Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακή Διατριβή
στα Πληροφοριακά και Επικοινωνιακά Συστήματα

Empirical Evaluation of State-of-the-Art Penetration Tools

Νέστορας Χουλιάρας

Επιβλέπων Καθηγητής
Ηλίας Αθανασόπουλος

Μάιος 2017

Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Empirical Evaluation of State-of-the-Art Penetration Tools

Νέστορας Χουλιάρας

Επιβλέπων Καθηγητής
Ηλίας Αθανασόπουλος

Η παρούσα μεταπτυχιακή διατριβή υποβλήθηκε
προς μερική εκπλήρωση των απαιτήσεων για απόκτηση

μεταπτυχιακού τίτλου σπουδών
στα Πληροφοριακά Συστήματα

από τη Σχολή Θετικών και Εφαρμοσμένων Επιστημών
του Ανοικτού Πανεπιστημίου Κύπρου

Μάιος 2017

ΛΕΥΚΗ ΣΕΛΙΔΑ

Περίληψη

Σκοπός της παρούσας μεταπτυχιακής διατριβής είναι η αξιολόγηση των εργαλείων διείσδυσης ανοιχτού κώδικα χρησιμοποιώντας μια εφαρμογή ιστού με γνωστές ευπάθειες.

Στα πλαίσια της μεταπτυχιακής διατριβής θα αναλυθεί τι είναι μια δοκιμή διείσδυσης, πότε πρέπει να εφαρμόζεται, με ποια εργαλεία, ποιες μεθόδους και σε ποιες περιοχές. Περιγράφονται οι δέκα πιο σημαντικές ευπάθειες των εφαρμογών ιστού σύμφωνα με τον οργανισμό OWASP με παραδείγματα και τρόπους αντιμετώπισης. Το ενδιαφέρον επικεντρώνεται στους αυτοματοποιημένους σαρωτές ιστού και αναλύεται η λειτουργία τους, τα πλεονεκτήματα και τα μειονεκτήματα τους καθώς και τα χαρακτηριστικά τους.

Για τον σκοπό αυτό δημιουργήθηκε ένα εργαστήριο όπου με εργαλεία διείσδυσης μαύρου κουτιού διεξήχθηκαν δοκιμές διείσδυσης για την εύρεση ευπαθειών στην σουίτα δοκίμων (testbed) WackoPicko με γνώστες ευπάθειες. Μετρήθηκε η κάλυψη, η απόδοση, ο χρόνος που απαιτήθηκε για την ολοκλήρωση της δόκιμης, το δικτυακό αποτύπωμα και τα χαρακτηριστικά τους.

Τα αποτελέσματα των ελέγχων αποδεικνύουν ότι τα εργαλεία διείσδυσης ανοιχτού κώδικα που χρησιμοποιούνται για την ανίχνευση ευπαθειών σε εφαρμογές ιστού, αν και ανίχνευσαν XSS και SQLi ευπάθειες, πάσχουν από μεγάλο ποσοστό ψευδώς θετικών αποτελεσμάτων.

Λέξεις Κλειδιά: Penetration Tools, Web Application Security, Web Vulnerabilities

Summary

The purpose of this master thesis is to evaluate open source penetration tools using a web application with known vulnerabilities.

In the framework of this master thesis, we analyze what is a penetration test, when it should be applied, with what tools, methods and areas. We describe the ten most significant vulnerabilities of Web applications according to OWASP with examples and ways of dealing with them. We focus on automated black box scanners, and we analyze their function, strengths and disadvantages and their features.

For this purpose, a laboratory was created where penetration testing was carried out with black box penetration tools to identify vulnerabilities in the WackoPicko testbed suite with known vulnerabilities. Coverage, performance, time required to complete the test, network footprint, and features of tools were measured.

Audit results show that open source penetration tools used to detect vulnerabilities in web applications, although they detected XSS and SQLi vulnerabilities, but suffer from a large percentage of false-positive results.

Keywords: Penetration Tools, Web Application Security, Web Vulnerabilities

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον υπεύθυνο καθηγητή μου Δρ. Ηλία Αθανασόπουλο για την κοινή επιλογή του θέματος και την καθοδήγηση του στην εκπόνηση της παρούσας μεταπτυχιακής διατριβής. Επίσης θα ήθελα να ευχαριστήσω την οικογένεια μου για την στήριξη που μου παρείχε στην υλοποίηση της.

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Κεφάλαιο 1
Εισαγωγή

Οι εφαρμογές ιστού τα τελευταία χρόνια έχουν γίνει βασική πηγή ενημέρωσης και συναλλαγών. Τράπεζες, κοινωνικά δίκτυα, ενημερωτικά δίκτυα και εμπορικές επιχειρήσεις χρησιμοποιούν τις εφαρμογές ιστού για να αναπτύξουν τις υπηρεσίες τους. Οι εφαρμογές ιστού χειρίζονται ευαίσθητα προσωπικά δεδομένα και πρέπει η ασφάλεια να είναι βασικό στοιχείο για την προστασία τους. Η ασφάλεια του διαδικτύου όμως δεν είναι εύκολη υπόθεση. Οι ιστοσελίδες με μη ενημερωμένες ευπάθειες είναι περίπου 15% [47]. Ευαίσθητα προσωπικά δεδομένα είκοσι ετών [44] εκτίθενται από μια ευπάθεια LFI. Η έκθεση της Verizon [46] αναφέρει ότι το 40% των συμβάντων ασφάλειας τραπεζικών δεδομένων είναι από τις εφαρμογές ιστού. Συμφώνα με την ετήσια αναφορά ασφαλείας της ENISA [42] οι επιθέσεις σε εφαρμογές ιστού είναι στην τρίτη θέση τον δεκαπέντε σημαντικότερων κυβερνοεπιθέσεων και θα αυξηθούν κατά 15%. Το 2017 προβλέπει η ARBOR ότι ο όγκος των επιθέσεων θα φτάσει τα 1,2 Tps [45].

Ένα αναπόσπαστο κομμάτι της διατήρησης της ασφάλειας ενός οργανισμού είναι οι δοκιμές διείσδυσης, οι οποίες πρέπει να διεξάγονται σε κάθε αλλαγή που γίνεται σε οποιοδήποτε μέρος της συνολικής ασφάλειας, όπως η αλλαγή στον κώδικα της εφαρμογής ιστού. Οι δοκιμές διείσδυσης προσομοιάζουν έναν επιτιθέμενο, γιατί ο έλεγχος γίνεται χωρίς να γνωρίζουμε τον πηγαίο κώδικα και την δομή της εφαρμογής που ελέγχουμε. Τα εργαλεία που χρησιμοποιούνται για την δοκιμή, στέλνουν ειδικές εισόδους στην εφαρμογή ιστού και αναλύουν την απόκριση καθορίζοντας εάν υπάρχει ή όχι ευπάθεια. Η χρήση τέτοιων εργαλείων βελτιώνει την ασφάλεια των και μειώνει τον χρόνο κατά την ανάπτυξη του κώδικα [09]. Τα εργαλεία αυτά είναι εύκολα στην χρήση και στην παραμετροποίηση για όσους δεν είναι εξοικειωμένοι [10]. Το μεγαλύτερο μειονέκτημα είναι η παρουσία ψευδώς θετικών αποτελεσμάτων [09] γι' αυτό και είναι σημαντικό να γνωρίζουμε τις ικανότητες τους και τις αδυναμίες τους.

1.1 Σχετικές Εργασίες

Υπάρχουν πολλές εργασίες που χρησιμοποιούν ευπαθείς εφαρμογές ιστού ως σουίτες δοκιμών, για να αξιολογήσουν τις δυνατότητες και τις αδυναμίες των εργαλείων που χρησιμοποιούν. Οι εφαρμογές μπορούν να χωριστούν σε δυο κατηγορίες.

Η πρώτη κατηγορία περιλαμβάνει εφαρμογές που δημιουργήθηκαν για κάποιο σκοπό [38] αλλά εμπεριείχαν ευπάθειες που μπορούν να χρησιμοποιηθούν από επιτιθέμενους. Τέτοιες εφαρμογές είναι οι προηγούμενες εκδόσεις πακέτων CMS όπως WordPress, Drupal, Joomla και PHPBB2 .

Η δεύτερη κατηγορία αφορά εφαρμογές που δημιουργήθηκαν για να δοκιμάζουν τα εργαλεία διείσδυσης και για την εκπαίδευση αξιολογητών σε ελέγχους διείσδυσης. Η δεύτερη κατηγορία μπορεί να χωριστεί σε δυο υποκατηγορίες, αν η σουίτα δοκιμών έχει δημιουργηθεί για εμπορικούς ή ερευνητικούς σκοπούς. Σουίτες που έχουν δημιουργηθεί για εμπορικούς σκοπούς είναι από εταιρίες που έχουν δημιουργήσει και εμπορικά εργαλεία διείσδυσης όπως η Acunetic, HP, IBM, McAfee. Εφαρμογές που έχουν προέλθει από ερευνητές, ερευνητικά προγράμματα ή οργανισμούς είναι το WackoPicko που χρησιμοποιείται στην παρούσα διπλωματική εργασία το WebGoat, DVWA και το Mutillidae.

Οι σουίτες δοκιμών έχουν αναπτυχθεί σε διάφορες πλατφόρμες, όπως php, asp.net, java και python.

Ο Bau [01] δημιουργεί μια σουίτα δοκιμών και αξιολογεί οκτώ εμπορικά εργαλεία και συμπεραίνει ότι τα εργαλεία χρειάζονται βελτίωση για να ανιχνεύσουν αποθηκευμένες XSS και SQLi ευπάθειες, στο ενεργό περιεχόμενο των scripting γλωσσών.

Ο Doupe [02] χρησιμοποιεί την σουίτα δοκιμών WackoPicko και αξιολογεί έντεκα εργαλεία εμπορικά και ανοιχτού κώδικα. Συμπεραίνει ότι έξι από τις οχτώ ευπάθειες δεν μπορούν να ανιχνευτούν και προτείνει πιο έξυπνους αλγορίθμους για το crawling των εφαρμογών ιστού.

Ο Sagala [28] χρησιμοποιεί ένα εμπορικό εργαλείο και ένα εργαλείο ανοιχτού κώδικα δοκιμάζει πέντε εφαρμογές ιστού συμπεραίνοντας ότι απαιτείται η χρήση τέτοιων εργαλείων τους για την βελτίωση της ασφάλειας εφαρμογών ιστού.

Ο Muñoz [38] χρησιμοποιώντας τρεις σουίτες δοκιμών ώστε να καλύψει περισσότερους τύπους ευπαθειών ιστού, αξιολογεί τα ποσοστά κάλυψης τριών εργαλεία ανοιχτού κώδικα.

Ο Vieira [39] αξιολόγησε τρία εμπορικά εργαλεία σάρωσης και τα αποτελέσματα δείχνουν ότι η κάλυψη είναι χαμηλή και το ποσοστό ψευδών θετικών είναι πολύ υψηλά.

Ο McAllister [40] δείχνει ότι εμπορικά και ανοιχτού κώδικα εργαλεία με μικρά ποσοστά ανίχνευσης μπορούν με την προτεινόμενη τεχνική να αυξήσουν τα ποσοστά ανίχνευσης ευπαθειών.

1.2 Ερευνητικές Ερωτήσεις

Τα ερευνητικά ερωτήματα που στοχεύει να διατυπώσει αυτή η εργασία παρέχονται σε αυτό το τμήμα. Θα μετρηθούν

- Η απόδοση των εργαλείων
- Ο χρόνος που χρειάζεται για να ολοκληρωθεί η δοκιμή.
- Ο όγκος των δεδομένων που στέλνουν τα εργαλεία στην σουίτα δοκιμών και που επιστρέφει.
- Το πλήθος των ευπαθειών.
- Τα ψευδώς θετικά αποτελέσματα.
- Τα χαρακτηριστικά των εργαλείων αυτών και

θα εξαχθούν συμπεράσματα για περαιτέρω ανάλυση.

1.3 Επισκόπηση Μεθοδολογίας

Η μεταπτυχιακή διατριβή θα επικεντρωθεί στην αξιολόγηση έξι εργαλείων διείσδυσης. Θα παρουσιαστούν τα χαρακτηριστικά τους, οι δυνατότητες και οι περιορισμοί τους. Τα εργαλεία που χρησιμοποιήθηκαν για αυτή την έρευνα, επιλέχθηκαν με βάση τον αριθμό και την ποικιλία των τρωτών σημείων που διερευνούν, τον ανοιχτό κώδικα τους, την σύγχρονη τεχνολογία τους και ότι έχουν γίνει αντικείμενο μελέτης σε μια τουλάχιστον δημοσιευμένη εργασία.

Γι τον σκοπό αυτό, θα δημιουργηθεί ένα εργαστήριο, θα διεξαχθούν δοκιμές διείσδυσης και έλεγχου των ευπαθειών σε σουίτα δοκίμων (testbed) με γνώστες ευπάθειες για να αξιολογηθεί η απόδοση τους.

Η δομή της μεταπτυχιακής διατριβής αναπτύσσεται παρακάτω:

Στο δεύτερο κεφάλαιο θα περιγραφεί η τυπική δομή μιας εφαρμογής ιστού, αναλύεται τι είναι μια δοκιμή διείσδυσης, ποιοι οι τύποι των δοκιμών, πότε πρέπει να εφαρμόζεται, με ποια εργαλεία και μεθόδους σε ποιες περιοχές και επικεντρωνόμαστε στις δοκιμές διείσδυσης των εφαρμογών ιστού. Ολοκληρώνουμε το κεφάλαιο αυτό, με τις δέκα σημαντικότερες ευπάθειες

των εφαρμογών ιστού σύμφωνα με τον οργανισμό OWASP παραθέτοντας παραδείγματα κώδικα, μηχανισμούς επίθεσης και τρόπους αντιμετώπισης.

Στο τρίτο κεφάλαιο περιγράφεται η λειτουργία των αυτοματοποιημένων σαρωτών ιστού, τα πλεονεκτήματα και τα μειονεκτήματα τους και αναλύονται τα χαρακτηριστικά τεσσάρων εμπορικών πακέτων και έξι σαρωτών ανοιχτού κώδικα που θα χρησιμοποιηθούν στο εργαστήριο.

Στο τέταρτο κεφάλαιο περιγράφεται η υλικοτεχνική υποδομή του εργαστήριου που δημιουργήθηκε για εύρεση ευπαθειών στην πλατφόρμα δοκιμών WackoPicko με γνώστες και δημοσιευμένες ευπάθειες. Παρατίθενται τα αποτελέσματα των δοκιμών με πίνακες και διαγράμματα.

Στο πέμπτο κεφάλαιο γίνεται αξιολόγηση των αποτελέσματα των ελέγχων και τέλος προτείνεται υλικό για μελλοντική έρευνα.

Κεφάλαιο 2
Βιβλιογραφική Ανασκόπηση

Στο κεφάλαιο αυτό περιγράφεται η λειτουργία μιας Εφαρμογής Ιστού, εξετάζεται τι είναι μια δοκιμή διείσδυσης, οι τύποι των δοκιμών, τα εργαλεία, οι μέθοδοι και σε ποιες περιοχές εφαρμόζεται, δίνοντας έμφαση στις δοκιμές διείσδυσης των εφαρμογών ιστού. Τέλος αναλύονται οι δέκα σημαντικότερες ευπάθειες των Εφαρμογών Ιστού σύμφωνα με τον οργανισμό OWASP παραθέτοντας παραδείγματα κώδικα, μηχανισμούς επίθεσης και προτείνονται τρόποι για την αντιμετώπισης τους.

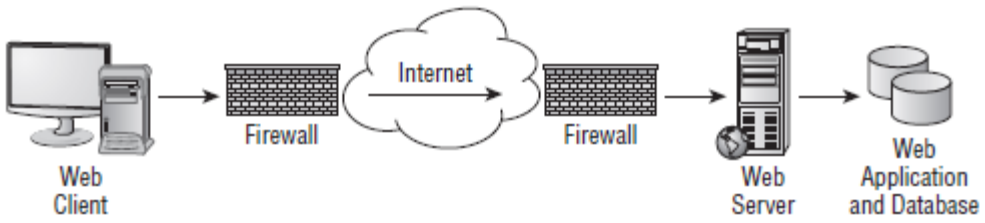
2.1 Εφαρμογές Ιστού

Οι διακομιστές ιστού και οι εφαρμογές ιστού έχουν πολύ υψηλή πιθανότητα να εκτεθούν. Ο κύριος λόγος είναι ότι τα συστήματα που εκτελούν λογισμικό διακομιστή ιστού πρέπει να είναι διαθέσιμα στο διαδίκτυο. Ο διακομιστής ιστού δεν μπορεί να απομονωθεί πλήρως και να είναι διαθέσιμος μόνο στους νόμιμους χρήστες. Μόλις ένας διακομιστής εκτεθεί, το σύστημα μπορεί να παρέχει στους επιτιθεμένους μια πόρτα στο δίκτυο. Όχι μόνο το λογισμικό διακομιστή αλλά και οι εφαρμογές που εκτελούνται στον διακομιστή είναι ανοικτές σε επίθεση και μπορούν να εκμεταλλευτούν.

Οι πληροφορίες που γίνονται στόχος σε έναν διακομιστή συνήθως βρίσκονται σε μια βάση δεδομένων στο διακομιστή ιστού. Αυτή η βάση δεδομένων είναι προσβάσιμη μέσω μιας διαδικτυακής εφαρμογής.

Οι διακομιστές χρησιμοποιούν πρωτόκολλο μεταφοράς Hypertext Transfer Protocol (HTTP) και Hypertext Transfer Protocol Secure (HTTPS) για να επιτρέπουν σε περιηγητές ιστού να συνδέονται σε αυτές, να τις προβάλλουν και να φορτώνουν αρχεία. Το HTTP είναι ένα πρωτόκολλο εφαρμογής επιπέδου στη στοίβα TCP/IP. Τα HTTP και HTTPS είναι πρωτόκολλα που χρησιμοποιούνται από τους περιηγητές και που έχουν πρόσβαση σε ιστοσελίδες που διαμένουν σε διακομιστές ιστού στο Διαδίκτυο. Η γλώσσα HTML (Hypertext Markup Language) χρησιμοποιείται για τη δημιουργία ιστοσελίδων και επιτρέπει την αναπαραγωγή αυτών των σελίδων.

Το πρωτόκολλο HTTP λειτουργεί όπως φαίνεται στο σχήμα 2.1 και περιγράφεται παρακάτω:



Σχήμα 2.1: Λειτουργία Πρωτοκόλλου HTTP.

- Ο υπολογιστής που έχει πρόσβαση στο διαδίκτυο αρχικά ανοίγει μια σύνδεση με τη διεύθυνση IP του εξυπηρετητή ιστού χρησιμοποιώντας τη θύρα TCP 80.
- Ο διακομιστής περιμένει ένα αίτημα GET από τον περιηγητή που ζητάει την αρχική σελίδα
- Ο διακομιστής ανταποκρίνεται με τον κώδικα HTML για την αρχική σελίδα
- Ο περιηγητής επεξεργάζεται τον κώδικα HTML και το πρόγραμμα περιήγησης παρέχει την ιστοσελίδα στον πελάτη.

Η κατανόηση του τρόπου με τον οποίο οι διακομιστές ιστού λειτουργούν είναι σημαντικό μέρος της ασφάλειας εφαρμογών ιστού. Αυτό περιλαμβάνει τη γνώση των τρωτών σημείων τους, καθώς και την κατανόηση των τύπων επιθέσεων που μπορεί να χρησιμοποιήσει ένας επιτιθέμενος [43].

2.2 Δοκιμές Διείσδυσης

Σύμφωνα με το τεχνικό εγχειρίδιο του NIST [04], ο έλεγχος διείσδυσης είναι μια δοκιμή ασφαλείας στην οποία ο αξιολογητής μιμείται πραγματικές επιθέσεις που αναγνωρίζουν μεθόδους που παρακάμπτουν χαρακτηριστικά ασφαλείας μιας εφαρμογής, ενός συστήματος ή δικτύου. Συχνά πραγματοποιούν πραγματικές επιθέσεις σε πραγματικά συστήματα και δεδομένα με εργαλεία και τεχνικές που χρησιμοποιούν οι εισβολείς. Οι δοκιμές διεισδύσεις αναζητούν συνδυασμούς ευπαθειών σε ένα ή περισσότερα συστήματα που μπορούν να χρησιμοποιηθούν για να έχουν ως το δυνατόν περισσότερη πρόσβαση από μόνο μια ευπάθεια.

2.2.1 Τύποι Ελέγχων

Τα εργαλεία που χρησιμοποιούνται για την ανίχνευση ευπαθειών χωρίζονται σε τρεις κατηγορίες με βάση την πληροφορία που παρέχεται για την υπό εξέταση εφαρμογή, σε εργαλεία λευκού κουτιού (white-box) όπου έχουμε πλήρη ενημέρωση για το υπό εξέταση σύστημα, σε μαύρου κουτιού (black-box) όπου δεν έχουμε καμία γνώση για το υπό εξέταση σύστημα και σε γκρι κουτιού (grey-box) που είναι μια ενδιάμεση κατάσταση. Στον πινάκα 2.1 υπάρχει σύγκριση των χαρακτηριστικών των τύπων.

	Μαύρου κουτιού	Λευκού κουτιού	Γκρι κουτιού
Πληροφορίες του Στόχου	Πληροφορίες σχετικά με τον στόχο ή την πρόσβαση στους πόρους του στόχου.	Πλήρη πληροφορία και πρόσβαση στους πόρους του στόχου	Μερική πληροφορία και μερική πρόσβαση στους πόρους του στόχου
Φύση της δοκιμής	Δοκιμή αποδοχής χρηστών	Διεξάγονται μόνο από προγραμματιστές και τους δοκιμαστές διείσδυσης	Δοκιμή αποδοχής χρηστών
Χρόνος και προσπάθεια	Εξαιρετικά εξαντλητικός και χρονοβόρος	Λιγότερο εξαντλητική και χρονοβόρα	Ενδιάμεσα των προηγούμενων
Βαθμός ανάλυσης της δοκιμής	Χαμηλή ανάλυση	Υψηλή ανάλυση	Μεσαία ανάλυση
Βασικές αρχές	Ο σχεδιασμός της δόκιμης είναι πλήρως βασισμένος σε εξωτερικές εξαιρέσεις εφόσον η εσωτερική συμπεριφορά των συστημάτων παραμένουν άγνωστα	Η εσωτερική συμπεριφορά του συστήματος είναι πλήρως γνωστή και ο σχεδιασμός της δόκιμης βασίζεται σε εσωτερικές και εξωτερικές εξαιρέσεις	Ο σχεδιασμός της δόκιμης είναι βασίζεται σε διαγράμματα βάσεων, δομές δεδομένων και εσωτερικές καταστάσεις των συστημάτων
Εύρος της δοκιμής	Μπορούν να ελεγχθούν μόνο με την μέθοδο δοκιμής και σφάλματος και μπορεί να ελεγχθούν τα εξωτερικά όρια	Μπορούν να ελεγχτούν δεδομένα και εσωτερικά όρια αλλά όχι εξωτερικά όρια	Εσωτερικά και εξωτερικά όρια υπερχείλισης δεδομένων μπορούν να ελεγχτούν
Περιορισμοί	Δεν είναι κατάλληλο για δοκιμές αλγορίθμων	Κατάλληλο για όλα δεν υπάρχουν περιορισμοί	Δεν είναι κατάλληλο για δοκιμές αλγορίθμων

Πίνακας 2.1: Πίνακας Σύγκρισης Δοκιμών Μαύρου, Λευκού και Γκρι Κουτιού

2.2.2 Περιοχές Ελέγχου Διείσδυσης.

