματικότητα τους. Μάλιστα οι περισσότερες demo εκδόσεις τέτοιων εργαλείων, λειτουργούν μόνο με ιστότοπους, που περιέχουν εφαρμογές ιστού με πολλές ευπάθειες προσπαθώντας να πείσουν τους πιθανούς πελάτες τους για την αποτελεσματικότητα τους. Σε αυτές τις περιπτώσεις είναι ανάγκη να δίνεται ιδιαίτερη βαρύτητα στην εταιρεία, που αναπτύσσει το κάθε εργαλείο. Τα ανοικτού λογισμικού εργαλεία υπερέχουν κατά κάποιο τρόπο σε αυτό το σημείο, διότι μπορεί κάποιος να ελέγξει τον κώδικά τους, να βρει πληροφορίες για τη λειτουργία τους από διάφορους σχετικούς ιστότοπους, καθώς επίσης μπορεί ακόμη και να ζητήσει πληροφορίες από τους υπεύθυνους ανάπτυξης τους. Τα συμπεράσματα, που εξάγονται από τη χρήση διαφόρων εργαλείων, είναι ότι τα εμπορικά εργαλεία είναι φιλικά προς το χρήστη με καλή τεκμηρίωση και οδηγίες για τη χρήση τους και γενικά μπορούν να χρησιμοποιηθούν εύκολα από χρήστες, που δεν έχουν εξειδικευμένες γνώσεις και εμπειρία, στον έλεγχο της ασφάλειας εφαρμογών ιστού. Πολλές φορές, παρέχουν αρκετές πληροφορίες ώστε να δοθεί η δυνατότητα, εκτός από πλήρεις αυτοματοποιημένους ελέγχους να εκτελεσθούν και χειροκίνητοι έλεγχοι από τους πιο έμπειρους και με προγραμματιστικές γνώσεις χρήστες. Παρέχουν τα αποτελέσματα ανίχνευσης (scan reports) μαζί με αναλυτικές πληροφορίες για τις ευπάθειες, που έχουν εντοπισθεί, καθώς επίσης και συμβουλές για την επίλυση των προβλημάτων. Όσον αφορά τα ανοικτού λογισμικού ή ελεύθερα εργαλεία, τα περισσότερα δεν έχουν καλή τεκμηρίωση των λειτουργιών τους και δεν παρέχουν αρκετές οδηγίες για τη χρήση τους. Ακόμη, τα περισσότερα εργαλεία αυτού του είδους παρέχουν κάποιες συγκεκριμένες λειτουργίες ή αυτοματοποιούν ένα περιορισμένο σύνολο από τεστ ελέγχου.

Ως τελικό συμπέρασμα που μπορεί να εξαχθεί, είναι η διαπίστωση της πολυπλοκότητας που αποτελεί το συγκεκριμένο τομέα της ασφάλειας. Η πλήρης

αυτοματοποίηση των ελέγχων ασφάλειας είναι πρακτικά αδύνατη, λόγω της διαφορετικότητας των εφαρμογών ιστού, τόσο στον τρόπο ανάπτυξης τους (γλώσσες προγραμματισμού) όσο και στον τρόπο ή το μέσο από όπου “σερβίρονται” στο κοινό (διαφορετικοί εξυπηρετητές ιστού και τεχνολογίες). Παράλληλα η συνεχής εξέλιξη, των δικτυακών εφαρμογών, που σκοπό έχει τη βελτίωση των υπηρεσιών που προσφέρουν και τον εμπλουτισμό των λειτουργιών τους, κάνει αυτό το έργο ακόμη δυσκολότερο, αλλά και ποιο απαραίτητο, καθώς δημιουργεί την ανάγκη της συχνής επανάληψης αυτών των ελέγχων. Εδώ εμφανίζεται επίσης η ανάγκη της χρήσης εργαλείων που εξελίσσονται και αυτά με τη σειρά τους και έχουν συνεχείς ενημερώσεις ώστε να συμβαδίζουν με τις αλλαγές και την εξέλιξη των εφαρμογών αλλά και των τεχνολογιών που τις αποτελούν. Για το λόγο αυτό, η χρήση αυτοματοποιημένων εργαλείων σάρωσης, όπως είναι το Nessus, γίνεται απαραίτητη, αφού παρέχουν μια πλειάδα από πληροφορίες που μπορεί να αποτελέσουν σημαντική βοήθεια, τόσο για τους μηχανικούς ανάπτυξης εφαρμογών όσο και για τους διαχειριστές και τεχνικούς που συντηρούν τις εφαρμογές και τους εξυπηρετητές όπου τις φιλοξενούν, ώστε να διασφαλίσουν την ορθή και ομαλή λειτουργία των εφαρμογών και του οργανισμού που του ανήκουν.