Οι σημαντικότερες περιοχές όπου διεξάγονται δοκιμές διείσδυσης [48] είναι οι εξής:

- Δοκιμές διείσδυσης δικτύου
- Δοκιμές διείσδυσης Εφαρμογών
- Δοκιμές διείσδυσης Κοινωνικής Μηχανικής
- Δοκιμές διείσδυσης Εφαρμογών ιστού

2.2.3 Δοκιμές Διείσδυσης Δικτύου

Η δόκιμη διείσδυσης αναγνωρίζει τις τρύπες ασφάλειας που έχουν σχέση με την σχεδίαση, την κατασκευή και την λειτουργία του δικτυού που ελέγχεται. Εξετάζονται όλες οι συσκευές δικτύου όπως οι μεταγωγείς δικτύου, οι δρομολογητές, το ασύρματο δίκτυο και οτιδήποτε μπορεί να αποτελέσει σημείο εισόδου από έναν επιτιθέμενο. Τα εργαλεία που χρησιμοποιούνται για τον σκοπό αυτό είναι:

Α/Α	Εργαλείο	Λειτουργικό σύστημα	Πηγή
1	Nmap	Linux, Unix, Mac OS X, Windows	http://www.nmap.org/
2	Hping	Linux, Unix, Mac OS X, Windows	http://www.hping.org/
3	Nessus (Personal Edition)	Linux, Unix, Mac OS X, Windows	http://www.tenable.com/products/nessus/
4	Metasploit (Community Edition)	Linux, Unix, Mac OS X, Windows	http://www.rapid7.com/products/metasploit/download.jsp
5	P0f	Linux, Unix, Mac OS X, Windows	http://www.net-security.org/software.php?id=164
6	SuperScan	Windows	http://www.mcafee.com/us/downloads/free-tools/superscan.aspx/
7	Xprobe2	Linux, Unix,	http://www.net-security.org/software.php?id=231
8	Httprint	Linux, Unix, Mac OS X, Windows	http://net-square.com/httprint/
9	Brutus	Windows	http://download.cnet.com/Brutus/3000-2344_4-10455770.html/

Πίνακας 2.2: Πίνακας Εργαλείων Διείσδυσης Δικτύου

2.2.4 Δοκιμές Διείσδυσης Εφαρμογών

Στην δοκιμή ελέγχονται οι εφαρμογές για ευπάθειες μη εξουσιοδοτημένης πρόσβασης που θα οδηγήσει σε απώλεια δεδομένων. Κυρίως γίνεται έλεγχος για ευπάθειες του κώδικα σε εφαρμογές, οι οποίες δεν έχουν ελεγχθεί για θέματα ασφαλείας από τους προγραμματιστές.

Α/Α	Εργαλείο	Λειτουργικό σύστημα	Πηγή
1	Flawfinder	Linux, Unix, Mac OS X, Windows	https://www.dwheeler.com/flawfinder/
2	Findbugs	Linux, Unix, Windows	http://findbugs.sourceforge.net/
3	Fxcop	Windows	https://www.microsoft.com/en-us/download/details.aspx?id=8279
4	Pychecker	Linux, Unix, Windows	http://pychecker.sourceforge.net/

Πίνακας 2.3: Πίνακας Εργαλείων Διείσδυσης Εφαρμογών

2.2.5 Δοκιμές Διείσδυσης Κοινωνικής Μηχανικής

Στις δοκιμές κοινωνικής μηχανικής ελέγχεται ολόκληρος ο κύκλος των εργασιών ενός οργανισμού με επίκεντρο τον ανθρώπινο παράγοντα. Συλλέγονται πληροφορίες για τους εργαζόμενους, τα συστήματα και τον τρόπο λειτουργίας αυτών, με στόχο τη μη εξουσιοδοτημένη πρόσβαση.

Α/Α	Εργαλείο	Λειτουργικό σύστημα	Πηγή
1	SET	Linux, Windows	https://www.trustedsec.com/social-engineer-toolkit/
2	MALTEGO	Linux, Unix, Mac OS X, Windows	https://www.paterva.com/web7/

Πίνακας 2.4: Πίνακας Εργαλείων Κοινωνικής Μηχανικής

2.2.6 Δοκιμές Διείσδυσης Εφαρμογών Ιστού

Οι εφαρμογές ιστού σε αντίθεση με άλλα αγαθά που προστατεύουμε, όπως εσωτερικό δίκτυο, τοπικές εφαρμογές χρήστη, είναι εκτιθέμενα στο διαδίκτυο από κάθε είδους επιθέσεις. Οι ευπάθειες που πρέπει να προστατεύουμε μπορούν να διακριθούν σε επίπεδο διακομιστή και εφαρμογών ιστού. Στο επίπεδο του διακομιστή, θα πρέπει να ενημερωθούν οι ευπάθειες του λειτουργικού συστήματος των διακομιστών. Οι εφαρμογές ιστού θα πρέπει να προστατεύουν από μια πληθώρα ευπαθειών όπως τις μη ασφαλείς επικυρώσεις εισόδου, τα λογικά σφάλματα, τα δυαδικά ελαττώματα της εφαρμογής. Οι δοκιμές διείσδυσης μπορούν να πραγματοποιηθούν με δυο τρόπους, είτε με στατική ανάλυση κώδικα (δοκιμή διείσδυσης λευκού κουτιού) είτε με αυτοματοποιημένο τρόπο (δόκιμη διείσδυσης μαύρου κουτιού). Οι προγραμματιστές εφαρμογών ιστού επιλέγουν τις περισσότερες φορές τα εργαλεία μαύρου κουτιού, καθώς με τον τρόπο αυτόν αξιολογούν αυτόματα την ασφάλεια των εφαρμογών ιστού με ελάχιστη ή καμία ανθρώπινη παρέμβαση.

Α/Α	Εργαλείο	Λειτουργικό σύστημα	Πηγή
1	W3AF	Linux, Unix, Mac OS X, Windows	http://www.w3af.org/
2	Nikto	Linux, Unix, Mac OS X, Windows	http://www.cirt.net/nikto2
3	OWASP ZAP	Linux, Unix, Mac OS X, Windows	https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project
4	Skipfish	Linux, Unix, Mac OS X, Windows	https://github.com/spinkham/skipfish
5	VEGA	Linux, Unix, Mac OS X, Windows	https://subgraph.com/vega/
6	Arachni	Linux, Unix, Mac OS X, Windows	http://www.arachni-scanner.com/download/
7	Grabber	Linux, Unix, Mac OS X, Windows	http://rgaucher.info/beta/grabber/
8	Wapiti	Linux, Unix, Mac OS X, Windows	http://wapiti.sourceforge.net/
9	WebScarab	Linux, Unix, Mac OS X, Windows	https://www.owasp.org/index.php/Category:OWASP_WebScarab_Project

Πίνακας 2.5: Πίνακας Εργαλείων Διείσδυσης Εφαρμογών Ιστού

2.2.7 Πλεονεκτήματα Ελέγχων.

Τα σημαντικότερα πλεονεκτήματα για την διεξαγωγή δοκιμών διείσδυσης είναι [22]:

Ευφυής διαχείριση ευπαθειών. Ο έλεγχος διείσδυσης παρέχει μια λεπτομερή ενημέρωση για υπαρκτές απειλές ασφαλείας, λόγω του ότι διεξάγοντας μια δοκιμή διείσδυσης γίνεται αναγνώριση των ευπαθειών αλλά και της κρισιμότητας τους. Ο οργανισμός εφαρμόζει την πολιτική ασφαλείας του με βάσει τα ευρημάτων που πρόεκυψαν από τον έλεγχο και διαχειρίζεται τους πόρους που διαθέτει για την ασφάλεια του πιο αποδοτικά

Αποφυγή του χρόνου μη διαθεσιμότητας δικτύου. Η αποκατάσταση από μια παραβίαση ασφαλείας μπορεί να κοστίσει σε έναν οργανισμό χρήματα, απώλεια πελατών, μείωση της παραγωγικότητας των εργαζομένων και του κύκλου εργασιών. Με τη διεξαγωγή ελέγχων αναγνωρίζονται οι αδυναμίες ασφαλείας και μειώνεται το ρίσκο να εμφανιστούν παραβιάσεις ασφαλείας που θα οδηγήσουν σε οικονομικές απώλειες.

Αποφυγή προστίμων από ρυθμιστικούς κανόνες. Η δοκιμές διείσδυσης βοηθούν έναν οργανισμό να συμμορφώνεται με τους ρυθμιστικούς κανόνες που θέτουν εθνικοί και υπερεθνικοί νόμοι (Ευρωπαϊκή Ένωση, Ηνωμένες πολιτείες της Αμερικής), υιοθετώντας αναγνωρισμένες μεθοδολογίες ή δομές ελέγχων όπως του NIST, PCI-DSS, κ.α.

Διατήρηση της εταιρικής φήμης και της αφοσίωσης πελατών. Ένα περιστατικό παραβίασης της ασφάλειας αποβαίνει σε βάρος ενός οργανισμού τόσο σε οικονομικό επίπεδο όσο και σε πελατειακό.

2.2.8 Περιορισμοί των Ελέγχων Διείσδυσης

Οι δοκιμές διείσδυσης βοηθούν την ασφάλεια αλλά υπάρχουν και περιορισμοί [23]:

- Στην ικανότητα των ελεγκτών που διενεργούν τους ελέγχους
- Στην ικανότητα των εργαλείων να εντοπίζουν όλες τις ευπάθειες
- Στον χρόνο διεξαγωγής του ελέγχου
- Στο εύρος του ελέγχου
- Στην πρόσβαση που δίνεται στους ελεγκτές που διεξάγουν τον έλεγχο
- Στις μεθόδους που χρησιμοποιούν οι ελεγκτές

2.2.9 Συχνότητα Εκτέλεσης μιας Δόκιμη Διείσδυσης

Οι δοκιμές διείσδυσης πρέπει να διεξάγονται σε τακτά χρονικά διαστήματα για την συνέχιση της ασφάλειας και την ανεύρεση νέων απειλών [22]. Ο τακτικός και σχεδιασμένος έλεγχος διείσδυσης είναι επιβεβλημένος, αλλά έλεγχοι πρέπει να διεξάγονται και όταν:

- Προστίθενται νέος εξοπλισμός δικτύου ή νέες εφαρμογές
- Όταν εφαρμόζονται σημαντικές αναβαθμίσεις ή τροποποιήσεις στην υποδομή ή σε εφαρμογές
- Δημιουργούνται νέα δομές στον οργανισμό
- Όταν εφαρμόζονται αναβαθμίσεις ασφαλείας
- Όταν αλλάζει η πολιτική ασφαλείας του χρήστη.

2.2.10 Διαχείριση των Ευπαθειών

Υπάρχουν πολλά στοιχεία στη διαδικασία διαχείρισης των ευπαθειών από την ανάλυση κώδικα, την σάρωση ευπαθειών ως την δοκιμή διείσδυσης. Κάθε μια διαδικασία μειώνει το ρίσκο του δέχεται κάθε οργανισμός και παράλληλα προστατεύει τις υποδομές του οργανισμού από πραγματικές απειλές.

Η ανάλυση του κώδικα πρέπει να γίνεται όσο το δυνατόν νωρίτερα κατά την διαδικασία συρραφής του κώδικα, αν και έχει διαπιστωθεί ότι οι εφαρμογές πάσχουν από ευπάθειες κυρίως λόγω της πίεσης που ασκείται στους προγραμματιστές κατά την συγγραφή ή λόγω της έλλειψης συγγραφής ασφαλούς κώδικα.

Οι σαρωτές ευπαθειών βοηθούν τους οργανισμούς να συλλέγουν πληροφορίες με όλες τις ευπάθειες που μπορεί να έχει ένα σύστημα και παράγονται κυρίως από τους χρήστες που τους χρησιμοποιούν. Το μεγαλύτερο πρόβλημα τους είναι τα ψευδώς θετικά αποτελέσματα των ελέγχων, που συνήθως είναι αρκετά υψηλά.

Οι δοκιμές διείσδυσης επιτρέπουν στους οργανισμούς να αξιολογούν και να εκμεταλλεύονται ευπάθειες, επιτρέποντας τους να αξιολογούν τις ικανότητες των συστημάτων ασφαλείας τους εφόσον λειτουργούν ως κακόβουλος χρήστης. Ο έλεγχος διείσδυσης καθορίζει την αποτελεσματικότητα των συστημάτων ασφαλείας ενός οργανισμού. Με την ανάλυση των μηχανισμών άμυνας προκύπτει εάν όντως οι μηχανισμοί αυτοί μπορούν να προστατέψουν έναν οργανισμό από επιθέσεις.

Το σημαντικότερο κατά την διεξαγωγή ενός ελέγχου διείσδυσης είναι ότι τα αποτελέσματα που προκύπτουν επιτρέπουν στα στελέχη της πληροφορικής να ορίσουν τα κρίσιμα ζητήματα ασφαλείας και να προτεραιοποιήσουν τις προσπάθειες για αποκατάσταση των ζητημάτων ασφαλείας [23].

Ο πίνακας 2.6 συγκρίνει τα χαρακτηριστικά των σαρωτών ευπάθειας και των δοκιμών διείσδυσης.

	Σαρωτές Ευπαθειών	Δοκιμές Διείσδυσης
Πεδίο *δοκιμών*	Σαρώνονται όλα τα πιθανά δίκτυα	Αναγνωρίζονται ευπάθειες και καθορίζεται εάν όντως μπορούν να εκμεταλλευτούν
Σύνδεση Ευπαθειών	Κατηγοριοποίηση των ευπαθειών με βάση πρότυπα, θεωρητικές πληροφορίες που δεν έχουν παραμετροποιηθεί για την πλατφόρμα δοκιμών που θα διεξαχθεί ο έλεγχος	Δοκιμάζονται ευπάθειες σε συγκεκριμένους πόρους δικτύου, επιτρέπει προτεραιοποίηση προσπαθειών αποκατάστασης
Αξιοποίηση των αποτελεσμάτων των δοκιμών	Παρέχει ψευδώς θετικά αποτελέσματα, αναγνωρίζουν ευπάθειες που δεν μπορούν να εκμεταλλευτούν.	Εκμετάλλευση ευπαθειών και αναγνώριση μόνο αυτών που μπορούν να γίνουν πραγματικές απειλές στους πόρους του δικτύου.
Χρησιμότητα των δοκιμών σύνδεσης	Δεν δρομολογεί συνδέσεις μεταξύ δικτύων και εφαρμογών	Εκμεταλλεύονται την σχέση εμπιστοσύνης μεταξύ δικτύων, εφαρμογών και τελικών χρηστών για να αποδείξει πραγματικές επιθέσεις
Δοκιμές σε άλλες συσκευές ασφαλείας	Δεν προσομοιώνει επιθέσεις για να δοκιμάζει IDS, IPS, τείχος προστασίας ή άλλες τεχνολογίες ασφαλείας και πολιτικές τελικού χρήστη.	Παρουσιάζουν πραγματικές επιθέσεις για να καθοριστεί εάν άλλες επενδύσεις ασφαλείας λειτουργούν αποτελεσματικά και οι χρήστες συμμορφώνονται με τους κανόνες ασφαλείας και τους κανονισμούς του οργανισμού.
Αξιολόγηση εκτίμησης κινδύνου	Αναγνωρίζει μόνο ενημερώσεις που δεν έχουν εγκατασταθεί, ή λανθασμένες παραμετροποιήσεις κάνοντας αδύνατο τον αποτελεσματικό υπολογισμό του ρίσκου ασφαλείας	Γίνονται ασφαλείς μιμήσεις των ενεργειών που έναν κακόβουλος χρήστης θα εκτελούσε παρέχοντας αξιολογήσεις ρίσκων που βασίζονται σε πραγματικές απειλές.

Πίνακας 2.6: Συγκριτικός Πίνακας Σαρωτών Ευπαθειών και Δοκιμών Διείσδυσης

2.2.11 Χειροκίνητοι Ή Αυτοματοποιημένοι Έλεγχοι Διείσδυσης.

Μέχρι πρόσφατα οι έλεγχοι διείσδυσης γίνονταν από εξειδικευμένο προσωπικό που διεξήγαγε μια πολύ εξειδικευμένη χειροκίνητη διαδικασία. Η ομάδα που θα διενεργούσε τον έλεγχο έγραφε τον δικό της κώδικα και διεξήγαγε περίπλοκες εργασίες. Επιπλέον, η ομάδα είναι δαπανηρή λόγω της ειδίκευση της με αποτέλεσμα πολλοί οργανισμοί να μην έχουν τους πόρους για να διεξάγουν ελέγχους διείσδυσης.

Η χρήση δόκιμων διείσδυσης με αυτοματοποιημένα εργαλεία έκανε πιο απλή υπόθεση την διενέργεια ελέγχου λόγω του ότι γιατί τα εργαλεία που δημιουργήθηκαν έγιναν από τεχνικούς ασφαλείας και προγραμματιστές, οι οποίοι δημιούργησαν πακέτα λογισμικού ικανά να βρίσκουν προηγμένες ευπάθειες ασφαλείας σε ευκολόχρηστο περιβάλλον χωρίς να απαιτούνται εξειδικευμένες γνώσεις. Η χρήση αυτοματοποιημένων εργαλείων για δοκιμές διείσδυσης παρέχει μια ολοκληρωμένη άποψη για την ασφάλεια ενός οργανισμού [03].

Τα χαρακτηριστικά των χειροκίνητων και αυτοματοποιημένων ελέγχων περιγράφονται στον πίνακα 2.7.

	Χειροκίνητοι έλεγχοι διείσδυσης	Αυτοματοποιημένοι έλεγχοι διείσδυσης
Διαδικασία Δόκιμης	Είναι χρονοβόρα, με σφάλματα και χωρίς προδιαγραφές ποιότητας. Απαιτούνται διαφορετικά εργαλεία. Τα αποτελέσματα ποικίλουν από έλεγχο σε έλεγχο. Μεγάλο το κόστος του ελέγχου γιατί απαιτεί πολύ εξειδικευμένο προσωπικό ασφαλείας.	Γρήγορο, εύκολο και με ελαχιστοποίηση των σφαλμάτων. Με προδιαγραφές για να παράγει επαναλήψημα αποτελέσματα. Εύκολο στην χρήση με αναφορές.
Τροποποίηση Δικτύου	Συχνά απαιτείται	Δεν επηρεάζει τα συστήματα
Ανάπτυξη και διαχείριση εκμετάλλευσης ευπαθειών	Η διαχείριση της βάσης εκμετάλλευσης ευπαθειών απαιτεί σημαντική εξειδίκευση	Ο προμηθευτής του εργαλείου συντηρεί όλα προγράμματα εκμετάλλευσης ευπαθειών. Τα προγράμματα αυτά μπορούν να εκτελεστούν σε διαφορετικές πλατφόρμες.
Αλλαγές στις ρυθμίσεις	Όλες οι αλλαγές που έγιναν για να πραγματοποιηθεί η δοκιμή πρέπει να επαλειφθούν.	Οι αλλαγές διορθώνονται με ευκολία χωρίς να αφήνουν ανοιχτές πόρτες ασφαλείας.
Κλιμάκωση επίθεσης.	Απαιτεί τροποποίηση των	Ο κώδικας δεν

	συστημάτων γιατί ο κώδικας πρέπει να μεταφορτωθεί στο εκτεθειμένο μηχάνημα.	μεταφορτώνεται και οι δοκιμές γίνονται απομακρυσμένα.
Αναφορές	Όλες οι αναφορές γίνονται χειρόγραφα	Οι αναφορές γίνονται αυτόματα και είναι παραμετροποιήσιμες
Σύνδεση- Έλεγχος	Αργή και συνήθως ανακριβή διαδικασία	Αυτόματες καταγραφές με λεπτομέρειες
Εκπαίδευση	Όσοι θα διεξάγουν τους ελέγχους θα πρέπει να μάθουν μεθόδους που δημιουργήθηκαν μόνο για τον έλεγχο αυτό.	Οι χρήστες μπορούν να μάθουν να τα εγκαθιστούν με μια μέρα

Πίνακας 2.7: Συγκριτικός Πίνακας Χειροκίνητων και Αυτοματοποιημένων Δοκιμών Διείσδυσης

2.2.12 Μέθοδοι και Τεχνικές

Μια δόκιμη έλεγχου διείσδυσης για να είναι επιτυχημένη πρέπει να βασιστεί σε μια μεθοδολογία για να υλοποιηθεί. Υπάρχουν πολλές μεθοδολογίες ή τεχνικές που έχουν προταθεί από άρθρα, εργασίες ή οργανισμούς. Οι φάσεις εκτέλεσης διαφέρουν γιατί κάθε πρόταση επικεντρώνεται σε διαφορετικό αντικείμενο, σκοπό αλλά και σε διαφορετικό τρόπο υλοποίησης. Έτσι κάποιες μεθοδολογίες επικεντρώνονται στην φυσική ασφάλεια, στην ασφάλεια δικτύου ή στην ασφάλεια των εφαρμογών ιστού. Γι' αυτό συνίσταται πριν επιλέγει η κατάλληλη μεθοδολογία να κατανοηθεί ο σκοπός και τα χαρακτηριστικά της. Όλες οι μεθοδολογίες έχουν τρις τουλάχιστον φάσεις, τη συγκέντρωση πληροφοριών (information gathering) ή αναγνώριση (reconnaissance), την ανάλυση ευπαθειών (vulnerability analysis) και την εκμετάλλευση ευπαθειών (exploitation).

Η πραγματοποίηση ενός έλεγχου διείσδυσης πρέπει να γίνεται από ειδικευμένο και εξουσιοδοτημένο προσωπικό. Πρέπει να τηρούνται οι κανόνες που έχουν συμφωνηθεί από τις δυο πλευρές (από το προσωπικό που διεξάγει τον έλεγχο και τον οργανισμό που δέχεται τον έλεγχο). Τέλος, θα πρέπει να τηρείται η σειρά των φάσεων έλεγχου και να ολοκληρώνεται κάθε μια από αυτές.

Η μεθοδολογία δοκιμών διεισδύσεις του National Institute of Standards and Technology (NIST) παρουσιάζει τις δοκιμές, τις μεθόδους εξέτασης και τους ελέγχους που ένας οργανισμός μπορεί να χρησιμοποιεί ως μέρος μιας αξιολόγησης και προσφέρει στους εκτιμητές των ελέγχων τις πιθανές επιπτώσεις στα συστήματα και το δίκτυο τους. Η μεθοδολογία δοκιμών διεισδύσης του NIST αποτελείται από τέσσερις φάσεις: σχεδιασμός, ανακάλυψη, επίθεση και αναφορά.

Το OSSTMM είναι μια ανοιχτού κώδικα μεθοδολογία δοκιμών ασφάλειας που παρουσιάστηκε το 2000 από το Institute for Security and Open Methodologies (ISECOM). Το OSSTMM έχει τα πλεονεκτήματα της άδειας ανοιχτού κώδικα αλλά η τέταρτη και τελευταία έκδοση δεν διατίθεται δωρεάν. Η τρίτη έκδοση της μεθοδολογίας OSSTMM ενσωματώνει λειτουργίες και κανάλια, τα οποία αντιπροσωπεύουν διαφορετικές περιοχές τομέα. Το OSSTMM είναι αρχικά μια μεθοδολογία ελέγχου και επομένως δεν είναι πλήρης όπως η ISSAF και δεν παρέχει εργαλεία ή μεθόδους για την κάλυψη των τομέων αλλά θέτει τους ρυθμιστικούς κανόνες για εταιρικά περιουσιακά στοιχεία.

Το OWASP είναι ένας μη κερδοσκοπικός οργανισμός που επικεντρώνεται στην βελτίωση της ασφάλειας των λογισμικών. Το OWASP παρέχει εργαλεία, οδηγούς, και μεθοδολογίες δοκιμών για ασφάλεια κυβερνοχώρου με άδειες ανοιχτού λογισμικού και ειδικότερα τον οδηγό OTG

(OWASP Testing Guide).Ο οδηγός OTG χωρίζεται σε τρία τμήματα, στην ανάπτυξη εφαρμογών ιστού, στην μεθοδολογία δοκιμών και στην αναφορά. Αναφέρεται σε όλον το κύκλο ανάπτυξης ενός λογισμικού εστιάζει ,όμως, στις Εφαρμογές Ιστού.

Το ISSAF είναι ένα framework δοκιμών διείσδυσης ανοιχτού κώδικα που δημιουργήθηκε από την Open Information Systems Security Group (OISSG). Το ISSAF περιγράφεται σαν ένα framework που ενσωματώνει πολλαπλές μεθοδολογίες. Το ISSAF προσπαθεί να καλύψει όλους τους πιθανούς τομείς μιας δοκιμής διείσδυσης από την αρχή έως την ολοκλήρωση. Η μεθοδολογία του ελέγχου διείσδυσης χωρίζεται σε τρις μεγάλες φάσεις, τον σχεδιασμό και προετοιμασία, αξιολόγηση και αναφορά και καθαρισμός. Ένα πλεονέκτημα του ISSAF είναι οι σχέσεις μεταξύ των εργασιών και τα εργαλεία που χρησιμοποιούνται.

Η μεθοδολογία δοκιμών διείσδυσης του Penetration Testing Execution Standard (PTES) είναι ένα πρότυπο που δημιουργήθηκε το 2009 από τον Nickerson. Η μεθοδολογία PTES περιλαμβάνει αλληλεπίδραση πριν την συμπλοκή, συλλογή πληροφοριών, μοντέλα απειλών, ανάλυση ευπαθειών, εκμετάλλευση ευπαθειών και αναφορές. Η μεθοδολογία PTES χρησιμοποιεί ως αναφορά άλλα framework και δε δημιουργεί δικά της, όπως το OWASP, για τις δοκιμές σε εφαρμογές ιστού [29].

Ο παρακάτω πίνακας 2.8 μας δείχνει τις φάσεις των πέντε μεθοδολογιών που περιγράφηκαν.

NIST	PTES	OSSTM	ISSAF(2006)	OWASP
Σχεδιασμός	Συγκέντρωση πληροφοριών	Συλλογή πληροφοριών	Σχεδιασμός και προετοιμασία	Συλλογή πληροφοριών
Ανακάλυψη	Μοντελοποίηση απειλών	Φάση συμπεράσματος	Αξιολόγηση	Παραμετροποίηση και Ανάπτυξη
Επίθεση	Ανάλυση ευπάθειας	Φάση αλληλεπίδρασης	Συλλογή πληροφοριών	Δοκιμές διαχείρισης
Αναφορά	Εκμετάλλευση	Φάση έρευνας	Χαρτογράφηση Δικτύου	Δοκιμές διαχείρισης ταυτότητας
	Δημοσίευση Εκμετάλλευσης	Φάση παρέμβασης	Αναγνώριση ευπάθειας	Δοκιμή ελέγχου ταυτότητας
	Αναφορά		Διείσδυση	Δοκιμή Εξουσιοδότησης
			Απόκτηση πρόσβαση και κλιμάκωση	Δοκιμές διαχείρισης συνόδου

			προνομίων	
			Περαιτέρω απαρίθμηση	Έλεγχος επικύρωσης εισόδου
			Έκθεση απομακρυσμένων χρηστών /ιστότοπων	Διαχείριση σφαλμάτων
			Διατήρηση της πρόσβασης	Κρυπτογράφηση
			Κάλυψη ιχνων	Δοκιμές Επιχειρηματικής Λογικής
			Αναφορά, καθαρισμός και καταστροφή αντικείμενων	Δοκιμή πλευράς πελάτη

Πίνακας 2.8: Πίνακας Μεθοδολογιών Δοκιμών Διείσδυσης (Φάσεις)

2.3 Ευπάθειες Εφαρμογών Ιστού

Η ανίχνευση των ευπαθειών των εφαρμογών ιστού έχουν οργανωθεί και ταξινομηθεί από διαφόρους οργανισμούς και εταιρίες. Οι πιο σημαντικοί είναι:

- Η WASC Threat Classification, από το WASC το 2009, και περιλαμβάνουν μια λίστα με προβλήματα ασφαλείας που δημιουργούνται σε εφαρμογές ιστού. Ο κατάλογος των προβλημάτων προέρχεται κυρίως από το WASC Classification of Threat και περιλαμβάνει 55 αντικείμενα που ομαδοποιούνται σε διάφορες κατηγορίες.

- Οι SANS και MITRE ανέπτυξαν το CWE / SANS TOP 25, μια λίστα με τα πιο επικίνδυνα σφάλματα λογισμικού. Είναι ένα υποσύνολο της CWE και, όπως και το CWE ταξινομεί όλα τα είδη ευπάθειας του λογισμικού. Αυτή η ταξινόμηση περιλαμβάνει παραδείγματα και μεθόδους ανίχνευσης.

- Μια άλλη ταξινόμηση που περιλαμβάνει όλα τα είδη ευπάθειας του λογισμικού είναι η Common Attack Pattern Enumeration and Classification. Αυτή η λίστα περιέχει πρότυπα επίθεσης και ενημερώνεται από την εταιρεία MITRE Corporation. Η έκδοση 2.1 κυκλοφόρησε το 2013. Περιλαμβάνει επίσης παραδείγματα και περιγραφές εκτέλεσης επιθέσεων [28].

- Το OWASP Top 10, παρέχεται από το Open Project Security Project (OWASP) και περιέχει τους κορυφαίους 10 πιο σημαντικούς κινδύνους ασφάλειας εφαρμογών Ιστού, Η τελευταία έκδοση του OWASP Top 10 κυκλοφόρησε το 2013 η επόμενη αναθεωρημένη έκδοση αναμένεται να εκδοθεί τον Ιούλιο ή τον Αύγουστο του 2017. Αυτός ο κατάλογος περιλαμβάνει περιγραφές των τρωτών σημείων, συνοδευόμενων από παραδείγματα και μεθόδους για την ανίχνευσή τους και εμπεριέχει ορισμένες μεθόδους για τον μετριασμό των επιπτώσεων των τρωτών σημείων.

Σύμφωνα με τον οργανισμό OWASP οι δέκα σημαντικότερες ευπάθειες σε εφαρμογές ιστού όπως δημοσιευτήκαν το 2013 είναι [27]:

A1 CWE-929 Έκχυση (Injection)

Οι ευπάθειες έκχυσης κώδικα (Injection flaws) όπως οι SQL, OS, και LDAP συμβαίνουν όταν μη επικυρωμένα δεδομένα στέλνονται ως μέρος μιας εντολής ή ερωτήματος. Τα εχθρικά δεδομένα μπορούν να ξεγελάσουν τον διερμηνέα και να εκτελέσουν εντολές ή να προσπελάσουν δεδομένα χωρίς την κατάλληλη εξουσιοδότηση.

Παράδειγμα ευπάθειας. Ο παρακάτω κώδικας είναι ευπαθής σε επιθέσεις τύπου SSJS . Μπορεί να αποτραπεί με την χρήση της συνάρτησης parseInt().

```
//Fix for A1 -1 SSJS Injection attacks - uses alternate method to eval
var preTax = parseInt(req.body.preTax);
var afterTax = parseInt(req.body.afterTax);
var roth = parseInt(req.body.roth);
```

Μέθοδος αποτροπής. Η ευπάθεια είναι δυνατόν να αποτραπεί επικυρώνοντας την είσοδο του χρήστη στον εξυπηρετητή πριν την επεξεργασία. Η χρήση της συνάρτησης eval() θεωρείται σκόπιμο να αποφεύγεται.

A1 – 1. Server Side JS Injection. Όταν οι συναρτήσεις eval(), setTimeout(), setInterval(), Function() χρησιμοποιούνται ως είσοδοι μπορούν να εκμεταλλευτούν από τον επιτιθέμενο και να εκτελεστεί κακόβουλος κώδικας JavaScript στον εξυπηρετητή. Αυτή η ευπάθεια είναι πολύ κρίσιμη και επιβλαβής γιατί επιτρέπει στον επιτιθέμενο να εκτελέσει διαφόρους τύπους εντολών, όπως επίθεση άρνησης υπηρεσιών (DOS).

Μέθοδος αποτροπής. Η ευπάθεια είναι δυνατόν αν αποτραπεί επικυρώνοντας την είσοδο του χρήστη στον εξυπηρετητή πριν την επεξεργασία. Η χρήση των συναρτήσεων eval(), setTimeout(), setInterval(), Function() θεωρείται σκόπιμο να αποφεύγονται.

A1 – 2. SQL and NoSQL Injection. Μέθοδος αποτροπής. Οι επιθέσεις έκχυσης SQL / NoSQL μπορούν να αποτραπούν ή να μειωθεί η επίδραση τους εάν χρησιμοποιηθούν έτοιμες προτάσεις ή εάν επικυρωθεί η είσοδος ή να ελαχιστοποιηθούν τα δικαιώματα πρόσβασης του χρήστη της εφαρμογής.

A2 CWE-930 Επισφαλής Διαχείριση Δεδομένων Διαπίστευσης και Συνεδρίας (Broken Authentication and Session Management)

Σε αυτή την επίθεση ο επιτιθέμενος χρησιμοποιεί τα ελαττώματα στην συνάρτηση εξουσιοδότησης ή στην διαχείριση συνεδρίας για να υποδυθεί έναν πιστοποιημένο χρήστη. Οι προγραμματιστές κατασκευάζουν εξατομικευμένες εξουσιοδοτήσεις λύσεις αλλά συνήθως έχουν ελαττώματα, όπως στην αποσύνδεση στην μυστική λέξη, στην διαχείριση των κωδικών κ.α. Η ανεύρεση αυτών των σφαλμάτων είναι δύσκολη.

A2 - 1 Διαχείριση συνεδρίας (Session Management)

Η διαχείριση συνεδρίας είναι ένα κρίσιμο στοιχειό της ασφάλειας εφαρμογών. Οι προγραμματιστές θα πρέπει να προστατευόσουν την ταυτότητα της συνεδρίας (session id), την διάρκεια της συνεδρίας, τα δεδομένα που μεταφέρονται και τα διαπιστευτήρια του χρήστη.

Μηχανισμός επίθεσης. Εάν δεν έχει ρυθμιστεί σωστά η εφαρμογή να αποσυνδέεται μετά από μικρό χρονικό διάστημα ένας κακόβουλος χρήστης μπορεί να χρησιμοποιήσει έναν υπολογιστή που δεν αποσυνδέθηκε αλλά έκλεισε τον περιηγητή και να χρησιμοποιήσει τα διαπιστευτήρια του προηγούμενου χρήστη.

A2 - 2 Επιθέσεις πρόβλεψης κωδικών

Η δημιουργία κωδικών με κριτήρια, πολυπλοκότητα και ελάχιστο μήκος, αποτρέπει τον επιτιθέμενο να τους προβλέψει.

Μηχανισμός επίθεσης. Ο επιτιθέμενος μπορεί να εκμεταλλευτεί την χρήση αδυνάμων κωδικών με την χρήση εργαλείων που παράγουν τυχαίους κωδικούς.

Μέθοδος αποτροπής. Το μήκος του κωδικού πρέπει να είναι τουλάχιστον οκτώ χαρακτήρες. Ο κωδικός πρέπει να είναι συνδυασμός αριθμών, γραμμάτων και ειδικών χαρακτήρων. Ο συνδυασμός του μήκους και της πολυπλοκότητας αποτρέπει τις επιθέσεις ωμής βίας (brute force) και είναι δύσκολο να τους προβλέψει ο επιτιθέμενος.

A3 CWE-931 Cross Site Scripting (XSS)

Οι αδυναμίες XSS προκύπτουν όταν μια εφαρμογή αποστέλλει μη αξιόπιστα δεδομένα και τα στέλνει σε έναν περιηγητή χωρίς κατάλληλη επικύρωση. Οι αδυναμίες XSS επιτρέπουν στους επιτιθέμενους να εκτελέσουν μια δέσμη ενεργειών (script) στον περιηγητή του θύματος και να έχουν πρόσβαση σε οποιαδήποτε cookie, συνέδρια ή άλλα ευαίσθητα δεδομένα που διατηρούνται στον περιηγητή ή αναδρομολογούν τον χρήστη σε μολυσμένους ιστότοπους.

Μηχανισμός επίθεσης. Οι επιθέσεις XSS μπορούν να χωριστούν σε δυο κύριες κατηγορίες με βάση τον τρόπο που ο κακόβουλος κώδικας (payload) επηρεάζει τον περιηγητή του θύματος:

- στις αντανακλώμενες (Reflected) XSS ή μη επίμονες (non-Persistent). Σε αυτή την περίπτωση ο κακόβουλος κώδικας εγχέεται / αντανακλάται από τον διακομιστή ιστού, όπως ένα μήνυμα σφάλματος, ένα αποτέλεσμα αναζήτησης ή οποιαδήποτε άλλη απάντηση που περιλαμβάνει μερικές ή όλες τις πληροφορίες που αποστέλλονται στο διακομιστή ως μέρος του αιτήματος. Οι αντανακλαστικές επιθέσεις παραδίδονται στα θύματα μέσω άλλης διαδρομής, όπως σε μήνυμα ηλεκτρονικού ταχυδρομείου ή σε κάποια άλλη ιστοσελίδα.

- στις αποθηκευμένες (Stored) XSS ή επίμονες (Persistent). Ο κακόβουλος κώδικας αποθηκεύεται μόνιμα στους διακομιστές προορισμού, όπως σε μια βάση δεδομένων, σε ένα φόρουμ μηνυμάτων, στο αρχείο καταγραφής επισκεπτών, στο πεδίο σχολίων κλπ. Το θύμα ανακτά στη συνέχεια τον κακόβουλο κώδικα από το διακομιστή όταν ζητά τα αποθηκευμένες πληροφορίες

Μέθοδος αποτροπής. Με πιστοποίηση της εισόδου, κωδικοποίηση της εξόδου για σωστό περιεχόμενο, υλοποίηση ασφάλειας περιεχομένου και εφαρμογή κωδικοποίησης στον εξυπηρετητή και στον πελάτη.

A4 CWE-932 Επισφαλής Άμεση Αναφορά σε Οντότητες (Insecure Direct Object Reference)

Επισφαλής άμεση αναφορά σε οντότητες της εφαρμογής προκύπτει όταν ένας προγραμματιστής εκθέτει μια αναφορά σε οντότητες της εφαρμογής όπως ένα αρχείο, κατάλογο ή κλειδί βάσης δεδομένων. Χωρίς έλεγχο της πρόσβασης η άλλη προστασία οι επιτιθέμενοι μπορεί να παραποιήσουν τις αναφορές αυτές και να έχουν πρόσβαση σε μη εξουσιοδοτημένα δεδομένα.

Μηχανισμός επίθεσης. Εάν μια εφαρμογή χρησιμοποίει το πραγματικό όνομα ή κλειδί της οντότητας όταν δημιουργούν ιστοσελίδες και δεν ταυτοποιούν τον χρήστη, τότε μπορεί να έχουμε επισφαλή άμεση αναφορά σε οντότητες. Ένας επιτιθέμενος μπορεί να εκμεταλλευτεί τέτοιες αδυναμίες παραποιώντας τις τιμές των παραμέτρων. Εάν η οντότητα δεν είναι απρόβλεπτη είναι εύκολο σε έναν επιτιθέμενο να έχει πρόσβαση σε όλα τα διαθέσιμα δεδομένα.

Μέθοδος αποτροπής.. Κάθε χρήση σε άμεση αναφορά σε οντότητες από επισφαλής πηγή πρέπει να περιέχει έλεγχο πρόσβασης για να διασφαλίσει ότι ο χρήστης είναι εξουσιοδοτημένος για την αιτούμενη οντότητα. Χρήση ανά χρήστη έμμεσων αναφορών σε οντότητες. Αντί να εκθέτουμε πραγματικά κλειδιά βάσεων δεδομένων ως μέρος διευθύνσεων πρόσβασης, χρησιμοποιούμε προσωρινές έμμεσες αναφορές ανά χρήστη. Ως παράδειγμα αντί να χρησιμοποιήσουμε τα πηγαία κλειδιά της βάσης για μια λίστα έξι χρηστών, μπορούμε να χρησιμοποιήσουμε τα νούμερα από το 1 έως το 6 για κάθε χρήστη που επιλέγεται. Η εφαρμογή θα πρέπει στην συνέχεια να συνδέσει την έμμεση αναφορά που χρησιμοποιήθηκε δηλαδή τον αριθμό από το ένα έως το έξι, με το κλειδί της βάσης. Ένας επίσης τρόπος αποφυγής ευπαθειών είναι οι δοκιμές και η ανάλυση κώδικα. Οι δοκιμαστές μπορούν εύκολα να παραποιήσουν τιμές παραμέτρων για να ανακαλύψουν τέτοιες αδυναμίες. Η ανάλυση κώδικα μπορεί γρήγορα να δείξει εάν έχει επαληθευτεί η εξουσιοδότηση.

Παράδειγμα κώδικα. Η επισφαλής εφαρμογή παίρνει το id του χρήστη από το url για να αποκτήσει την πρόσβαση στο allocations.

```
var userId = req.params.userId;
allocationsDAO.getByUserId(userId, function(error, allocations) {
if (error) return next(error);
return res.render("allocations", allocations);
 });
```

Ένας ασφαλέστερος τρόπος είναι η επαναφορά του allocations για συνδεδεμένους χρήστες χρησιμοποιώντας req.session.userId αντί να χρησιμοποιούμε το url απευθείας.

A5 CWE-933 Κακή Παραμετροποίηση Ασφαλείας (Security Misconfiguration)

Αυτή η ευπάθεια επιτρέπει στον επιτιθέμενο να αποκτήσει πρόσβαση σε προεπιλεγμένους λογαριασμούς, μη χρησιμοποιημένες σελίδες, μη ενημερωμένες ευπάθειες, μη προστατευμένα αρχεία, καταλόγους και να αποκτήσει μη εξουσιοδοτημένη πρόσβαση ή γνώση του συστήματος. Η κακή παραμετροποίηση ασφαλείας μπορεί να γίνει σε οποιαδήποτε επίπεδο στην εφαρμογή, στην πλατφόρμα, στην βάση δεδομένων, στον εξυπηρετητή εφαρμογών ή και στον πηγαίο κώδικα. Οι προγραμματιστές και διαχειριστές ενός συστήματος πρέπει να συνεργαστούν για να διασφαλίσουν ότι έχει γίνει ορθή παραμετροποίηση.

Μηχανισμοί επίθεσης. Η ευπάθεια περιλαμβάνει πολλών ειδών επιθέσεις, τις οποίες ένας επιτιθέμενος μπορεί να εκμεταλλευτεί:

- Εάν ο εξυπηρετητής εφαρμογών έχει ρυθμιστεί να εκτελεί κώδικα με εξουσιοδότηση διαχειριστή, τότε ένας επιτιθέμενος μπορεί να εκτελέσει κακόβουλο λογισμικό ή διεργασίες στον εξυπηρετητή.
- Να γράψει, διαβάσει ή διαγράψει τα αρχεία συστήματος.
- Εάν το αιτούμενο μέγεθος του επιτρέπεται να φορτώσει στον εξυπηρετητή δεν έχει όριο, ο επιτιθέμενος μπορεί να φορτώσει στον εξυπηρετητή έναν πολύ μεγάλο αρχείο που θα υπερφορτώσει τους καταχωρητές και τον επεξεργαστή και ο εξυπηρετητής θα σταματήσει να προσφέρει υπηρεσίες λόγο υπερφόρτωσης.

Μέθοδος αποτροπής.

- Χρησιμοποιώντας τις τελευταίες σταθερές εκδόσεις του και ελέγχοντας για τις ευπάθειες που δημοσιεύονται.
- Αποφυγή εκτέλεσης εφαρμογών με δικαιώματα διαχειριστή.
- Ο έλεγχος για προεπιλεγμένες επικεφαλίδες HTTP προκειμένου να αποφευχθεί η αποκάλυψη για την εσωτερική υλοποίηση.
- Με την χρήση γενικών ονομάτων για cookies
- Θέτοντας όριο στο μέγεθος του αιτήματος http, στον τύπο και στα αρχεία που φορτώνονται στον εξυπηρετητή.
- Καθορίζοντας συγκεκριμένες επικεφαλίδες HTTP

Παράδειγμα πηγαίου κώδικα. Μια προκαθορισμένη επικεφαλίδα HTTP μπορεί να περιέχει λεπτομέρειες υλοποίησης στο επιτιθέμενο header. Μπορούν να αποκλειστούν περιέχοντας αυτόν τον κώδικα app.disable("x-powered-by");

A6 CWE-934 Έκθεση Ευαίσθητων Δεδομένων (Sensitive Data Exposure)

Αυτή η ευπάθεια επιτρέπει στον επιτιθέμενο να έχει πρόσβαση σε ευαίσθητα προσωπικά δεδομένα, όπως στοιχεία πιστωτικών καρτών, πιστοποιητικά εξουσιοδότησης και σε δεδομένα που επιτρέπουν την κλοπή ταυτότητας, διευκολύνοντας την άπατη με πιστωτικές κάρτες και κάθε άλλου είδους παράνομη χρήση των προσωπικών δεδομένων. Η απώλεια δεδομένων τέτοιου περιεχομένου δημιουργεί επιπτώσεις στο επιχειρείν και επιφέρει πλήγμα στην εταιρική φήμη. Τα ευαίσθητα δεδομένα χρειάζονται επιπλέον προστασία, όπως κρυπτογράφηση και ειδική μεταχείριση, όταν ανταλλάσσονται σε έναν περιηγητή.

Μηχανισμός Επίθεσης. Εάν ένας ιστότοπος δεν χρησιμοποιεί SSL/TLS για όλες τις ιστοσελίδες που απαιτούν διαπίστευση, τότε ο επιτιθέμενος μπορεί να παρακολουθεί την κίνηση του δικτύου και να κλέψει τα cookie του χρήστη, να χρησιμοποιήσει τη συνεδρία του χρήστη και να έχει πρόσβαση σε ευαίσθητα δεδομένα.

Μέθοδος αποτροπής.

- Με την χρήση ασφαλούς HTTPS δικτυακού πρωτόκολλου
- Με την κρυπτογράφηση όλων τα ευαίσθητων δεδομένων στην αποθήκευση και μεταφορά
- Με τη μη αποθήκευση ευαίσθητων προσωπικών δεδομένων εάν δεν απαιτούνται και να διαγράφονται όσο γρηγορότερα γίνεται
- Με διασφάλιση της χρήσης ισχυρών αλγορίθμων και ισχυρών κλειδιών
- Με απενεργοποίηση των πεδίων αυτόματης συμπλήρωσης φορμων τα οποία συλλέγουν προσωπικά δεδομένα και απενεργοποίηση της κρυφής μνήμης που περιέχει τέτοια.

Παράδειγμα πηγαίου κώδικα. Μια επισφαλής δοκιμαστική εφαρμογή χρησιμοποίει HTTP σύνδεση για την επικοινωνία με τον εξυπηρετητή. Χρειάζεται ένα ιδιωτικό κλειδί και ένα πιστοποιητικό. Ένα παράδειγμα πηγαίου κώδικα

```
// Load keys for establishing secure HTTPS connection
var fs = require("fs");
var https = require("https");
var path = require("path");
var httpsOptions = {
```

```
    key: fs.readFileSync(path.resolve(__dirname, "./app/cert/key.pem")),
    cert: fs.readFileSync(path.resolve(__dirname, "./app/cert/cert.pem"))
};
```

A7 CWE-935 Έλλειψη Έλεγχου Πρόσβασης Λειτουργίας (Missing Function Level Access Control)

Οι περισσότερες εφαρμογές ιστού επαληθεύουν το επίπεδο πρόσβαση πριν δώσουν την συνάρτηση στο γραφικό περιβάλλον. Όμως πρέπει να εκτελέσουν τους ίδιους ελέγχους πρόσβασης σε επίπεδο συναρτήσεων και σε επίπεδο εξυπηρετητή.

Μηχανισμός Επίθεσης. Εάν οι αιτήσεις δεν είναι επαληθευμένες για δικαιώματα πρόσβασης στον εξυπηρετητή, οι επιτιθέμενοι μπορούν να τις παραποιήσουν για να έχουν πρόσβαση χωρίς την κατάλληλη εξουσιοδότηση.

Μέθοδος αποτροπής. Οι εφαρμογές ιστού συνήθως δεν εμφανίζουν συνδέσμους και κουμπιά σε μη εξουσιοδοτημένες συναρτήσεις. Επίσης, πρέπει να υλοποιηθούν έλεγχοι στον ελεγκτή ή στην επιχειρησιακή λογική.

Παράδειγμα πηγαίου κώδικα. Σε αυτή την ευπαθή εφαρμογή δεν υπάρχει έλεγχος εξουσιοδότηση στο Benefits που έχουν σχέση με δρομολογήσεις στο routes/index.js

```
// Benefits Page
app.get("/benefits", isLoggedIn, benefitsHandler.displayBenefits);
app.post("/benefits", isLoggedIn, benefitsHandler.updateBenefits);
```

Μπορεί να επιδιορθωθεί προσθέτοντας έλεγχο του ρόλου του χρήστη.

```
// Benefits Page
app.get("/benefits", isLoggedIn, isAdmin, benefitsHandler.displayBenefits);
app.post("/benefits", isLoggedIn, isAdmin, benefitsHandler.updateBenefits);
```

A8 CWE-936 Cross Site Request Forgery

Μια επίθεση CSRF υποκλέπτει cookies των θυμάτων και άλλες πληροφορίες πιστοποίησης που χρησιμοποιούνται για την σύνδεση σε μια ευπαθή ιστοσελίδα. Μόλις ολοκληρωθεί η διαδικασία, ο εισβολέας μπορεί να ελέγξει τη συνεδρία του θύματος και να έχει τον πλήρη έλεγχο του λογαριασμού του.

Μηχανισμός Επίθεσης. Καθώς τα προγράμματα περιήγησης αποστέλλουν αυτόματα τα διαπιστευτήρια, όπως cookies συνεδρίας με τα αιτήματα HTTP στον εξυπηρετητή από όπου τα cookies ελήφθησαν, οι επιτιθέμενοι μπορούν να δημιουργήσουν κακόβουλες ιστοσελίδες που δημιουργούν πλαστά αιτήματα και οι οποίες είναι πανομοιότυπες με έγκυρες ιστοσελίδες.