Βιβλιογραφία

1. Μακρής Χαράλαμπος (2011), «Αξιολόγηση ευπαθειών και τρόποι αντιμετώπισης τους σε διαδικτυακές εφαρμογές και εξυπηρετητές», Τμήμα Ψηφιακών Συστημάτων, Πανεπιστήμιο Πειραιά.

2. Παπακρίβος Χρήστος (2008), «Μέτρηση Ασφάλειας Εφαρμογών Ιστού», Διπλωματική εργασία, Τμήμα Εφαρμοσμένης Πληροφορικής, Πανεπιστήμιο Μακεδονίας, Θεσσαλονίκη.

3. Thomas Wilhelm (August 2009), Professional Penetration Testing – Creating and Operating a formal hacking Lab.

4. Michel de Graaf (2009), "Intelligent fuzzing of web applications", Master thesis, department of Software Engineering, University Van Amsterdam

5. Kathleen Hickey (March 24, 2011), GCN The online authority for government IT professionals, «Cyberattacks on agencies increase as preparedness lags» - http://gcn.com/articles/2011/03/24/omb-reports-spike-in-agency-cyber-attacks.aspx

6. White or Black box testing - http://onjava.com/onjava/2003/05/07/blackboxwebtest.html

7. OWASP, "A Guide to Building Secure Web Application" : www.owasp.org

8. OWASP, "OWASP Testing Guide 2008 V3.0" : https://www.owasp.org/images/5/56/OWASP_Testing_Guide_v3.pdf

9. OWASP Top 10 for 2010 : https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

10. OWASP Top 10 for 2007 : https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

11. Καζακώνης Αναστάσιος (Ιούνιος 2005), OWASP, The Open Web Application Security, Application Security FAQ – OWASP_faq_Greek.pdf

12. Ron Gula, Michel Arboi (2011), «Performing PCI DSS and OWASP Web Application Audits with Nessus» :

http://www.nessus.org/sites/drupal.dmz.tenablesecurity.com/files/uploads/documents/whitepapers/nesssus-web-based-auditing_0.pdf

13. CERT The Open Web Application Project : http://www.cert.org
14. CERT The Open Web Application Project : http://www.cert.org/stats/
15. SECTOOLS.ORG : http://sectools.org/
16. TENABLE Network Security, Nessus : http://nessus.org
17. Fuzzing, OWASP The open web application security project : https://www.owasp.org/index.php/Fuzzing
18. Nessus 5 User Guide : http://static.tenable.com/documentation/nessus_5.0_user_guide.pdf
19. Performing PCI DSS and OWASP Web Application Audits with Nessus, nesssus-web-based-auditing.pdf
20. Nessus 5.0 Installation and Configuration Guide, nessus 5.0 installation guide.pdf
21. Web Application Scanning with Nessus, Tenable Web App Scanning.pdf
22. Using Nessus In Web Application Vulnerability Assessments, Using Nessus in Web Application Vulnerability Assessments.pdf
23. CIRT.net Suspicion Breeds Confidence, Nikto : http://cirt.net/nikto2
24. Gunter Ollmann: "Web Based Session Management", http://www.technicalinfo.net/papers/WebBasedSessionManagement.html
25. RFC 2965 - http://www.ietf.org/rfc/rfc2965
26. ha.ckers, web application security lab : http://ha.ckers.org/xss.html
27. Permalinks, Wikipedia, the free encyclopedia : http://en.wikipedia.org/wiki/Permalink
28. Web Crawlers, Wikipedia, the free encyclopedia : http://en.wikipedia.org/wiki/Web_crawler

ΠΑΡΑΡΤΗΜΑ Α

Αποτελέσματα σάρωσης εφαρμογών επίσημου ιστότοπου τμήματος Ψηφιακών Συστημάτων Πανεπιστημίου Πειραιώς

Για λεπτομέρειες παρακαλώ επικοινωνήστε με το συγγραφέα της διπλωματικής εργασίας στο ακόλουθο email:

gpafios -*at*- gmail -*dot*- com