Μέθοδος αποτροπής. Μπορούμε να αποτρέψουμε την επίθεση με δύο ξεχωριστούς ελέγχους άμυνας για CSRF που δεν απαιτεί την παρέμβαση του χρήστη. Απαιτείται η αγνόηση των αιτημάτων προέλευσης (π.χ., CORS) και επαληθεύεται η κεφαλίδα της αίτησης ώστε να είναι ίδιαμε την προέλευση και τα CSRF token.

Παράδειγμα πηγαίου κώδικα. Στην επιθεση CSRF τα δεδομένα που έχουν αλλάξει είναι στην παράμετρο "EmailAddress". Εάν ο χρήστης παραπλανηθεί και επισκεφτεί την ιστοσελίδα υπό τον έλεγχο του επιτιθεμένου, ο παρακάτω κώδικας μπορεί να χρησιμοποιηθεί για να αλλάξει την ηλεκτρονική διεύθυνση, η οποία αποτελεί τα διαπιστευτήρια για την είσοδο στην ιστοσελίδα [26].

```
<html><body>
<H1>Hello</H1>
<img src=''http://vulnerablesite.com/MyAccount?EmailAddress=anaddress@asite.com"
width="1" height="1" />
</body></html>
```

A9 CWE-937 Χρήση Στοιχείων Που Έχουν Ήδη Γνωστά Σφάλματα (Using Components with known Vulnerabilities)

Συστατικά όπως βιβλιοθήκες και frameworks συνήθως εκτελούνται με δικαιώματα πλήρους πρόσβασης. Εάν ένα ευπαθές συστατικό έχει εκμεταλλευτεί από έναν επιτιθέμενο τότε υπάρχει πιθανότητα να υποκλαπούν δεδομένα από τον εξυπηρετητή. Εφαρμογές που χρησιμοποιούν συστατικά με γνωστές ευπάθειες μπορεί να υπονομευόσουν την άμυνα της εφαρμογής και να ενεργοποιήσουν μια σειρά από πιθανές επιθέσεις με επιπτώσεις. Κάποια έργα βοηθούν στην ενημέρωση των ευπαθειών:

- Το Node Security project είναι πηγή γνωστών ευπαθειών
- Το Snyk.io είναι ακόμα ένα Node.js εργαλείο και πλατφόρμα για σάρωση και ανίχνευση ευπαθών πακέτων.
- Ο έλεγχος npm-check για απαρχαιωμένες, εσφαλμένες και αχρησιμοποίητες εξαρτήσεις
- bithound.io" είναι μια Node.js υπηρεσία ανάλυσης κώδικα

Μηχανισμοί επίθεσης. Τα πακέτα npm είναι στοιχειώδες συστατικό μιας node εφαρμογής. Αυτά τα πακέτα μπορούν εάν δεν έχουν περάσει ελέγχους ασφάλειας ή με πρόθεση να περιέχουν κακόβουλο κώδικα. Μέσω των ανασφαλών πακέτων ένας επιτιθέμενος μπορεί να:

- δημιουργήσει και να εκτελέσει κώδικα σε διάφορες φάσεις κάνοντας την εγκατάσταση ή την χρήση του πακέτου.
- Γράψει, διαβάσει και να διαγράψει αρχεία στο σύστημα
- Γράψει και να εκτελέσει δυαδικά αρχεία
- Συλλέξει ευαίσθητα δεδομένα και να τα στείλει σε απομακρυσμένο σημείο

Μέθοδος αποτροπής. Υπάρχουν μέτρα για την αποτροπή κακόβουλων πακέτων npm:

- Να μην εκτελούνται εφαρμογές με δικαιώματα root
- Προτιμούμε πακέτα που να περιέχουν στατική ανάλυση κώδικα.
- Προτιμούμε πακέτα που να περιέχουν περιεκτικές δοκιμές και αξιολογήσεις για τις συναρτήσεις που χρησιμοποιούν οι εφαρμογές
- Ανάλυση του κώδικα για αρχεία που έχουν πρόσβαση στην βάση δεδομένων.

- Έρευνα για το πόσο δημοφιλές είναι το πακέτο, από ποιον γράφτηκε και τι αλλά πακέτα χρησιμοποιεί.

A10 CWE-938 Μη Επικυρωμένες Δρομολογήσεις και Προωθήσεις (Unvalidated Redirects and Forwards)

Οι εφαρμογές ιστού συχνά δρομολογούν και προωθούν χρήστες σε άλλες σελίδες και ιστοσελίδες, και χρησιμοποιούν μη έμπιστα δεδομένα για τον καθορισμό της σελίδας προορισμού. Χωρίς την κατάλληλη επικύρωση οι επιτιθέμενοι μπορούν να δρομολογήσουν τα θύματα σε σελίδες phishing ή κακόβουλες ιστοσελίδες

Μηχανισμός επίθεσης. Ένας επιτιθέμενος μπορεί να χρησιμοποιήσει μη επικυρωμένους συνδέσμους ως μέσο για να δρομολογήσει έναν χρήστη σε κακόβουλο περιεχόμενο και να το επιλέξει.

Μέθοδος αποτροπής. Η ασφαλής χρήση δρομολογήσεων και προωθήσεων μπορεί να γίνει με διάφορους τρόπους:

- Αποφεύγοντας τις δρομολογήσεις και τις προωθήσεις
- Εάν χρησιμοποιούνται να μην περιέχουν παραμέτρους στον υπολογισμό του προορισμού.
- Εάν οι παράμετροι προορισμού δεν μπορούν να αποφευχθούν πρέπει να σιγουρέψουμε ότι οι τιμές που δίνονται είναι επικυρωμένες και να αποφεύγονται πραγματικά URL η τμήματα URL.

Παράδειγμα πηγαίου κώδικα. Η εφαρμογή δρομολογεί σε ένα άλλη ιστοσελίδα χωρίς επικύρωση του συνδέσμου URL.

```
// Handle redirect for learning resources link

  app.get("/learn", function (req, res, next) {

    return res.redirect(req.query.url);

  });
```

Ο επιτιθέμενος μπορεί να αλλάξει την παράμετρο του ερωτήματος του URL και να δείχνει μια κακόβουλη ιστοσελίδα. Το θύμα είναι πιθανόν να το επιλέξει γιατί τμήμα του συνδέσμου δείχνει μια έμπιστη ιστοσελίδα.

Έχει εκδοθεί προσωρινή έκδοση στης 10 Απριλίου με τις 10 σημαντικότερες ευπάθειες του 2017. Στον παρακάτω πινάκα 2.9 υπάρχει σύγκριση της έκδοσης του 2013 και της προσωρινή του 2017.

OWASP Top 10 – 2013 (Previous)	OWASP Top 10 – 2017 (New)
A1 – Injection	A1 – Injection
A2 – Broken Authentication and Session Management	A2 – Broken Authentication and Session Management
A3 – Cross-Site Scripting (XSS)	A3 – Cross-Site Scripting (XSS)
A4 – Insecure Direct Object References - Merged with A7	A4 – Broken Access Control (Original category in 2003/2004)
A5 – Security Misconfiguration	A5 – Security Misconfiguration
A6 – Sensitive Data Exposure	A6 – Sensitive Data Exposure
A7 – Missing Function Level Access Control - Merged with A4	A7 – Insufficient Attack Protection (NEW)
A8 – Cross-Site Request Forgery (CSRF)	A8 – Cross-Site Request Forgery (CSRF)
A9 – Using Components with Known Vulnerabilities	A9 – Using Components with Known Vulnerabilities
A10 – Unvalidated Redirects and Forwards - Dropped	A10 – Underprotected APIs (NEW)

Πίνακας 2.9: Συγκριτικός Πινάκας Εκδόσεων OWASP TOP 10 2013 και 2017

Στον πινάκα 2.9 βλέπουμε ότι οι αλλαγές είναι πολύ λίγες. Οι ευπάθειες A1, A2, A3, A5, A6, A8 και A9 δεν έχουν αλλάξει θέση. Η ευπάθεια A4 και η A7 έχει συγχωνευτεί και τοποθετηθεί ως A4 και έχουμε δυο νέες εισόδους την A7 (Ανεπαρκής προστασία από επιθέσεις) και A10 (μη προστατευμένα APIs) [25].

Κεφάλαιο 3
Σαρωτές Εφαρμογών Ιστού

Στο παρόν κεφάλαιο αναλύουμε τους αυτοματοποιημένους σαρωτές ιστού με τα πλεονεκτήματα και τα μειονεκτήματα τους και θα περιγράψουμε τα χαρακτηριστικά τεσσάρων εμπορικών πακέτων και έξι σαρωτών ανοιχτού κώδικα που θα χρησιμοποιήσουμε στο εργαστήριο μας.

3.1 Ποια η Λειτουργία των Αυτοματοποιημένων Σαρωτών

Οι σαρωτές μαύρου κουτιού είναι μια κατηγορία εργαλείων ασφαλείας που μπορούν να χρησιμοποιηθούν για τον προσδιορισμό θεμάτων ασφαλείας στις εφαρμογές ιστού. Τα εργαλεία αυτά αξιολογούν αυτόματα την ασφάλεια των εφαρμογών ιστού με ελάχιστη ή καμία ανθρώπινη παρέμβαση. Λειτουργούν όπως και οι χρήστες και βρίσκουν ευπάθειες ανεξάρτητα από την γλώσσα προγραμματισμού που έχει δημιουργηθεί η εφαρμογή ιστού. Είναι ικανά να έχουν πρόσβαση και να δοκιμάζουν διάφορα συστατικά μιας εφαρμογής που μπορεί να κρύβονται σε javascript ή σε εφαρμογές flash. Τα εργαλεία αυτά δεν δοκιμάζουν τον πηγαίο κώδικα αλλά δημιουργούν ειδικές εισόδους που τις στέλνουν στην εφαρμογή και αναλύουν την συμπεριφορά της απάντησης για ευπάθειες ασφαλείας. Πάσχουν όμως από περιορισμούς που έχουν να κάνουν με την πολυπλοκότατα της εφαρμογής [05].

3.1.1 Πλεονεκτήματα Σαρωτών Ιστού Ανοιχτού Κώδικα

Τα πλεονεκτήματα των σαρωτών είναι [05]:

- Είναι δωρεάν εκτός εάν υπάρχουν πρόσθετα που έχουν αναπτυχτεί από άλλους προγραμματιστές και μπορεί να υπάρχει χρέωση.
- Επειδή είναι δωρεάν μπορεί ένας προγραμματιστής ή μια μικρού μεγέθους εταιρία που δεν μπορεί να αγοράσει εμπορικό λογισμικό να ελέγξει την υπό ανάπτυξη ιστοσελίδα τους για ευπάθειες ασφαλείας.
- Είναι ανεξάρτητη από την τεχνολογία που έχει υλοποιηθεί ο υπό ανάλυση ιστότοπος δηλαδή java, php, .net, κ.ο.κ.
- Μπορούν να εξομοιώσουν έναν επιτιθέμενο και να ελέγξουν τα αποτελέσματα της επίθεσης.
- Λόγο του ανοιχτού κώδικα ένας προγραμματιστής ασφαλείας μπορεί να εκπαιδευτεί στην ανάπτυξη εφαρμογών σαρωτών ιστού.

3.1.2 Μειονεκτήματα Σαρωτών Ιστού Ανοιχτού Κώδικα

Οι αδυναμίες των σαρωτών και οι περιορισμοί τους είναι [05]:

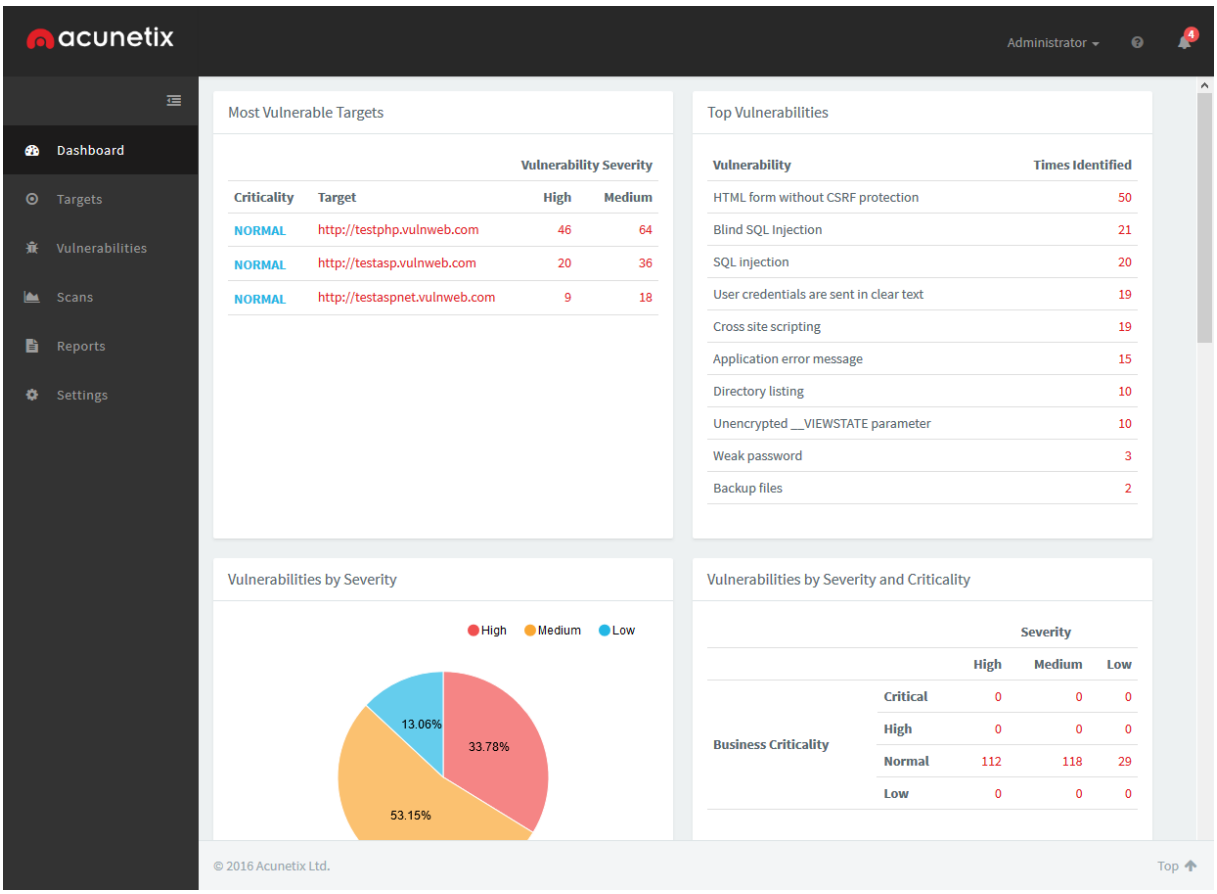
- Δεν περιέχουν όλα τα χαρακτηριστικά μιας πιστοποιημένης εμπορικής εφαρμογής
- Δεν έχει συχνές ή καθόλου ενημερώσεις σε προγραμματιστικά σφάλματα (bugs) ή γίνονται με καθυστέρηση σε σχέση με τα εμπορικά πακέτα.
- Δεν έχει συχνές ενημερώσεις με νέες ευπάθειες ασφαλείας
- Δεν καλύπτουν όλες τις υλοποιήσεις σε εφαρμογές ιστού με δυναμικό περιεχόμενο όπως για παράδειγμα εάν η ιστοσελίδα να περιέχει flash.
- Είναι δύσκολο να ανιχνεύσουν αδυναμίες σε κρυπτογραφικές συναρτήσεις ή λογικά σφάλματα
- Είναι δύσχρηστα γιατί δεν διαθέτουν όλα γραφικό περιβάλλον, υποστήριξη από την εταιρία υλοποίησης ή εγχειρίδια χρήσης.

3.2 Εργαλεία Δοκιμών Διείσδυσης σε Εφαρμογές Ιστού

Σε αυτή την ενότητα θα ερευνηθούν τέσσερα εμπορικά εργαλεία για δοκιμές διείσδυσης σε εφαρμογές ιστού, τα οποία είναι αποτελεσματικά στην ανίχνευση ευπαθειών και έχουν μικρά ποσοστά στα ψευδώς θετικά αποτελέσματα.

3.2.1 Εμπορικά Εργαλεία

Το Web Vulnerability Scanner από την Acunetix είναι στην έκδοση 11. Το Acunetix είναι ένας σαρωτής ευπάθειας ιστού (WVS) που σαρώνει και εντοπίζει ευπάθειες ιστού. Ανιχνεύει έναν ιστότοπο και εντοπίζει XSS, SQLi και άλλες ευπάθειες. Είναι ταχύτατο εργαλείο και εύκολο στη χρήση. Σαρώνει ιστοσελίδες WordPress και μπορεί να εντοπίσει περισσότερες από 1200 ευπάθειες στο WordPress [11].



Σχήμα 3.2 : Acunetix Web Vulnerability Scanner

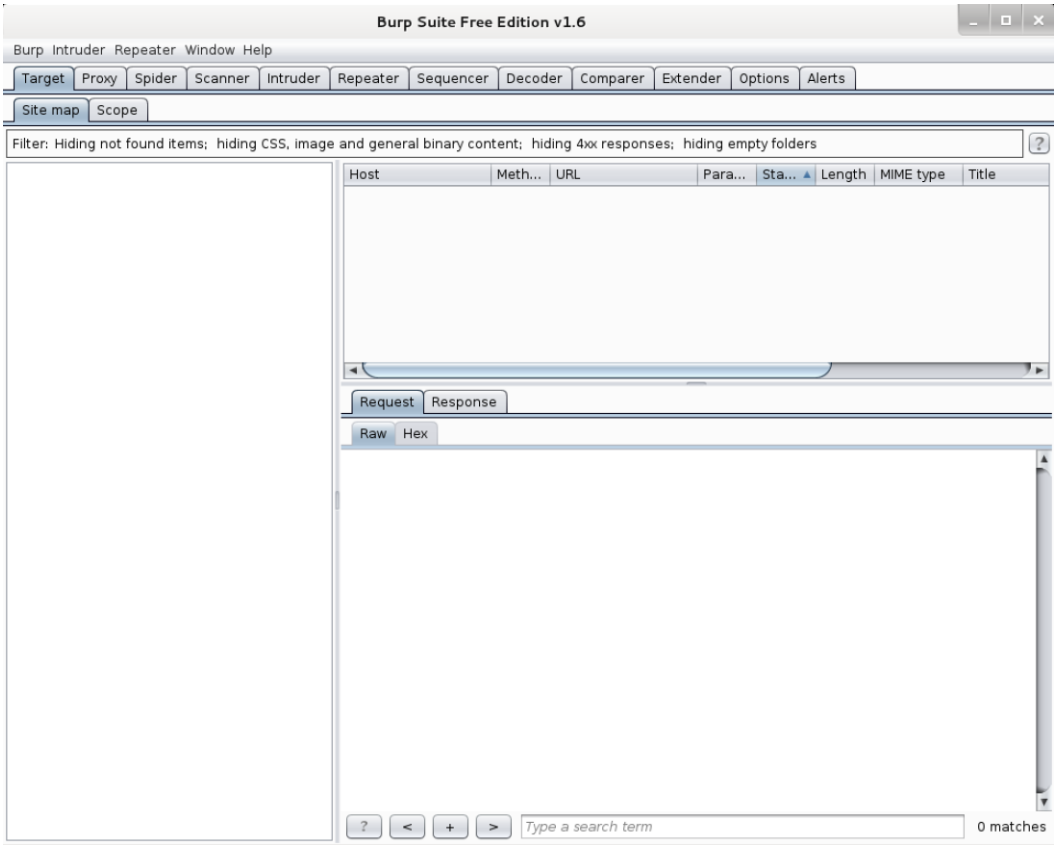
Το Burp Suite είναι ένα εργαλείο που βασίζεται στην Java για τη διεξαγωγή δοκιμών ασφαλείας εφαρμογών ιστού. Κύριο χαρακτηριστικό του είναι ότι διάφορα εργαλεία που απαιτούνται για τη δοκιμή ενσωματώνονται σε μια ενιαία πλατφόρμα. Διατίθεται σε δωρεάν και εμπορική έκδοση[14].

Η δωρεάν έκδοση του Burp Suite έχει τα ακόλουθα χαρακτηριστικά:

- Υπάρχει διακομιστής μεσολάβησης για να αναλύει το αίτημα και την απόκριση.
- Διαθέτει το Burp Spider για να ανιχνεύσει τις σελίδες και τη σύνδεση της εφαρμογής στόχου.
- Διαθέτει το Burp Repeater για τον χειρισμό και την επανάληψη του αιτήματος πολλαπλού χρόνου.
- Διαθέτει το Burp Sequencer για την ανάλυση της τυχαιότητας και της αντοχής του συμβόλου σύνδεσης.
- Διαθέτει το Burp Intruder να εκτελεί εξατομικευμένη αυτοματοποιημένη επίθεση για την εύρεση και αξιοποίηση ευπαθειών.

Υπάρχουν ορισμένες λειτουργίες που υπάρχουν μόνο στην επαγγελματική έκδοση και περιλαμβάνουν:

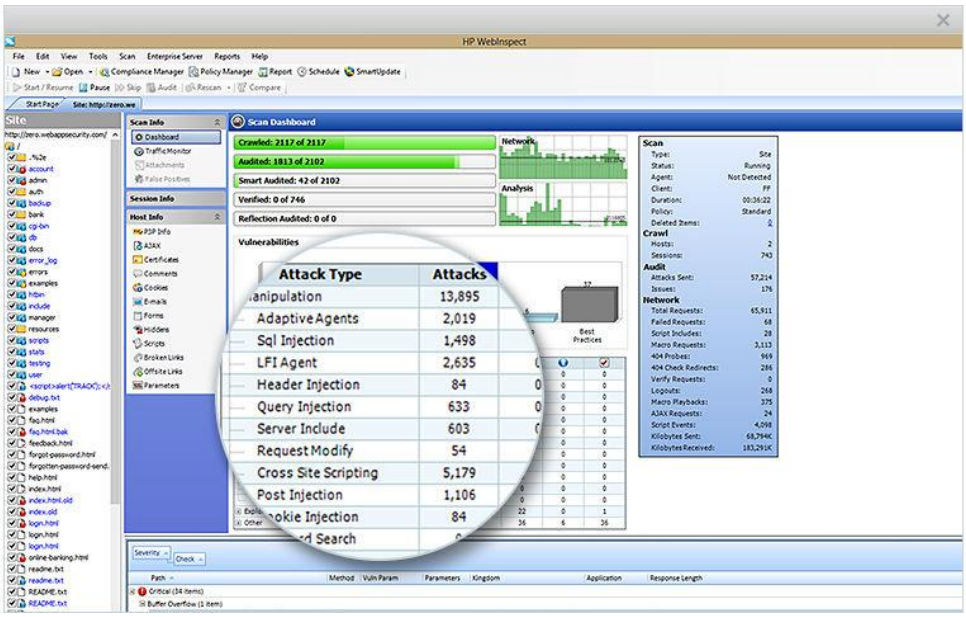
- Έναν προηγμένο σαρωτή εφαρμογών ιστού για την αυτόματη ανίχνευση των τρωτών σημείων στις εφαρμογές ιστού
- Επέκταση του Burp suite που επιτρέπει την δημιουργία νέων plug-ins
- Δυνατότητα αποθήκευσης της τρέχουσας κατάστασης και χρήσης αργότερα
- Δυνατότητα δημιουργίας αναφοράς σάρωσης



Σχήμα 3.3 : Burp Suite

Το HPE WebInspect [49] είναι μια εφαρμογή αξιολόγησης ασφάλειας που έχει σχεδιαστεί για να αναλύσει τις σύγχρονες σύνθετες εφαρμογές ιστού και τις υπηρεσιες ιστού για ευπάθειες ασφαλείας. Με τεχνολογία που παρακολουθεί τον χρόνο εκτέλεσης των εφαρμογών μέσω του HPE WebInspect Agent, η HPE WebInspect παρέχει ευρύτερη δοκιμή ασφάλειας δυναμικών εφαρμογών (DAST) για να ανιχνεύει νέους τύπους ευπάθειας που συχνά δεν ανιχνεύονται από τις τεχνολογίες ελέγχου της ασφάλειας μαύρου κουτιού. Συγκεκριμένα περιλαμβάνει.

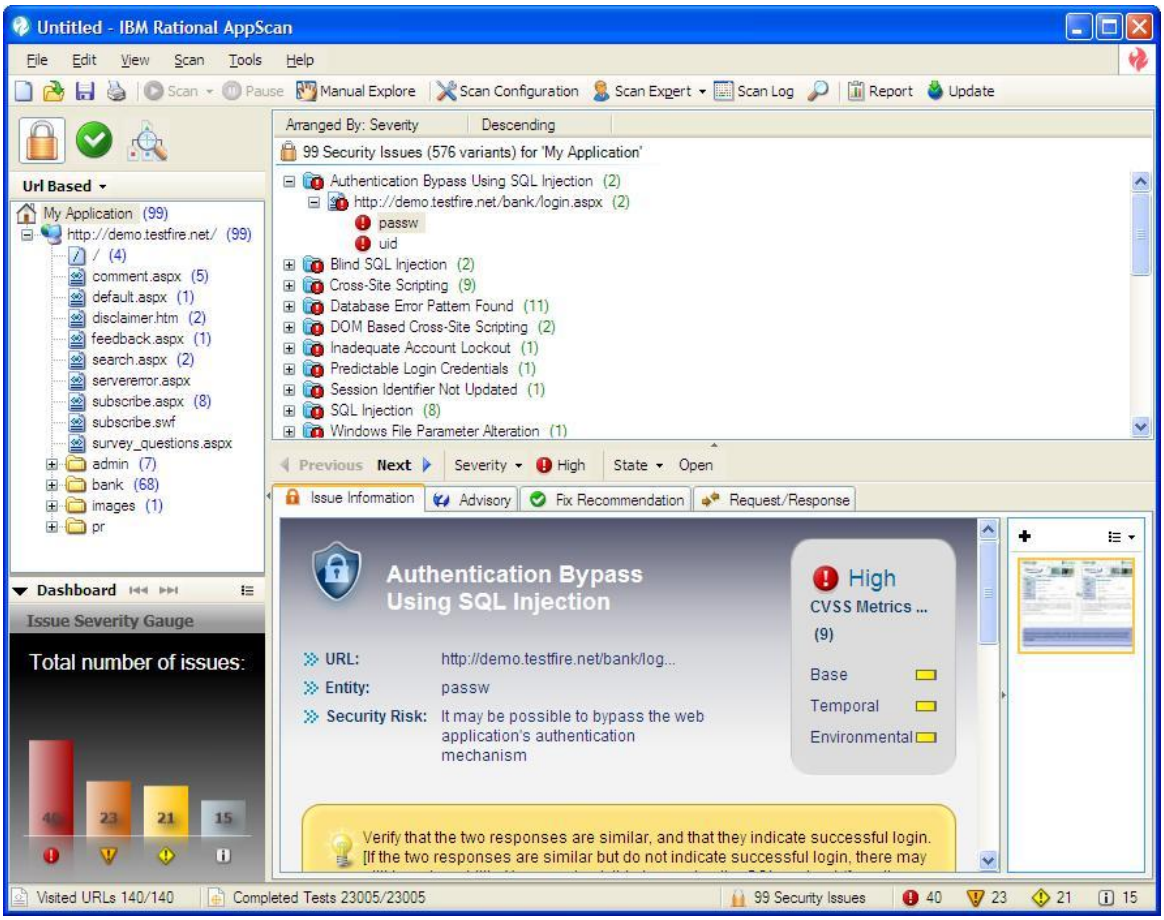
- Ενσωματωμένο δυναμικό κώδικα και ανάλυση χρόνου εκτέλεσης για εντοπισμό αδυναμιών και γρήγορης διόρθωσης
- Παρατήρηση της αντίδραση της εφαρμογής σε επιθέσεις σε επίπεδο κώδικα κατά τη διάρκεια δυναμικών σαρώσεων
- Προσδιορισμό και ανίχνευση περισσότερων εφαρμογών για την επέκταση της κάλυψης της επιφάνειας επίθεσης
- Δίνεται η δυνατότητα ανάπτυξης με ακριβείς λεπτομέρειες, όπως είναι η γραμμή λεπτομέρειας κώδικα, track stack και ερωτήματα SQL
- Εμπεριέχει τεχνολογίες όπως τη ταυτόχρονη ανίχνευση, τον έλεγχο και τη ταυτόχρονη σάρωση
- Υποστήριξη για τις πιο πρόσφατες τεχνολογίες ιστού, συμπεριλαμβανομένων των HTML5, JSON, AJAX, JavaScript



Σχήμα 3.4 : HPE WebInspect

Το IBM AppScan [13] βοηθά τους οργανισμούς να μειώσουν την πιθανότητα επιθέσεων με εφαρμογές ιστού και δαπανηρές παραβιάσεις δεδομένων, αυτοματοποιώντας τις δοκιμές ευπάθειας ασφαλείας εφαρμογών. Το IBM S AppScan μπορεί να χρησιμοποιηθεί για τη μείωση του κινδύνου επιτρέποντάς σας να δοκιμάσετε εφαρμογές πριν από την ανάπτυξη και για συνεχή αξιολόγηση κινδύνου σε περιβάλλοντα παραγωγής. Το IBM AppScan υποστηρίζει:

- Ευρεία κάλυψη για σάρωση και δοκιμή, ακριβής σάρωση με προηγμένες δοκιμές που προσφέρουν υψηλά επίπεδα ακρίβειας, ενισχυμένη γνώση και συμμόρφωση
- Αυτοματοποιημένη Δοκιμή Ασφάλειας Δυναμικής Εφαρμογής (DAST) και Δοκιμή Ασφάλειας Διαδραστικών Εφαρμογών (IAST) σύγχρονων εφαρμογών και υπηρεσιών web.
- Πλήρης υποστήριξη JavaScript στα πλαίσια Web 2.0,
- Δοκιμές υπηρεσιών SOAP και REST web, που καλύπτουν την υποδομή XML και JSON. Υποστήριξη προτύπων ασφαλείας WS, κρυπτογράφησης XML και υπογραφών XML.
- Πάνω από 40 εκθέσεις κανονιστικής συμμόρφωσης



Σχήμα 3.5: IBM AppScan

3.2.2 Εργαλεία Ανοιχτού Κώδικα

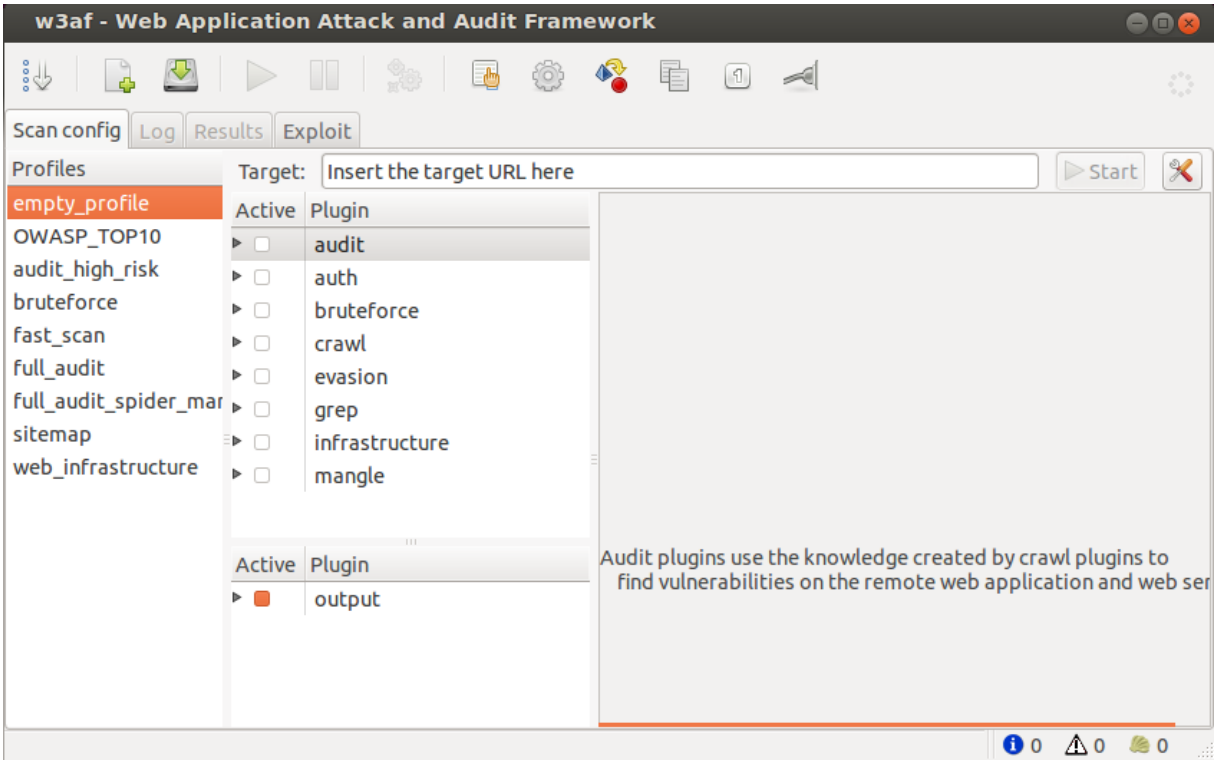
Το Skipfish [05, 20] είναι ένα σαρωτής ευπαθειών εφαρμογών ιστού ανοιχτού κώδικα γραμμένος σε γλώσσα προγραμματισμού C. Αναπτύχτηκε από την εταιρία Google και στόχος του είναι η ανίχνευση πιθανών προβλημάτων ασφαλείας, τα οποία εμπεριέχονται σε εφαρμογές ιστού. Εκτελεί σαρώσεις ευπαθειών ασφαλείας υψηλού (XSS, SQLi, XMLi), μεσαίου (XSS αποθηκευόμενες και αντανακλώμενες), χαμηλού ρίσκου (ανακατεύθυνση σε url του επιτιθέμενου), εσωτερικές προειδοποιήσεις (εάν έχει ανιχνευτεί από IPS φίλτρο) και ενημερωτικές εγγραφές (πληροφορίες πιστοποιητικού SSL). Η εφαρμογή Skipfish στο τέλος της σάρωσης δημιουργεί έναν διαδραστικό χάρτη ιστού για την εφαρμογή που έγινε η ανίχνευση με σχόλια. Οι ενεργές αιτήσεις HTTP μπορούν να ξεπεράσουν τις 2000 ανά δευτερόλεπτο εφόσον ο ιστότοπος που δοκιμάζεται μπορεί να αντέξει το φόρτο των δεδομένων.

Σχήμα 3.6: Skipfish

Χαρακτηριστικά:

- Διαθέσιμο για τα σημαντικότερα λειτουργικά συστήματα (Windows, Linux, και MacOS)
- Γραφικό Περιβάλλον
- Υψηλή ταχύτητα
- Ευκολία στη χρήση
- Χαμηλά ψευδώς θετικά αποτελέσματα ανίχνευσης
- Αναφορές

Το εργαλείο W3AF (Web Application Audit and Attack Framework) [05, 19], είναι μια εφαρμογή για έλεγχο και επίθεση σε εφαρμογές ιστού. Αναπτύχτηκε από τον Andres Riancho και είναι γραμμένο σε γλώσσα προγραμματισμού python. Η χρήση του είναι ευρέως διαδεδομένη για την ανεύρεση και εκμετάλλευση των αδυναμιών των εφαρμογών ιστού. Αποτελείται από δυο μέρη: τον πυρήνα και τα πρόσθετα. Ο πυρήνας συντονίζει τις διεργασίες και παρέχει τα χαρακτηριστικά που χρησιμοποιούνται από τα πρόσθετα. Η διαδικασία γίνεται σε τρία βήματα. Πρώτα γίνεται η αναγνώριση όλων των συνδέσεων, των εισόδων και των παραμέτρων του ερωτήματος. Έπειτα στέλνεται μια ειδική συμβολοσειρά σε κάθε είσοδο και αναλύεται η έξοδος. Τέλος, δημιουργείται μια έκθεση με τα ευρήματα.

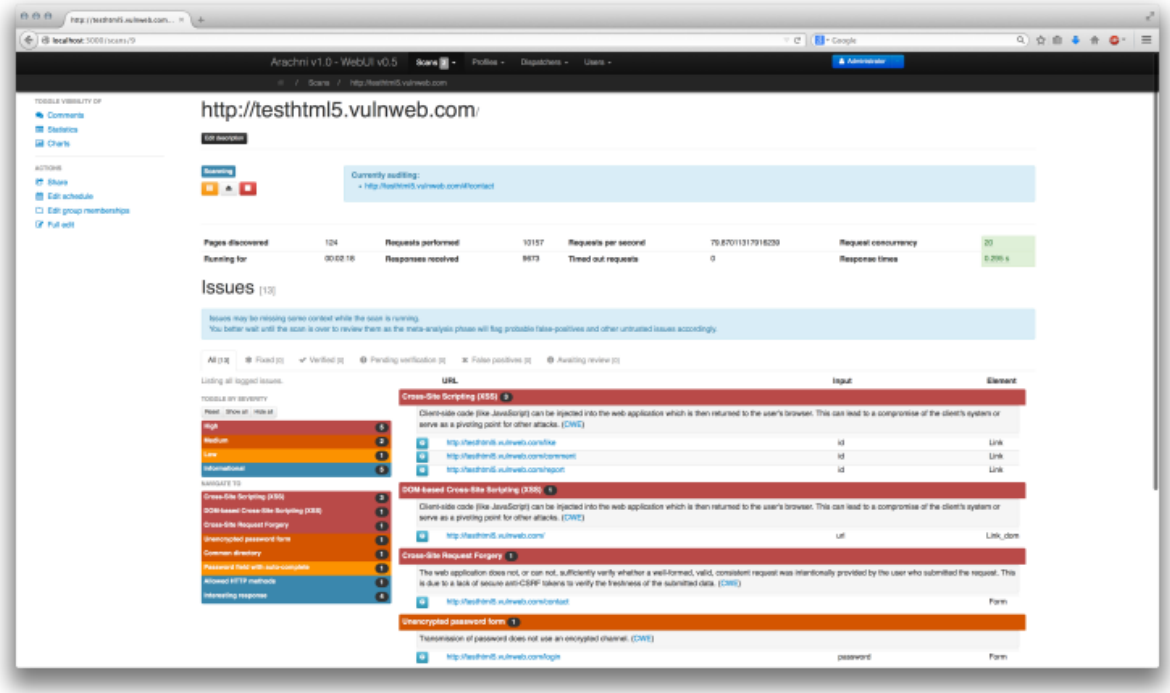


Σχήμα 3.7: W3AF

Χαρακτηριστικά

- Είναι διαθέσιμο για τα σημαντικότερα λειτουργικά συστήματα (Windows, Linux, και MacOS)
- Έχει πάνω από 130 πρόσθετα
- Είναι εύκολο στην χρήση και επεκτάσιμο
- Έχει γραφικό περιβάλλον και γραμμών κώδικα
- Έχει αναφορές
- Υπάρχει δυνατότητα εκμετάλλευσης ευπαθειών

Η εφαρμογή Arachni [05, 17] είναι υψηλής απόδοσης με δυνατότητες παραμετροποίησης και εμπεριέχει πολλά χαρακτηριστικά ανίχνευσης ευπαθειών εφαρμογών ιστού. Αναπτύχτηκε από τον Τάσο Λάσκο και είναι γραμμένη σε γλώσσα προγραμματισμού Ruby. Διατίθεται δωρεάν, όπως και ο πηγαίος κώδικας του. Θεωρείται πολυ-πλατφόρμα, υποστηρίζοντας όλα τα μεγάλα λειτουργικά συστήματα (MS Windows, Mac OS X και Linux) και διανέμεται μέσω φορητών πακέτων που επιτρέπουν την άμεση ανάπτυξη. Είναι αρκετά ευέλικτη για να καλύψει πολλές περιπτώσεις χρήσης, που κυμαίνονται από ένα απλό βοηθητικό πρόγραμμα σάρωσης γραμμής εντολών, έως μια βιβλιοθήκη Ruby, η οποία επιτρέπει τη διενέργεια ελέγχων σε μια πολλαπλών χρηστών πολλαπλών σαρώσεων πλατφόρμα. Τέλος, λόγω του ολοκληρωμένου περιβάλλοντος περιήγησης, μπορεί να υποστηρίξει εξαιρετικά πολύπλοκες εφαρμογές ιστού, οι οποίες χρησιμοποιούν τεχνολογίες όπως JavaScript, HTML5, χειρισμό DOM και AJAX. Έχει όλα τα πλήρη χαρακτηριστικά υποστήριξης και ανάλυσης ευπάθειας εφαρμογών ιστού. Αναγνωρίζονται ευπάθειες, όπως XSS (με παραλλαγές DOM), SQL injection, NoSQL injection, Code injection, H Arachni μπορεί να χειριστεί σύνθετες σύγχρονες εφαρμογές ιστού χάρη στον πραγματικό μηχανισμό του προγράμματος περιήγησης, παρέχοντας ανίχνευση των τρωτών σημείων που βασίζονται στο DOM.



Σχήμα 3.8: Arachni

Χαρακτηριστικά

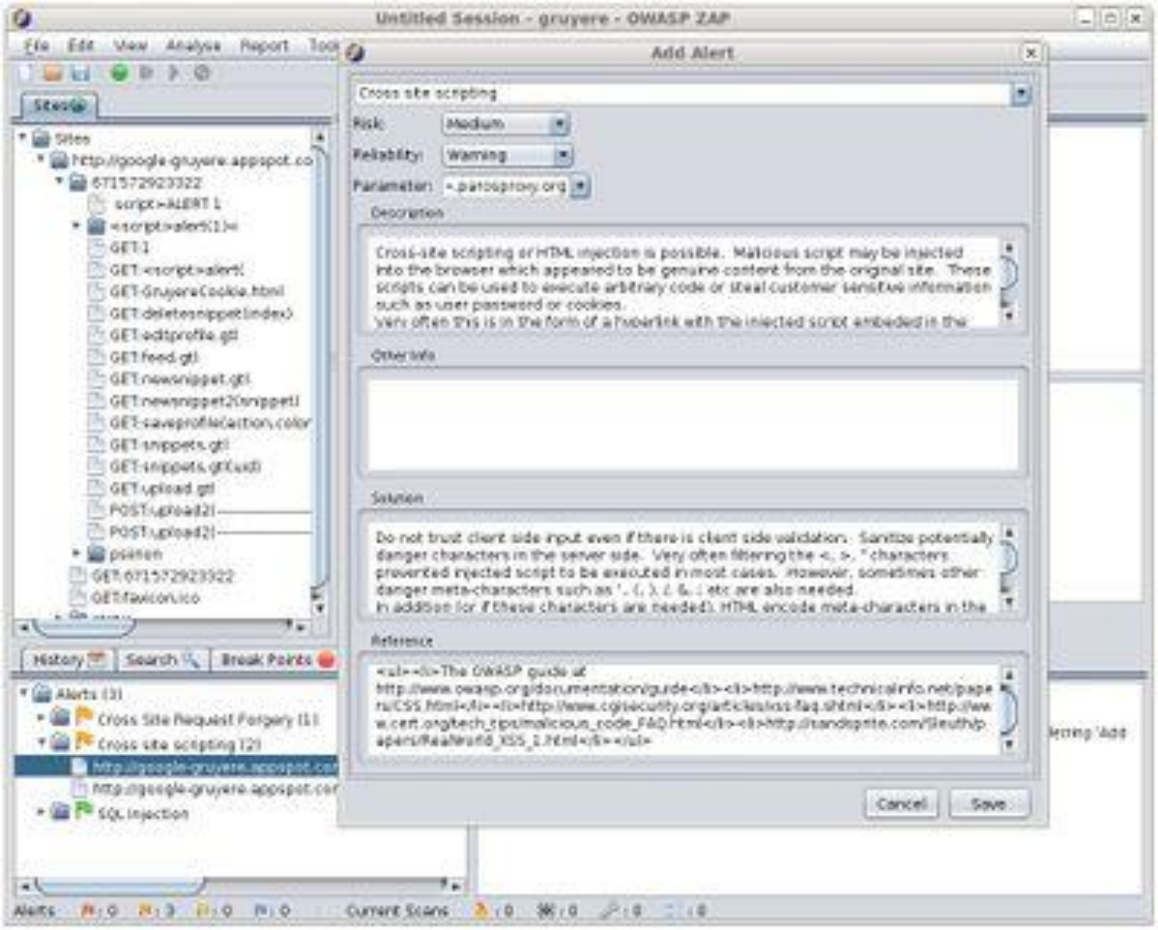
- Υψηλή απόδοση, πολύ λεπτομερής, καλά δομημένες εκθέσεις
- Υποστήριξη Cookie-jar / cookie-string
- Υποστήριξη SSL
- Υποστήριξη μεσολάβησης για SOCKS4, SOCKS4A, SOCKS5, HTTP / 1.1 και HTTP / 1.0
- Επαλήθευση μεσολάβησης
- Έλεγχος ταυτότητας ιστότοπου (Αυτοματοποιημένη βάση φόρμας, Cookie-Jar, Basic-Digest, NTLMv1 κ.α.)
- Αυτόματη ανίχνευση αποσύνδεσης και επανασύνδεσης κατά τη διάρκεια της σάρωσης (όταν η αρχική σύνδεση πραγματοποιήθηκε μέσω των plugin plug-in ή του διακομιστή μεσολάβησης)
- Διεπαφή γραμμή εντολών
- Διαδικτυακή διασύνδεση χρήστη

Το Nikto [15] είναι ένας ανοιχτού κώδικα (GPL) σαρωτής. Αναπτύχτηκε από τους Chris Sullo και David Lodge και είναι γραμμένος σε γλώσσα Perl. Εκτελεί ελέγχους σε διακομιστές ιστού για πολλαπλά στοιχεία, συμπεριλαμβανομένων πάνω από 6700 δυνητικώς επικίνδυνων αρχείων εφαρμογών, ελέγχει για παρωχημένες εκδόσεις σε πάνω από 1250 διακομιστές. Ελέγχει επίσης στοιχεία από τις παραμέτρους συστήματος των διακομιστών, όπως είναι η παρουσία των πολλαπλών αρχείων δείκτη, επιλογές διακομιστή HTTP, και εντοπίζει εγκατεστημένους εξυπηρετητές ιστού και λογισμικού. Τα στοιχεία και τα πρόσθετα σάρωσης που διαθέτει ενημερώνονται συχνά και μπορούν να ενημερώνονται αυτόματα. Το Nikto δοκιμάζει μια εφαρμογή ιστού στο συντομότερο δυνατό χρόνο, με αποτέλεσμα να γίνεται αντιληπτό σε αρχεία καταγραφής ή σε ένα IPS / IDS. Ωστόσο, υπάρχει υποστήριξη για τις μεθόδους αντι-IDS .

Χαρακτηριστικά

- Είναι διαθέσιμο για τα σημαντικότερα λειτουργικά συστήματα (Windows, Linux, και MacOS)
- Υποστηρίζει SSL
- Παρέχεται πλήρης υποστήριξη διακομιστή μεσολάβησης HTTP
- Ελέγχει για παρωχημένες εκδόσεις στα στοιχεία των εξυπηρετητών
- Υπάρχει πληθώρα αναφορών με επιλογή για παραμετροποίηση
- Σαρώνει πολλαπλές πόρτες σε εξυπηρετητή ή εξυπηρετητές μέσω αρχείων εισόδου
- Υποστηρίζει αντι-IDS
- Ενημερώνεται μέσω γραμμών εντολών
- Αναγνωρίζει το εγκατεστημένο λογισμικό μέσω αρχείων, επικεφαλίδων και εικόνων.

Το OWASP Zed Attack Proxy (ZAP) [05, 18] είναι ένα από τα δημοφιλέστερα δωρεάν εργαλεία ασφάλειας στον κόσμο και συντηρείται ενεργά από εκατοντάδες εθελοντές. Μπορεί να εντοπίσει με αυτόματο τρόπο ευπάθειες ασφαλείας σε εφαρμογές ιστού, αλλά να χρησιμοποιηθεί και με χειροκίνητο τροπο.



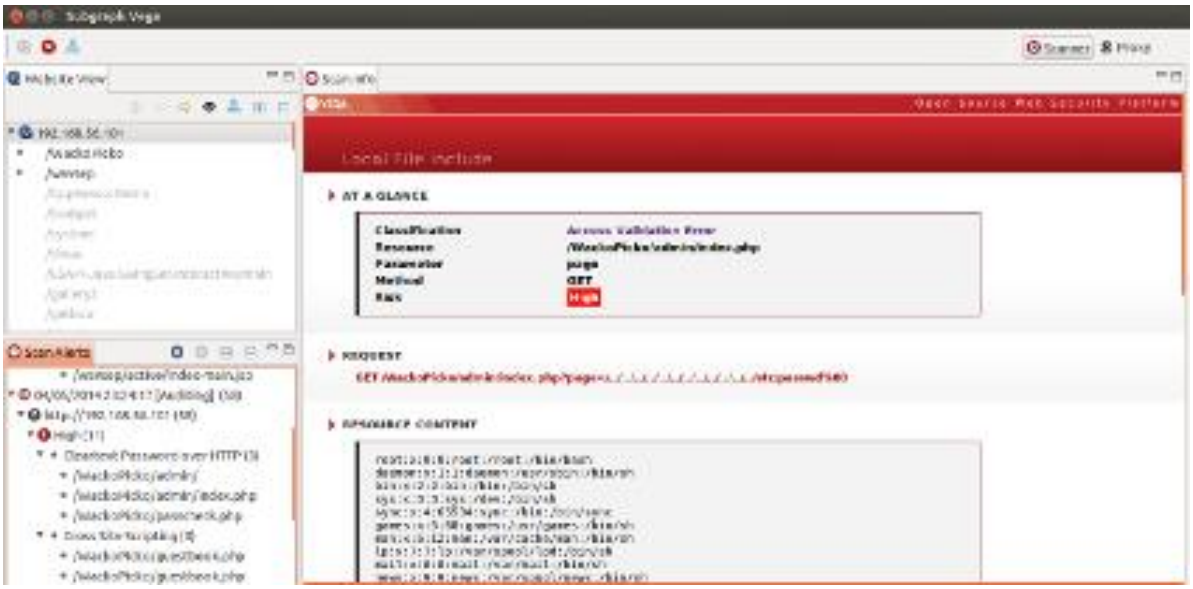
Σχήμα 3.9: OWASP ZAP

Χαρακτηριστικά

- Είναι ανοιχτού κώδικα
- Είναι διαθέσιμο για τα σημαντικότερα λειτουργικά συστήματα (Windows, Linux, και MacOS) και για Raspberry Pi
- Είναι εύκολο στην εγκατάσταση (Απλά απαιτεί java)
- Παρέχεται εντελώς δωρεάν
- Είναι εύκολο στην χρήση
- Διαθέτει αυτόματο και παθητικό σαρωτή,

- Διαθέτει δυναμικά πιστοποιητικά SSL
- Παρέχει υποστήριξη για ένα ευρύ φάσμα γλωσσών προγραμματισμού
- Γίνεται έλεγχος ταυτότητας και υποστήριξη περιόδου λειτουργίας
- Διαθέτει πρόσθετα και API με βάση το REST
- Έχει μεταφραστεί σε περισσότερες από 20 γλώσσες
- Υποστηρίζεται από ενεργή κοινότητα και υπό ενεργό ανάπτυξη

Το Vega [16] είναι ένας ανοιχτού κώδικα σαρωτής ασφάλειας. Μπορεί να βρει και να εντοπίσει ευπάθειες έγχυση SQL, XSS, ευαίσθητες πληροφορίες και άλλες ευπάθειες. Είναι γραμμένο σε Java, βασισμένο σε GUI και τρέχει σε Linux, OS X και Windows. Περιλαμβάνει ένα αυτοματοποιημένο σαρωτή για γρήγορες δοκιμές και ένα proxy για τακτική επιθεώρηση. Ο ανιχνευτής Vega μπορεί να επεκταθεί χρησιμοποιώντας API. Το Vega αναπτύχθηκε από την Subgraph.



Σχήμα 3.10: VEGA

Χαρακτηριστικά

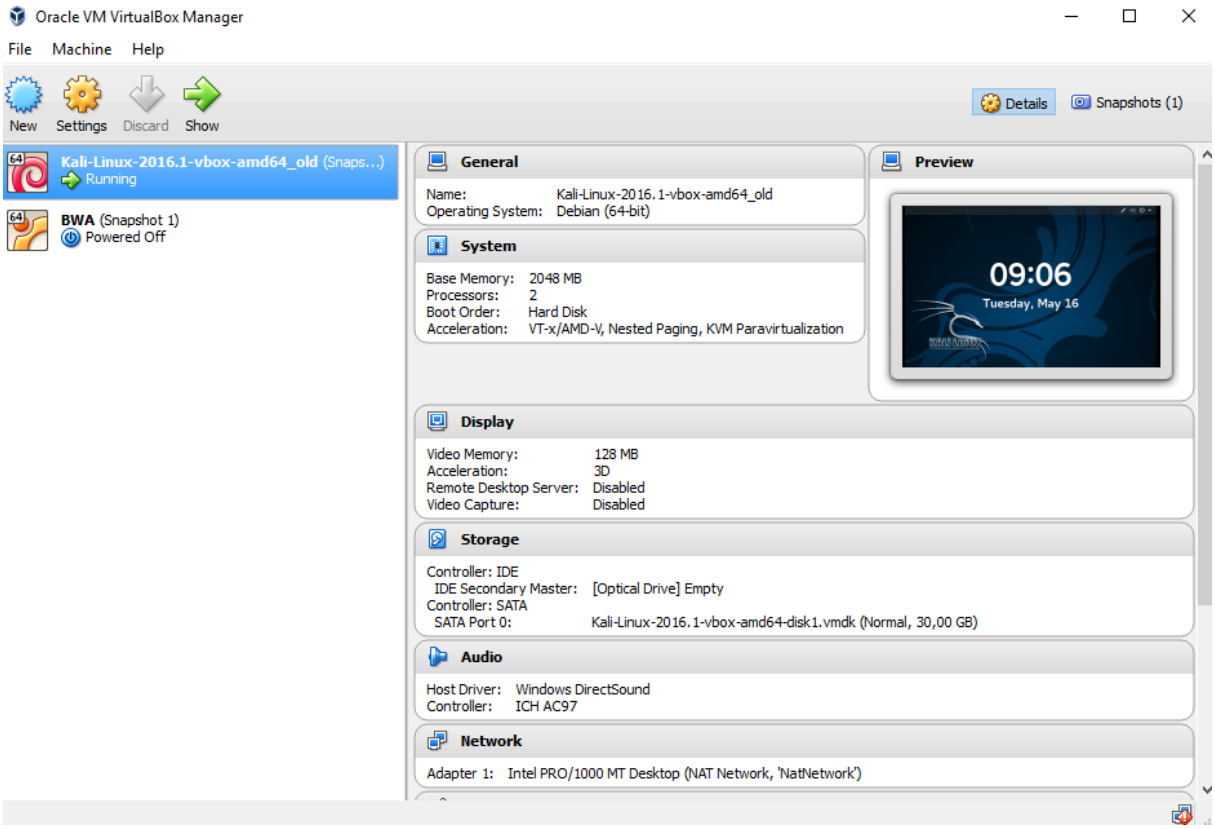
- Είναι διαθέσιμο για τα σημαντικότερα λειτουργικά συστήματα (Windows, Linux, και MacOS)
- Λειτουργεί ως αυτοματοποιημένος Crawler και ανιχνευτής ευπάθειας
- Διαθέτει ανίχνευση ιστόσελιδων
- Διαθέτει παρακολούθηση του διακομιστή μεσολάβησης
- Διαθέτει SSL MITM
- Κάνει ανάλυση περιεχομένου
- Διαθέτει επεκτασιμότητα μέσω API Javascript
- Εμπεριέχει προσαρμόσιμες ειδοποιήσεις
- Λειτουργεί ως βάση δεδομένων και μοντέλο κοινόχρηστων δεδομένων

Κεφάλαιο 4
Αξιολόγηση

Στο παρόν κεφάλαιο αναπτύσσουμε την υποδομή του εργαστήριου που δημιουργήσαμε για εύρεση ευπαθειών εφαρμογών ιστού στην πλατφόρμα δοκιμών WackoPicko με γνώστες και δημοσιευμένες ευπάθειες. Θα αναλύσουμε τις γνώστες ευπάθειες της σουίτας δοκίμων και θα παραθέσουμε τα αποτελέσματα των δοκιμών με πίνακες και διαγράμματα.

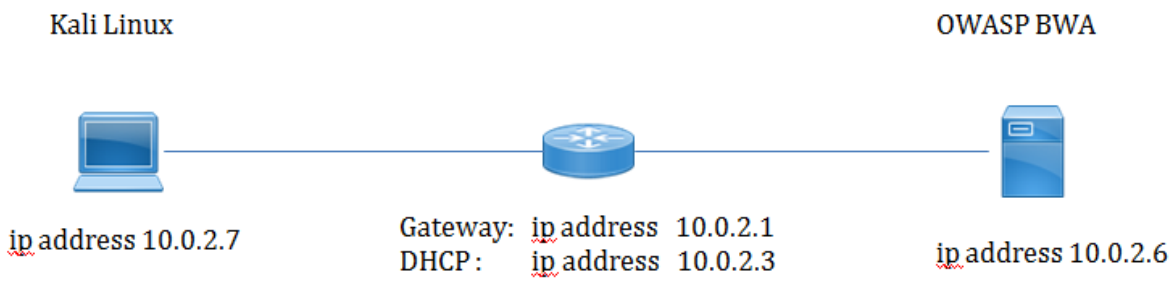
4.1 Δημιουργία του Εργαστηρίου - Υλικοτεχνική υποδομή

Το εργαστήριο που δημιουργήθηκε αποτελείτε από ένα ηλεκτρονικό υπολογιστή με λειτουργικό σύστημα Windows 10 Professional με επεξεργαστή Intel i3-4160 στα 3,6GHz και 8 GB μνήμης RAM. Εγκαταστάθηκε η εφαρμογή Virtual Box της Oracle. Στην εφαρμογή Virtual Box δημιουργήθηκαν δυο εικονικές μηχανές και εσωτερικό δίκτυο (Nat Network) που δεν επηρέαζε τα δεδομένα και την κίνηση του δικτυού. Η κίνηση του δικτύου μετρήθηκε μέσω του αναλυτή πρωτοκόλλου δικτύου Wireshark έκδοσης 2.2.2.



Σχήμα 4.11: Εφαρμογή Virtual Box

Η εικονική μηχανή που θα χρησιμοποιήσουμε για την διεξαγωγή της επίθεσης έχει λειτουργικό σύστημα kali linux 2016 v.1, διεύθυνση ip 10.0.2.7 και μνήμη 2GB. Η εφαρμογή με ευπάθειες που θα ελεγχθεί είναι εγκατεστημένη στην εικονική μηχανή OWASPBWA v1.2 [08] έχει ip 10.0.2.6, μνήμη 1GB . Το δίκτυο είναι class c (10.0.2.0/24) με Default Gateway 10.0.2.1 και DHCP Server 10.0.2.3

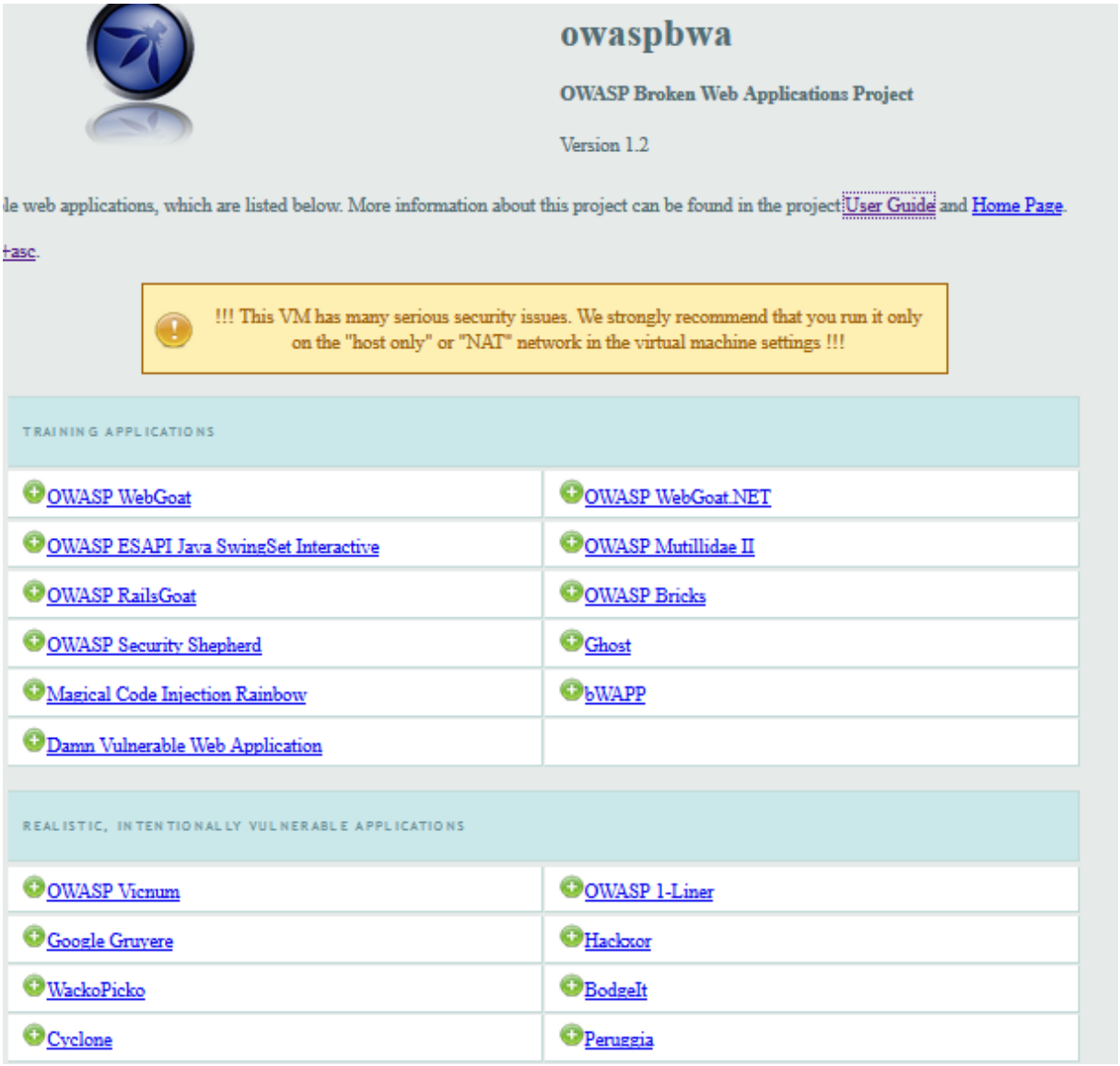


4.12: Σχηματική Παράσταση Τοπολογίας Εργαστήριου

Το Kali Linux είναι μια διανομή Linux βασισμένη στο Debian με στόχο την δοκιμή διείσδυσης και έλεγχο ασφαλείας. Το Kali περιέχει εργαλεία τα οποία χρησιμοποιούνται στην ασφάλεια πληροφοριών, όπως δοκιμές διείσδυσης, έρευνα ασφάλειας, ηλεκτρονική εγκληματολογία και αντίστροφη μηχανική. Το Kali Linux αναπτύσσεται, χρηματοδοτείται και συντηρείται από την Offensive Security.

Σχήμα 4.13: kali linux

Το OWASP BWA (Broken Web Applications Project) είναι μια συλλογή εφαρμογών ιστού με ευπάθειες σε μια εικονική μηχανή. Οι παρακάτω εφαρμογές περιέχονται στην έκδοση 1.2.



Σχήμα 4.15: OWASP BWA Web Interface

Εφαρμογές για εκπαίδευση. Εφαρμογές σχεδιασμένες για μάθηση που καθοδηγούν τον χρήστη σε συγκεκριμένες, εκ προθέσεως τρωτά σημεία.

- OWASP WebGoat version 5.4+SVN (Java)
- OWASP WebGoat.NET version 2012-07-05+GIT (ASP.NET)
- OWASP ESAPI Java SwingSet Interactive version 1.0.1+SVN (Java)
- OWASP Mutillidae II version 2.6.24+SVN (PHP)
- OWASP RailsGoat (Ruby on Rails)

- OWASP Bricks version 2.2+SVN (PHP)
- OWASP Security Shepherd version 2.4+GIT (Java)
- Ghost (PHP)
- Magical Code Injection Rainbow version 2014-08-20+GIT (PHP)
- bWAPP version 1.9+GIT (PHP)
- Damn Vulnerable Web Application version 1.8+GIT (PHP)

Ρεαλιστικές, σκοπίμως ευάλωτες εφαρμογές. Εφαρμογές που έχουν ευρεία ποικιλία σκόπιμων τρωτών σημείων ασφαλείας, αλλά έχουν σχεδιαστεί για να φαίνονται και να λειτουργούν σαν μια πραγματική εφαρμογή.

- OWASP Vicnum version 1.5 (PHP/Perl)
- OWASP 1-Liner (Java/JavaScript)
- Google Gruyere version 2010-07-15 (Python)
- Hackxor version 2011-04-06 (Java JSP)
- WackoPicko version 2011-07-12+GIT (PHP)
- BodgeIt version 1.3+SVN (Java JSP)
- Cyclone Transfers (Ruby on Rails)
- Peruggia version 1.2 (PHP)

Παλιές εκδόσεις πραγματικών εφαρμογών. Εφαρμογές ανοιχτού κώδικα με ένα ή περισσότερα γνωστά ζητήματα ασφαλείας.

- WordPress 2.0.0 (PHP, released December 31, 2005) with plugins: * myGallery version 1.2
- Spreadsheet for WordPress version 0.6
- OrangeHRM version 2.4.2 (PHP, released May 7, 2009)
- GetBoo version 1.04 (PHP, released April 7, 2008)
- gtd-php version 0.7 (PHP, released September 30, 2006)
- Yazd version 1.0 (Java, released February 20, 2002)

- WebCalendar version 1.03 (PHP, released April 11, 2006)
- Gallery2 version 2.1 (PHP, released March 23, 2006)
- TikiWiki version 1.9.5 (PHP, released September 5, 2006)
- Joomla version 1.5.15 (PHP, released November 4, 2009)
- AWStats version 6.4 (build 1.814, Perl, released February 25,2005)

Ο λόγος που επιλέξαμε τις δυο εικονικές μηχανές για την δημιουργία του εργαστήριου είναι για το kali linux

- η πληθώρα των εργαλείων για δοκιμές διείσδυσης
- το εύρος των περιοχών που καλύπτουν τα εργαλεία
- η ενεργή κοινότητα
- και η συνεχής ενημερώσεις των εργαλείων

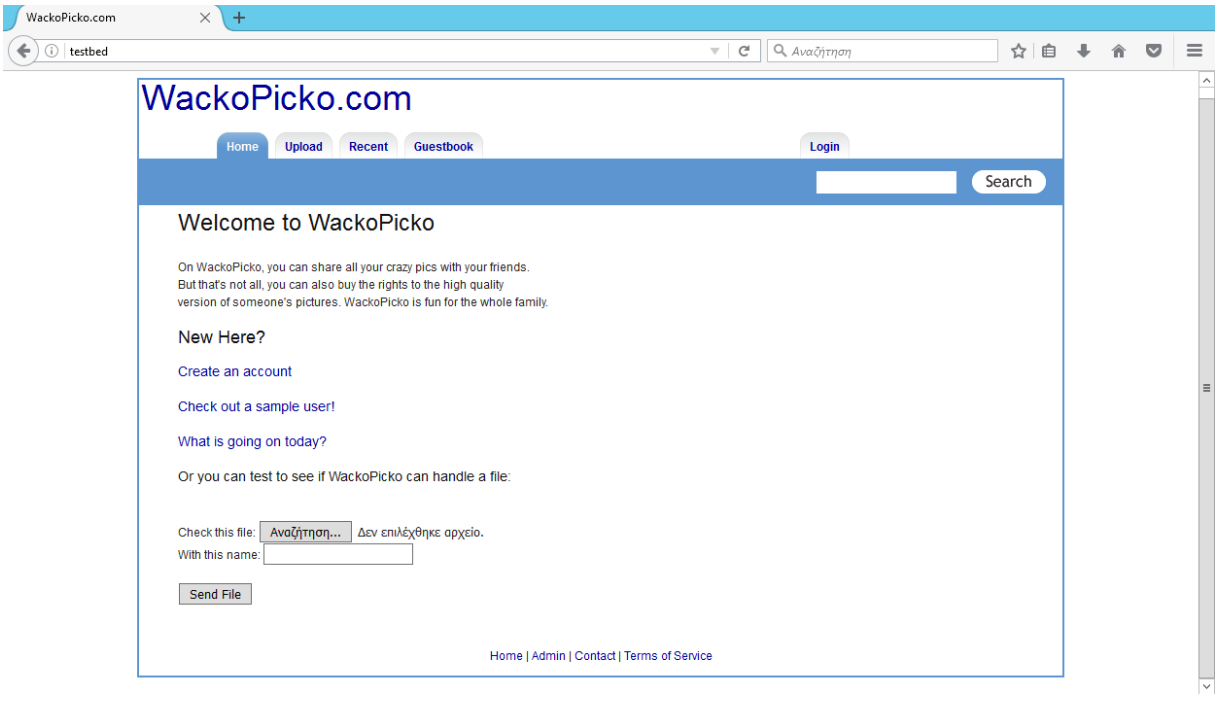
για το OWASP BWA

- η πληθώρα των ευπαθών εφαρμογών που περιέχει
- η πληθώρα των διαφορετικών τεχνολογιών υλοποίησης των εφαρμογών (php, .net, ruby, java)
- η ευκολία στην υλοποίηση
- η τεκμηρίωση

4.2 Η Πλατφόρμα Δοκιμών WackoPicko

Βασικό συστατικό της αξιολόγησης των εργαλείων διείσδυσης είναι η πλατφόρμα δοκιμών που θα ελεγχτεί. Η πλατφόρμα θα πρέπει να καλύπτει κάποιες προϋποθέσεις. Αρχικά να είναι εφαρμογή ιστού γιατί τα εργαλεία που θα εξεταστούν είναι εργαλεία διείσδυσης ιστού, να έχουν γνωστές αδυναμίες γιατί θα διαπιστωθεί η απόδοση ανίχνευσης, και να είναι αναπτυγμένη με πρόσφατη τεχνολογία για να είναι αντιπροσωπευτική των εφαρμογών ιστού που υπάρχουν σήμερα. Η πλατφόρμα δοκιμών που επιλέχτηκε από το OWASP BWA είναι η εφαρμογή Ιστού WackoPicko, αναπτύχτηκε από τον Adam Doupé. Η Εφαρμογή Ιστού WackoPicko έχει χρησιμοποιηθεί σε πλήθος δημοσιεύσεων [02, 06, 21] λόγω των χαρακτηριστικών που διαθέτει για την αξιολόγηση εργαλείων διείσδυσης σε εφαρμογές ιστού.

Η Εφαρμογή Ιστού WackoPicko είναι μια ιστοσελίδα διαμοιρασμού και αγοράς φωτογραφιών. Ένας χρήστης μπορεί να ανεβάσει φωτογραφίες, να αναζητήσει άλλων χρηστών φωτογραφίες, να τις σχολιάσει και να τις αγοράσει.



Σχήμα 4.13: WackoPicko

4.2.1 Τα Χαρακτηριστικά της Εφαρμογής WackoPicko

Η εφαρμογή WackoPicko έχει τα παρακάτω χαρακτηριστικά:

Πιστοποίηση. Οι χρήστες μπορούν να εγγραφούν στην ιστοσελίδα και να έχουν συγκεκριμένα δικαιώματα.

Ανέβασμα Φωτογραφιών. Όταν ένας εγγεγραμμένος χρήστης ανεβάσει φωτογραφίες άλλοι χρήστες μπορούν να γράψουν σχόλια και να τις αγοράσουν.

Σχόλια στις Φωτογραφίες. Όταν μια φωτογραφία ανεβαίνει στην Ιστοσελίδα, όλοι οι εγγεγραμμένοι χρήστες μπορούν να γράψουν σχόλια συμπληρώνοντας μια φόρμα, και τα σχόλια άλλων των χρηστών θα σχετίζονται με την φωτογραφία.

Αγορά φωτογραφιών. Ένας εγγεγραμμένος χρήστης μπορεί να αγοράσει μια φωτογραφία με την χρήση κάρτας αγοράς.

Αναζήτηση. Η εφαρμογή διαθέτει γραμμή αναζήτησης που αναζητά με βάση τις ετικετες που έχουν προσθέσει σε κάθε φωτογραφία.

Βιβλίο Επισκεπτών. Για την λήψη ανατροφοδότησης υπάρχει βιβλίο επισκεπτών με πεδία χρήστη και σχολίων

Σελίδα Διαχειριστή. Υπάρχει σελίδα διαχείριση με ειδικά δικαιώματα, όπως διαγραφή χρήστη και αλλαγή ετικετών φωτογραφιών.

4.2.2 Ευπάθειες της Εφαρμογής Ιστού WackoPicko.

Οι ευπάθειες της πλατφόρμας δοκιμών WackoPicko είναι οι παρακάτω[02, 21].

Reflected XSS. http://localhost/pictures/search.php?query=blah. Είναι μια ευπάθεια XSS στην σελίδα αναζήτησης που είναι προσβάσιμη χωρίς την σύνδεση του χρήστη στην εφαρμογή.

Stored XSS. http://localhost/guestbook.php. Είναι μια αποθηκευμένη (Stored) ή επίμονη (Persistent) XSS ευπάθεια στην σελίδα του βιβλίου επισκεπτών. Στο πεδίο comment μπορεί ο επιτιθέμενος να δημιουργήσει κώδικα javascript και να εκμεταλλευτεί την ευπάθεια.

SessionID vulnerability. http://localhost/admin/login.php. Η πληροφορία του session που σχετίζεται με τον λογαριασμό του διαχειριστή χειρίζεται διαφορετικά σε σχέση με τους απλούς χρήστες. Έτσι ο διαχειριστής χρησιμοποίει ένα εξατομικευμένο session cookie που παρακολουθεί τα session. Όμως η τιμή που δίνεται στο session cookie δεν είναι τυχαία αλλά αύξουσα και ένας επιτιθέμενος μπορεί να μαντέψει την ταυτότητα του session και να έχει πρόσβαση στην εφαρμογή με δικαιώματα διαχειριστή.

Weak username/password. https://localhost/admin/login.php. Η σελίδα που περιέχει τον λογαριασμό του διαχειριστή έχει εύκολο συνδυασμό ονόματος χρήστη και κωδικό admin/admin

Reflected SQL Injection. http://localhost/users/login.php. Η αντανακλώμενη SQL ευπάθεια στο πεδίο username στην φόρμα σύνδεσης. Μπορούμε να εκτελέσουμε τυχαία ερωτήματα και να πάρουμε τα ονόματα χρηστών και κωδικών από την βάση.

Command-line Injection. http://localhost/passcheck.php. Η εφαρμογή παρέχει μια υπηρεσία ελέγχου κωδικών που βρίσκεται σε λεξικό. Η παράμετρος του password μπορεί να αιτηθεί χωρίς προστασία. Η αδυναμία αυτή μπορεί να εκμεταλλευτεί εάν στην παράμετρο του password προστεθεί το $.

File Inclusion. http://localhost/admin/index.php?page=login. Το περιβάλλον του διαχειριστή μπορεί να προσπελαθεί μέσω της σελίδας index.php. Η σχεδίαση αυτή περιέχει ευπάθεια ενσωμάτωσης αρχείου. Ένας επιτιθέμενος μπορεί να εκμεταλλευτεί αυτή την ευπάθεια και να εκτελέσει απομακρυσμένο κώδικα PHP π.χ. http://hacker/blah.php%00 ως παράμετρο της σελίδας αντί για index.php. Το %00 στο τέλος του string έχει ως αιτία να αγνοηθεί το ".php" στην

παράμετρο της σελίδας και αντί για index.php να κατεβάσει και να εκτελέσει κώδικα που βρίσκεται στο http://hacker/blah.php

Reflected XSS Behind JavaScript. http://localhost/piccheck.php. Η αρχική σελίδα έχει μια φόρμα που ελέγχει εάν ένα αρχείο έχει το σωστό format για το επεξεργαστεί η εφαρμογή. Η φόρμα έχει δυο παραμέτρους μια για το αρχείο και μια για το όνομα. Εάν το αρχείο ανέβει το επιστρέφει πίσω στον χρήστη και αυτό δημιουργεί αντανακλώμενη (Reflected) ευπάθεια XSS. Παρόλα αυτά επειδή η φόρμα παράγεται δυναμικά μέσω javascript, αποτρέπει ένα crawler να βρίσκει το URL μέσω απλής ταυτοποίησης μοτίβου.

Parameter Manipulation. http://localhost/users/sample.php?userid=1. Η αρχική σελίδα παρέχει έναν σύνδεσμο σε μια δοκιμαστική σελίδα. Ο σύνδεσμος χρησιμοποιεί την παράμετρο "userid" GET για να δει τον δοκιμαστικό χρήστη που έχει id 1. Ο επιτιθέμενος μπορεί να έχει πρόσβαση σε όλους τους χρήστες παραποιώντας αυτή την μεταβλητή.

Stored SQL Injection. http://localhost/users/register.php -> http://localhost/users/similar.php. Όταν οι χρήστες δημιουργούν έναν λογαριασμό πρέπει να δώσουν το όνομα τους. Αυτή η τιμή συγκρίνεται με τα ονόματα άλλων χρηστών για βρεθούν ίδια ονόματα. Ένας επιτιθέμενος μπορεί να εκμεταλλευτεί αυτή την αποθηκευμένη (Stored) ευπάθεια δημιουργώντας έναν χρήστη με όνομα "' ; DROP users;#" και να επισπευτεί άλλες σελίδες χρηστών.

Directory Traversal. http://localhost/pictures/upload.php. Όταν ανεβαίνει μια φωτογραφία η εφαρμογή αντιγράφει το αρχείο σε έναν υποκατάλογο του καταλόγου upload. Το όνομα του υποκαταλόγου είναι η ετικέτα του αρχείου που ανέβηκε από τον χρήστη. Ένας επιτιθέμενος μπορεί να παραποιήσει την ετικέτα του αρχείου και να εκτελέσει μια επίθεση όδευσης καταλόγου. Επίσης με την χρήση "../../" ο επιτιθέμενος μπορεί έχει πρόσβαση και σε αρχεία εκτός του καταλόγου upload.

Multi-Step Stored XSS http://localhost/pictures/view.php?picid=3. Αντίστοιχα με την αποθηκευμένη (Stored) ή επίμονη (Persistent) ευπάθεια XSS που βρίσκεται στο βιβλίο επισκεπτών τα σχόλια στην φωτογραφία είναι ευπαθές σε επιθέσεις XSS. Όμως η συγκεκριμένη ευπάθεια εκμεταλλεύεται πιο δύσκολα γιατί ο χρήστης πρέπει να είναι συνδεδεμένος και να επιβεβαίωση τα σχόλια της φωτογραφίας.

Forceful Browsing http://localhost/pictures/highquality.php?picid=3&key=highquality. Ένα από τα βασικά συστατικά της εφαρμογής είναι η αγορά φωτογραφιών. Όμως η πρόσβαση στην

αγορά φωτογραφιών δεν έχει ελεγχτεί με αποτέλεσμα ο επιτιθέμενος να αποκτήσει τις φωτογραφίες με την γνώση του URL χωρίς να έχει συνδεθεί και να τις έχει αγοράσει.

Logic Flaw. http://localhost/cart/review.php. Το σύστημα παροχής κουπονιών πάσχει από λογική ατέλεια και ένα κουπόνι έκπτωσης μπορεί να χρησιμοποιηθεί πολλές φορές μειώνοντας το κόστος της αγοράς της φωτογραφίας.

Reflected XSS Behind a Flash Form. http://localhost/submitname.php. Στην αρχική σελίδα υπάρχει μια φόρμα σε flash που ρωτά τον χρήστη για το αγαπημένο του χρώμα. Η σελίδα με τα αποτελέσματα έχει αντανακλώμενη (Reflected) ευπάθεια XSS. Η παράμετρος "value" επιστρέφει χωρίς να έχει απομονωθεί.

4.3 Διεξαγωγή του Έλεγχου.

Παρακάτω παρατίθενται οι ρυθμίσεις των εργαλείων και οι εντολές που χρησιμοποιήθηκαν κατά την διεξαγωγή του ελέγχου.

Kali linux. Ενημέρωση της διανομής με τις τελευταίες εκδόσεις.

apt-get update

apt-get dist-upgrade

Στα εργαλεία Arachni, Nikto και Skipfish η σάρωση πραγματοποιήθηκε μέσω γραμμών εντολών οι οποίες παραθέτονται παρακάτω:

Arachni. Διεξαγωγή του ελέγχου και αποθήκευση της αναφοράς σε μορφή afr στο φάκελο της εφαρμογής root με όνομα wackopicko.afr

arachni --output-verbose --scope-include-subdomains http://10.0.2.6/WackoPiocko --report-save-path=wackopicko.afr

μετατροπή της αναφοράς από μορφή afr σε html

arachni_reporter wackopicko.afr --reporter=html:outfile=wackopicko.html.zip

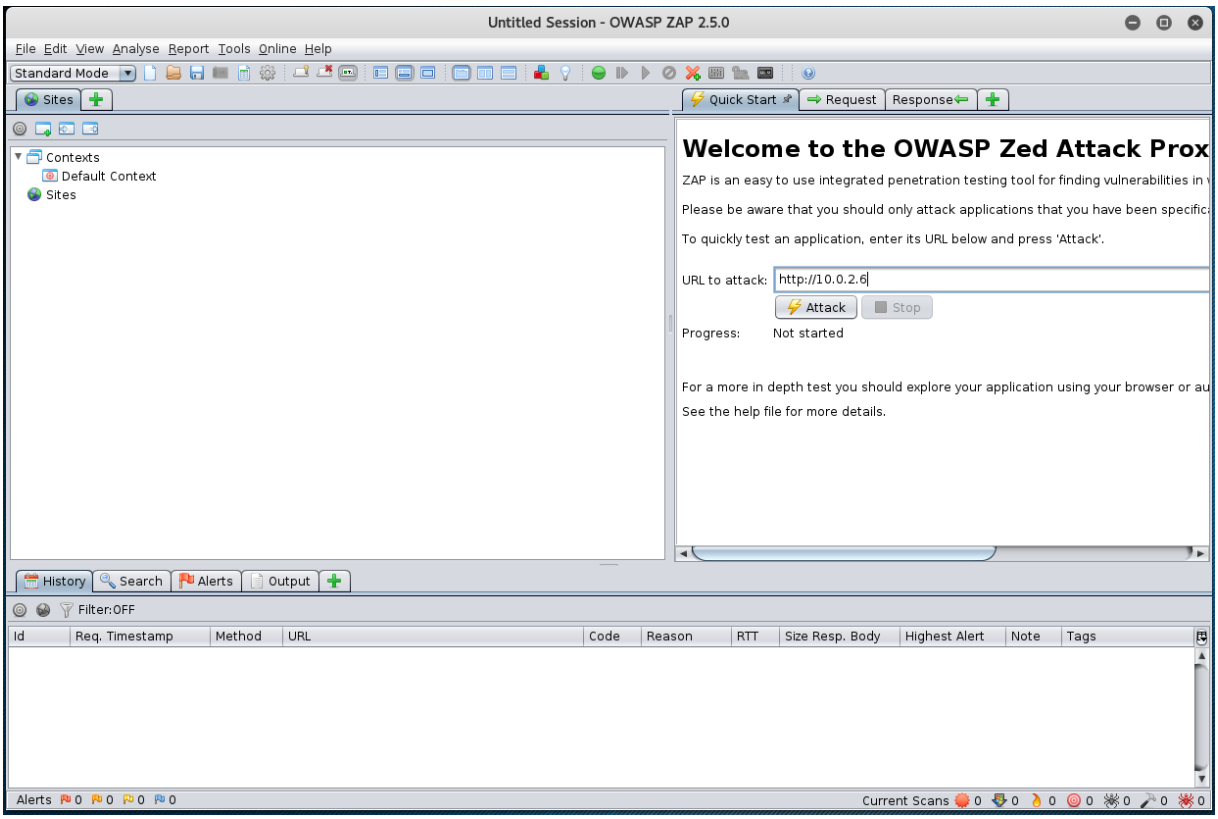
Nikto. Διεξαγωγή του ελέγχου και αποθήκευση της αναφοράς σε μορφή html στο φάκελο root με όνομα nikto_wackopicko.html

Nikto –h http://10.0.2.6/WackoPicko -output nikto_wackopicko.html

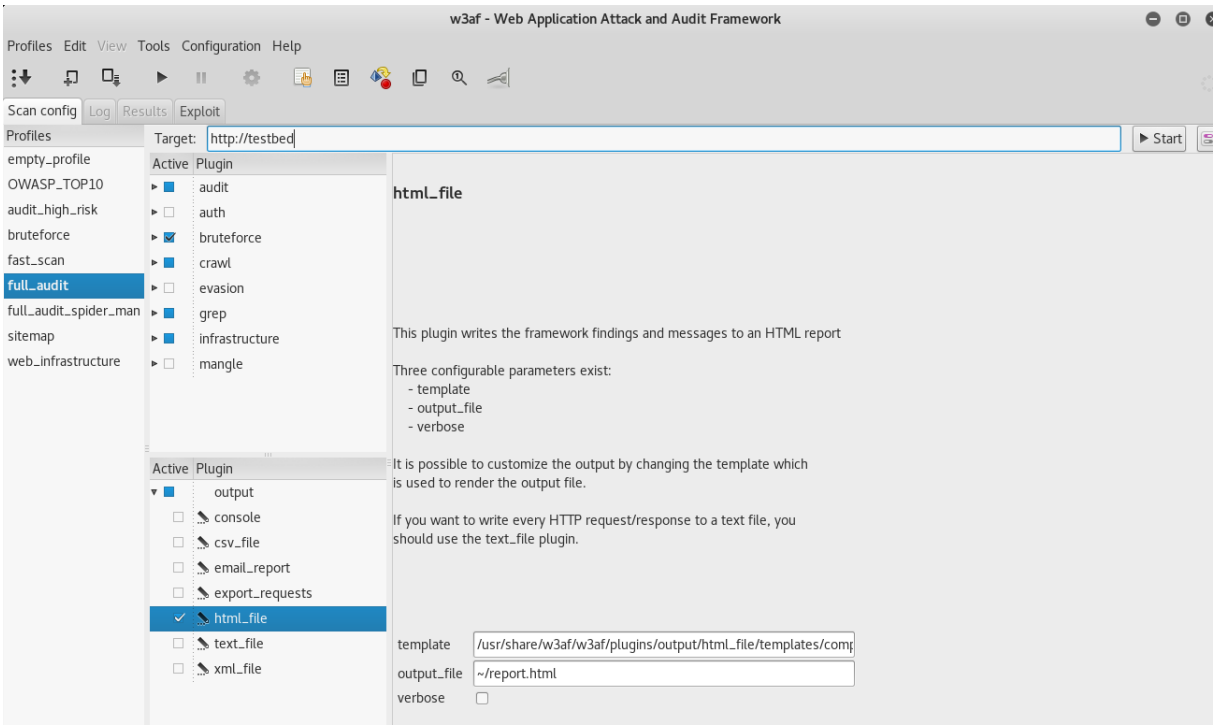
Skipfish. Διεξαγωγή του ελέγχου και αποθήκευση της αναφοράς μορφή html στην διαδρομή /root/skipfish/wackopicko

Skipfish –o /root/skipfish/wackopicko http://10.0.2.6/WackoPicko

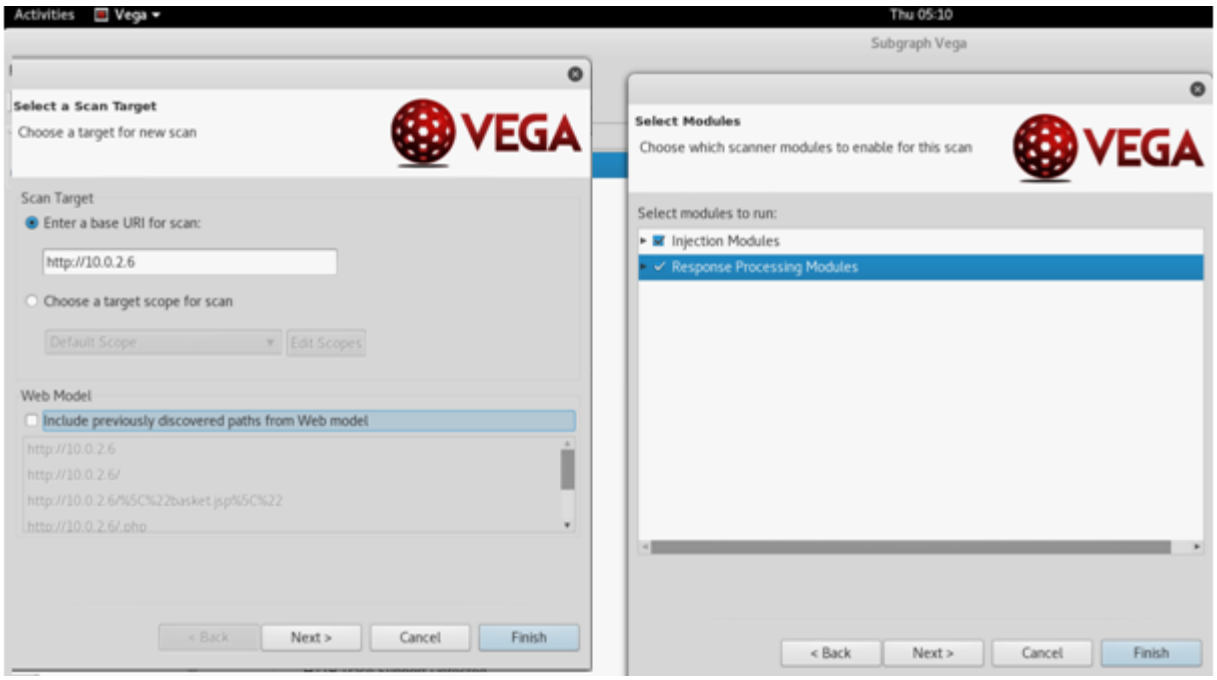
Στα εργαλεία OWASP ZAP, W3AF και VEGA η σάρωση πραγματοποιήθηκε μέσω γραφικό περιβάλλον και οι ρυθμίσεις οι οποίες παρουσιάζονται στις παρακάτω εικόνες:



Σχήμα 4.14: Ρύθμιση Σάρωσης OWASP ZAP



Σχήμα 4.16: Ρύθμιση Σάρωσης Full Audit W3AF



Σχήμα 4.15 : Ρύθμιση Σάρωσης VEGA και των δυο Modules

4.4 Αποτελέσματα Ελέγχων

Οι δοκιμές διεξήχθηκαν σε ελεγχόμενο περιβάλλον, πραγματοποιήθηκαν τουλάχιστον τρις φορές για την ακρίβεια και την εγκυρότητα των μετρήσεων. Στις ενότητες 4.4.1-4 παραθέτουμε τα αποτελέσματα των μετρήσεων με σχήματα και πινάκες.

Στον πινάκα 4.10 συγκρίνουμε τα εργαλεία ανοιχτού κώδικα με βάση την χρησιμότητα τους την μέθοδο σάρωσης που χρησιμοποιούν και τα αποτελέσματα των εξόδων τους [24].

Α/Α	Σαρωτής	Κλίμακα χρησιμότητας					Μέθοδος σάρωσης			Έξοδος	
		Γραφικό περιβάλλον	spider	Σταθερότητα	Ευχρηστία	Παραμετροποίηση	Manual crawl	Ανάλυση αρχείων	Απόδοση	logging	Αναφορά
1	W3AF	Ναι	Ναι	Εύθραυστο	Σύνθετη	Σύνθετη	Ναι	Όχι	Αργή	Ναι	Ναι
2	Nikto	Όχι		Σταθερό	Πολύ απλή	Απλή			Γρήγορη	Ναι	Ναι
3	OWASP ZAP	Ναι	Ναι	Πολύ Σταθερό	Πολύ απλή	Πολύ απλή	Ναι	Όχι	Γρήγορη	Ναι	Ναι
4	Skipfish	Όχι	Ναι	Σταθερό	Σύνθετη	Απλή	Όχι	Ναι	Γρήγορη	Ναι	Ναι
5	VEGA	Ναι		Σταθερό	Πολύ απλή	Πολύ απλή			Πολύ γρήγορη	Ναι	Όχι
6	Arachni	Ναι	Ναι	Σταθερό	Πολύ απλή	Απλή	Ναι	Όχι	Γρήγορη	Όχι	Ναι

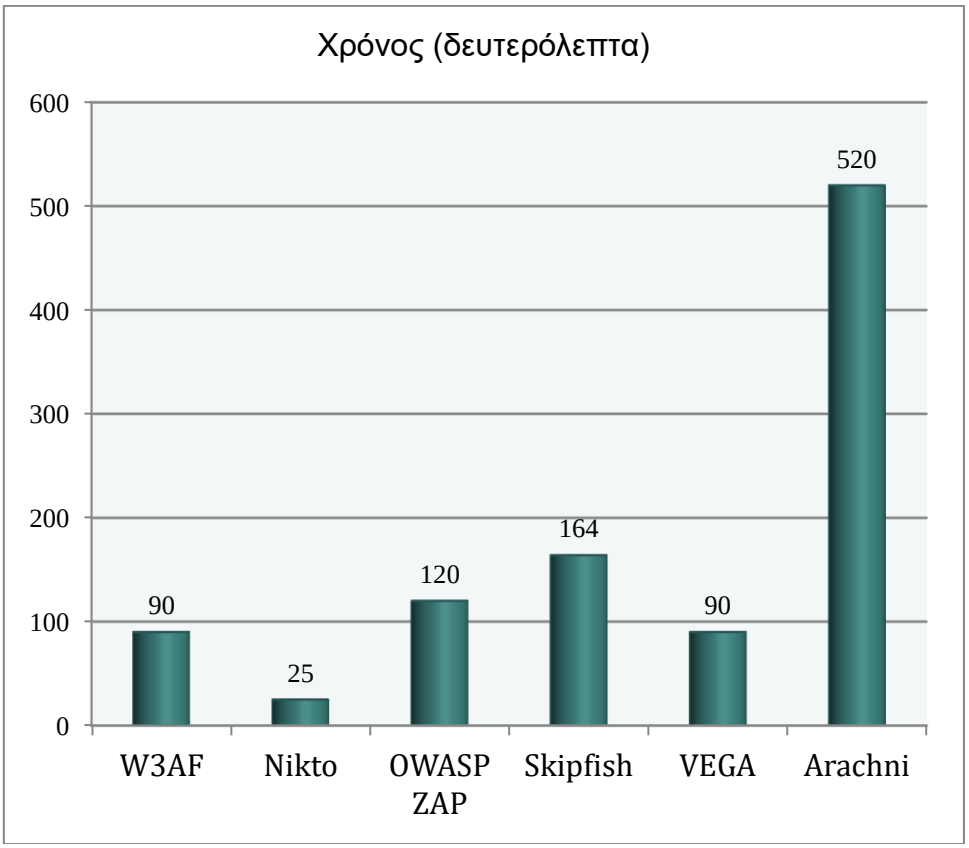
Πινάκας 4.10: Πινάκας Σύγκρισης Σαρωτών Ανοιχτού Κώδικα

4.4.1 Χρόνος σάρωσης

Στον πίνακα 4.11 αποτυπώνουμε τον χρόνο που απαιτείται για την σάρωση της σουίτας δοκιμών WackoPicko. Ο χρόνος που απαιτήθηκε κυμαίνεται από 25 έως 520 σε δευτερόλεπτα με ποιο γρήγορο το Nikto και ποιο αργό το Arachni, χωρίς αυτό όμως να σημαίνει κάτι για τα απόδοση των εργαλείων σάρωσης.

Α/Α	Σαρωτής	Χρόνος (δευτερόλεπτα)
1	W3AF	90
2	Nikto	25
3	OWASP ZAP	120
4	Skipfish	164
5	VEGA	90
6	Arachni	520

Πινάκας 4.11: Πινάκας Χρόνου Εκτέλεσης της Δοκιμής



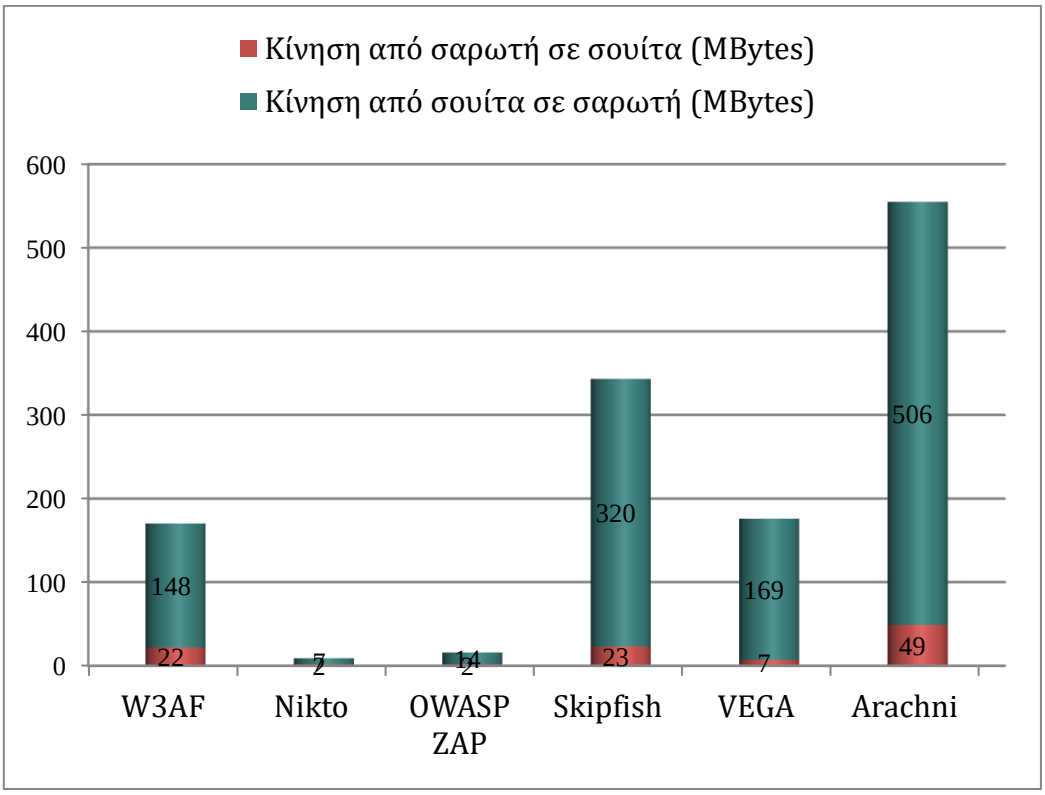
Σχήμα 4.17: Γραφική Παράσταση Χρόνου Σάρωσης

4.4.2 Αποτύπωμα Δικτυακής Κίνησης

Στον πίνακα 12 παραθέτουμε τον αριθμό των bytes που στάλθηκαν και λήφθησαν από κάθε εργαλείο στην σουίτας δοκιμών WackoPicko και μετρήθηκαν από τον αναλυτή πρωτοκόλλου wireshark. Το εργαλείο με την μικρότερη κίνηση δικτύου είναι το Nikto και με την μεγαλύτερη το Arachni.

Α/Α	Σαρωτής	Σύνολο κίνησης (MBytes)	Κίνηση από σαρωτή σε σουίτα (MBytes)	Κίνηση από σουίτα σε σαρωτή (MBytes)
1	W3AF	170	2	1
2	Nikto	9	2	7
3	OWASP ZAP	16	2	14
4	Skipfish	343	23	320
5	VEGA	176	7	169
6	Arachni	555	49	506

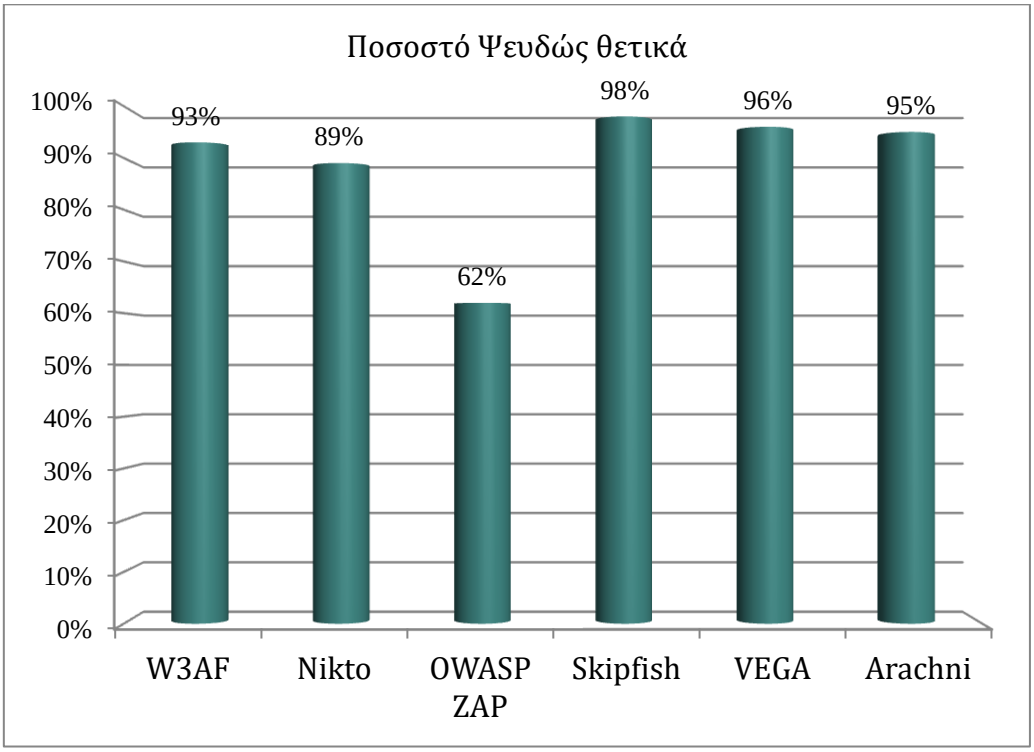
Πινάκας 4.12: Πινάκας Αποτυπώματος Δικτυακής Κίνησης



Σχήμα 4.18: Γραφική Παράσταση Συνόλου Κίνησης Δεδομένων σε MBytes

4.4.3 Ψευδώς Θετικά Αποτελέσματα

Στον πίνακα 4.13 απεικονίζονται ο αριθμός των ψευδώς θετικών αποτελεσμάτων και το ποσοστό τους επί τις εκατό. Ο αριθμός τους κυμαίνεται από οχτώ ως και εκατόν σαράντα οχτώ και το ποσοστό τους αντίστοιχα από 62% έως 98%. Ο μεγάλος αριθμός ψευδώς θετικών είναι κυρίως ενημερωτικού χαρακτήρα όπως αποτυπώθηκε στης αναφορές των εργαλείων και όχι υψηλού ή μέσου κίνδυνου. Όμως παρόλα αυτά ο αριθμός τους είναι μεγάλος και αποδεικνύει ότι τα εργαλεία ανοιχτού κώδικα πάσχουν από ψευδώς θετικά αποτελέσματα.



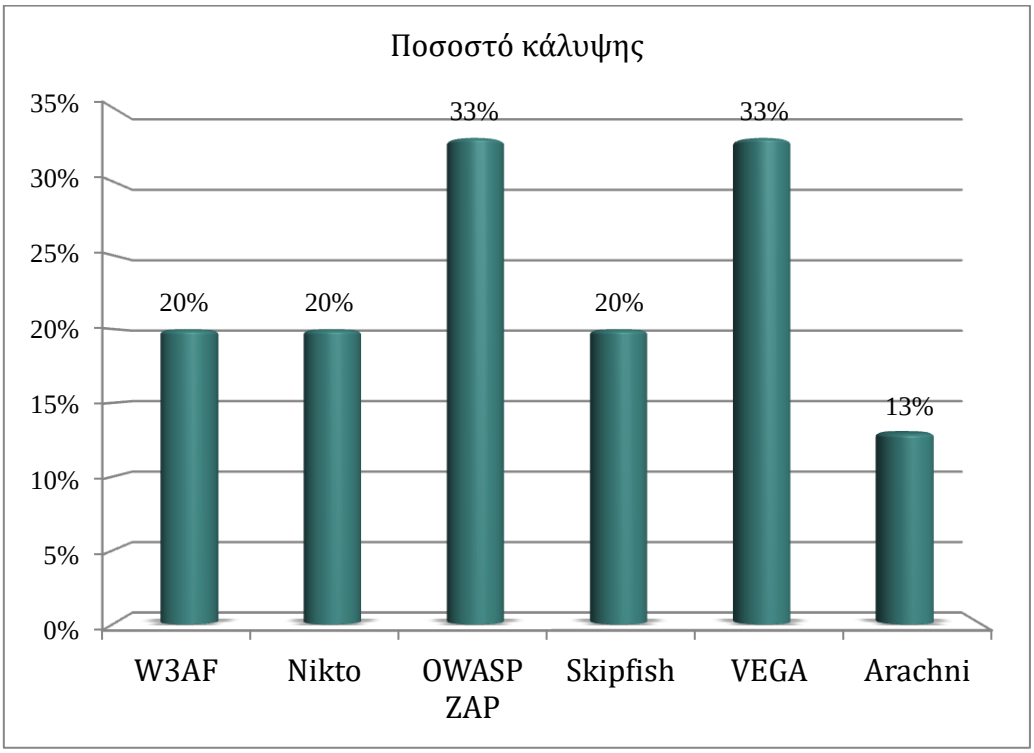
Σχήμα 4.19: Γραφική Παράσταση Ποσοστού Ψευδώς Θετικών

4.4.4 Απόδοση Σάρωσης

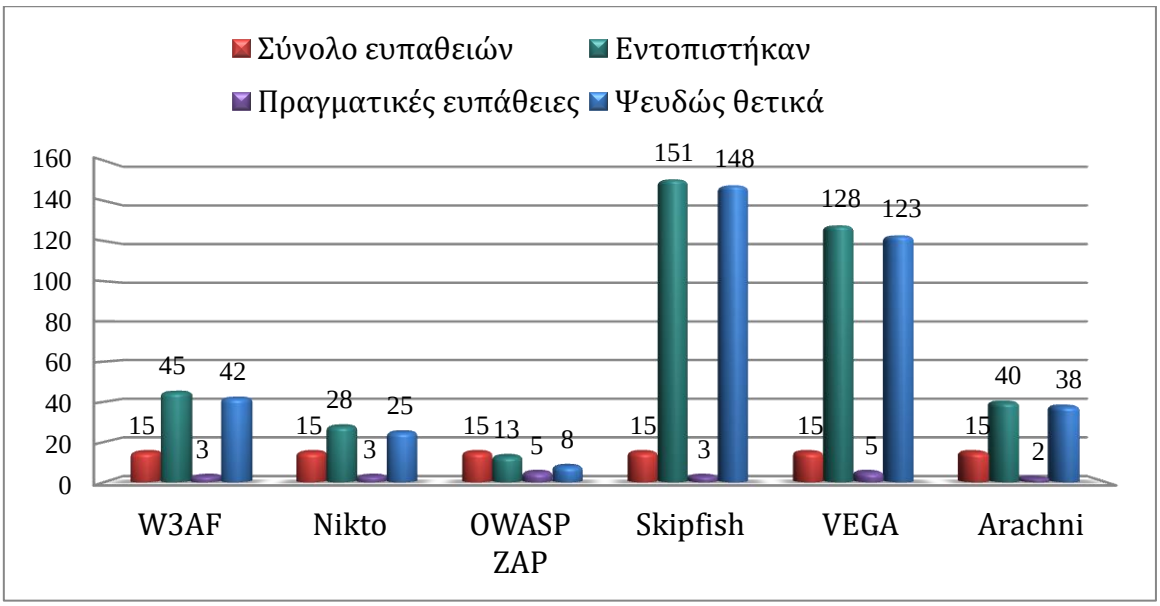
Στον πίνακα 4.13 παρουσιάζουμε το συνολικό ποσοστό κάλυψης των εργαλείων. Το ποσοστό κάλυψης είναι ο αριθμός των γνωστών ευπαθειών ως προς το σύνολο των ευπαθειών που βρέθηκαν. Το ποσοστό κυμαίνεται από δεκατρία έως τριάντα τρία τις εκατό. Αν και το ποσοστό είναι μικρό τα εργαλεία έχουν σημαντικές επιτυχίες στις ευπάθειες XSS και λιγότερο στις SQLi όπως βλέπουμε στον πίνακα 4.14.

Α/Α	Σαρωτής	Σύνολο ευπαθειών	Εντοπιστήκαν	Πραγματικές ευπάθειες	Ποσοστό κάλυψης	Ψευδώς θετικά	Ποσοστό Ψευδώς θετικά
1	W3AF	15	45	3	20%	42	93%
2	Nikto	15	28	3	20%	25	89%
3	OWASP ZAP	15	13	5	33%	8	62%
4	Skipfish	15	151	3	20%	148	98%
5	VEGA	15	128	5	33%	123	96%
6	Arachni	15	40	2	13%	38	95%

Πινάκας 4.13: Πινάκας Ποσοστών Ανίχνευσης και Ψευδώς Θετικών Αποτελεσμάτων



Σχήμα 4.20: Γραφική Παράσταση Ποσοστού Κάλυψης



Σχήμα 4.21: Γραφική Παράσταση Συνόλου Ευπαθειών

Ο πίνακας 4.14 μας δείχνει ποιες από τις ευπάθειες της σουίτας δοκιμών WackoPicko έχουν ανιχνευτεί από ποιο εργαλείο σάρωσης.

A/A	Ευπάθειες	W3AF	Arachmi	ZAP	Skipfish	Nikto	Vega
1	Reflected XSS	x	x	x	x		x
2	Stored XSS			x		x	x
3	SessionID vulnerability					x	
4	Stored SQL Injection			x	x		
5	Reflected SQL Injection	x		x	x		x
6	Directory Traversal						
7	Multi-Step Stored XSS						
8	Forceful Browsing			x			
9	Command-line Injection						x
10	File Inclusion	x					x
11	Parameter Manipulation						
12	Reflected XSS Behind JavaScript		x				
13	Logic Flaw						
14	Reflected XSS Behind a Flash Form						
15	Weak username/password					x	

Πινάκας 4.14: Αναλυτικός Πινάκας Ευπαθειών ανά Εργαλείο

Κεφάλαιο 5
Συμπεράσματα

Η ανάλυση των μετρήσεων οδήγησε στα παρακάτω συμπεράσματα.

Το ZAP και το VEGA είναι τα μόνα εργαλεία που βρήκαν αντανακλώμενη και αποθηκευμένη XSS ευπάθεια. Όλα βρήκαν τουλάχιστον μια ευπάθεια XSS. Το Arachni είναι το μονό που βρήκε και τις δυο αντανακλώμενες XSS ευπάθειες. Το Nikto είναι το μονό που δεν βρήκε την αντανακλώμενη XSS ευπάθεια. Γενικά τα εργαλεία είχαν πολύ καλή συμπεριφορά στην εύρεση ευμαθειών XSS.

Όλα σχεδόν τα εργαλεία ανίχνευσαν την αντανακλώμενη SQLi ευπάθεια. Το OWASP ZAP και το Skipfish βρήκαν όλες τις ευπάθειες SQLi, δηλαδή την αντανακλώμενη και την αποθηκευμένη. Αντίθετα τα Nikto και η Arachni δεν ανίχνευσαν καμία SQLi ευπάθεια. Οπότε η συμπεριφορά των εργαλείων στην ανεύρεση ευπαθειών SQLi κρίνεται θετική.

Τα εργαλεία χρειάστηκαν κάτω από τρία λεπτά για την ολοκλήρωση του έλεγχου και ο όγκος των δεδομένων που διακινήθηκε ήταν χαμηλός. Εξαίρεση αποτελεί το εργαλείο Arachni που πλησίασε τα δέκα λεπτά και ο όγκος των δεδομένων ξεπέρασε το μισό gigabyte. Αν συμπεριλάβουμε και τα αποτελέσματα της ανίχνευσης και τα ψευδώς θετικά αποτελέσματα το εργαλείο κατατάσσεται στην χειρότερη θέση από άποψη επιδόσεων.

Τα εργαλεία OWASP ZAP και VEGA βρήκαν τις περισσότερες ευπάθειες. Όμως το OWASP ZAP είχε καλύτερη απόδοση στις XSS και SQLi ευπάθειες όπως και τα λιγότερα ψευδώς θετικά αποτελέσματα που το κατατάσσει στο πιο αποτελεσματικό εργαλείο.

Συνολικά κανένα εργαλείο δεν σημείωσε μεγάλα ποσοστά ανίχνευσης, και από το σύνολο των δεκαπέντε ευπαθειών πέντε ευπάθειες δεν ανιχνευτήκαν από κανένα εργαλείο. Επίσης ο μέσος όρος των ψευδός θετικών αποτελεσμάτων προσεγγίζει το 90%. Από τα παραπάνω αποδεικνύεται ότι η κάλυψη είναι μικρή και τα ψευδός θετικά αποτελέσματα υψηλά [09].

Επίλογος

Η προστασία των εφαρμογών ιστού είναι μια συνεχής πρόκληση. Νέες ευπάθειες αλλά και νέες τεχνικές εκμετάλλευσης ευπαθειών αποκαλύπτονται όλο και πιο συχνά. Γι τον λόγο αυτό, οι εφαρμογές ιστού δεν πρέπει να διατίθενται ή να τροποποιούνται στο διαδίκτυο χωρίς να έχει διεξαχθεί έλεγχος για την ύπαρξη ευπαθειών. Ο εντοπισμός των ευπαθειών γίνεται είτε με την ανάλυση του κώδικα είτε με την χρήση δοκιμών διείσδυσης.

Η παρούσα διπλωματική εργασία, ασχολήθηκε με την ανάλυση των σημαντικότερων ευπαθειών των εφαρμογών ιστού με παραδείγματα και τρόπους αντιμετώπισης. Προτάθηκε η χρήση εργαλείων διείσδυσης για το εντοπισμό των ευπαθειών και δημιουργήθηκε ένα εργαστήριο για την διεξαγωγή δοκιμών διείσδυσης. Αξιολογήθηκαν έξι εργαλεία διείσδυσης ανοιχτού κώδικα πραγματοποιώντας δόκιμες διείσδυσης από μια εφαρμογή ιστού με γνωστές ευπάθειες. Ελέγχθηκε η απόδοση των εργαλείων μετρώντας τον χρόνο που απαιτείται για την ολοκλήρωση της δοκιμής, την κίνηση των δεδομένων του δικτύου που στέλνονται και λαμβάνονται, τις ευπάθειες που ανιχνευτήκαν και τα ψευδώς θετικά αποτελέσματα.

Τα εργαλεία που χρησιμοποιήθηκαν για τον σκοπό αυτό είναι τα Arachni, Vega, Skipfish, W3AF, Nikto και OWASP ZAP. Η σουίτα δοκιμών που εξετάστηκε ήταν η WackoPicko. Η δικτυακή κίνηση μετρήθηκε από το πρωτόκολλο ανάλυσης πακέτων Wireshark.

Μελλοντική Ερευνά

Προτείνουμε για μελλοντική εργασία στην υπάρχον σουίτα δοκιμών WackoPicko να προσθέτουμε επιπλέον ευπάθειες. Οι ευπάθειες αυτές θα πρέπει να καλύπτουν το σύνολο των ευπαθειών του OWASP TOP 10. Υπάρχουν δημοσιευόμενες εργασίες με παραδείγματα ευπαθειών [02, 38, 41] ιστοσελίδες [33] και βιβλία [07] που μπορούμε να χρησιμοποιήσουμε για τον σκοπό αυτό.

Επίσης εάν χρησιμοποιηθεί μια δεύτερη σουίτα δοκιμών που έχει αναπτυχτεί σε γλώσσα προγραμματισμού asp .net όπως η hacme-bank [35] ή VulnApp [36] και προσθέτουν ευπάθειες

καλύπτοντας το σύνολο του OWASP TOP 10, θα βελτιώσουμε την κάλυψη των τεχνολογίων υλοποίησης στο 55,5% [34].

Καλύπτοντας τις δυο πιο δημοφιλής τεχνολογίες υλοποίησης php και .net και τις σημαντικότερες ευπάθειες σε εφαρμογές ιστού [25] θα έχουμε ένα πλαίσιο αναφοράς για να αξιολογούνται τα εργαλεία διείσδυσης.

Βιβλιογραφία

[01] Bau J, Bursztein E, Gupta D, Mitchell J (2010) State of the art: automated black-box web application vulnerability testing. In: Proceedings of the 2010 IEEE Symposium on Security and Privacy, SP '10, IEEE Computer Society, Washington, DC, USA, pp 332–345

[02] Doupé, A., Cova, M., & Vigna, G. (2010). Why Johnny can't pentest: An analysis of black-box web vulnerability scanners. In Detection of Intrusions and Malware, and Vulnerability Assessment (pp. 111-131). Springer Berlin Heidelberg.

[03] Alidoosti M., Mirjalili M., Nowroozi A. (2014). A survey on web penetration test, ACSIJ Advances in Computer Science: an International Journal, Vol. 3, Issue 6, No.12 , ISSN : 2322-5157

[04] Cody A, Orebaugh A, Scarfone K, Souppaya M, 2008. SP 800-115. Technical Guide to Information Security Testing and Assessment. Technical Report. NIST, Gaithersburg, MD, United States.

[05] Khari, M., Singh, N. (2014, May). An Overview of Black Box Web Vulnerability Scanners. In Computer Science and Software Engineering, International Journal of Advanced Research.

[06] A. L. Doupé, (2014) "Advanced Automated Web Application Vulnerability Analysis," Ph.D. Thesis, UNIVERSITY OF CALIFORNIA Santa Barbara.

[07] The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws, Second Edition, Dafydd Stuttard, Marcus Pinto

[08] https://www.owasp.org/index.php/OWASP_Broken_Web_Applications_Project. (πρόσβαση 29/04/2017.

[09] Baral P (2011) Web Application Scanners: A Review of Related Articles. IEEE Potentials 30(2):10-14

[10] Makino Y, Klyuev V (2015) Evaluation of web vulnerability scanners. In: Proceedings of the IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), vol 1, Warsaw, PL, pp 399–402

[11] https://www.acunetix.com/support/docs/wvs/overview/ (πρόσβαση 29/04/2017.

[12] Suteva N, Zlatkovski D, Mileva A (2013) Evaluation and testing of several free/open source web vulnerability scanners. In: Proceedings of the 10th Conference for Informatics and Information Technology (CIIT 2013), Bitola, MK, pp 221–224

[13] http://www-03.ibm.com/software/products/en/ (πρόσβαση 29/04/2017)

[14] https://portswigger.net/burp/ (πρόσβαση 28/04/2017)

[15] https://cirt.net/Nikto2 (πρόσβαση 28/04/2017)

[16] https://subgraph.com/vega/ (πρόσβαση 28/04/2017)

[17] http://www.arachni-scanner.com/ (πρόσβαση 28/04/2017)

[18] https://www.owasp.org/index.php/ZAP (πρόσβαση 28/04/2017)

[19] http://w3af.org/(πρόσβαση 28/04/2017)

[20] https://github.com/spinkham/skipfish (πρόσβαση 28/04/2017)

[21] https://github.com/adamdoupe/WackoPicko (πρόσβαση 28/04/2017)

[22] http://www.coresecurity.com/ (πρόσβαση 01/04/2016)

[23] http://www.pen-tests.com/limitations-of-penetration-testing.html (πρόσβαση 28/04/2017)

[24] http://www.sectoolmarket.com (πρόσβαση 28/04/2017)

[25] https://github.com/OWASP/Top10/raw/master/2017/OWASP%20Top%2010%20-%202017%20RC1-English.pdf (πρόσβαση 28/04/2017)

[26] http://www.computerweekly.com/tip/Cross-site-request-forgery-Lessons-from-a-CSRF-attack-example (πρόσβαση 28/04/2017)

[27] http://nodegoat.herokuapp.com/tutorial (πρόσβαση 28/04/2017)

[28] Sagala A, Manurung E (2015) Testing and comparing result scanning using web vulnerability scanner. Adv Sci Lett 21(11):3458–3462

[29] Selection of penetration testing methodologies: A comparison and evaluation, This paper was originally presented at The Proceedings of [the] 13th Australian Information Security Management Conference, held from the 30 November – 2 December, 2015 (pp. 65-72), Edith Cowan University Joondalup Campus, Perth, Western Australia.

[30] J. Scambray and M. Shema. Hacking Web Applications Exposed: Web Application Security Secrets and Solutions, 2002.

[31] M. Curphey. A Guide to Building Secure Web Applications and Web Services, September 2002.

[32] A. Conry-Murray. Web application security for all. *Network Magazine*, 20(2):31 – 51, 2005.

[33] https://www.owasp.org/index.php/Category:Cheatsheets (πρόσβαση 28/04/2017)

[34] https://trends.builtwith.com/framework (πρόσβαση 28/04/2017)

[35] https://www.mcafee.com/us/downloads/free-tools/hacme-bank.aspx

[36] www.nth-dimension.org.uk

[37] Shelly, D. A. (2010). Using a Web Server Test Bed to Analyze the Limitations of Web Application Vulnerability Scanners (Doctoral dissertation, Virginia Polytechnic Institute and State University).

[38] Cortes I. I. S., Muñoz F. R., Villalba L.J.G, (2017) Enlargement of vulnerable web applications for testing,. doi:10.1007/s11227-017-1981-2

[39] Fonseca, J., Vieira, M., & Madeira, H. (2007, December). Testing and comparing Web vulnerability scanning tools for SQL injection and XSS attacks. In Dependable Computing, 2007. PRDC 2007. 13th Pacific Rim International Symposium on (pp. 365-372). IEEE.

[40] McAllister, S., Kirda, E., & Kruegel, C. (2008, January). Leveraging user interactions for in-depth testing of web applications. In Recent Advances in Intrusion Detection (pp. 191-210). Springer Berlin Heidelberg.

[41] Deepa, G., Thilagam, P.S., Khan, F.A. (2017). Black-Box Detection of XQuery Injection and Parameter Tampering Vulnerabilities in Web Applications et al. International Journal of Information Security doi:10.1007/s10207-016-0359-4

[42] www.enisa.europa.eu (πρόσβαση 28/04/2017)

[43] Graves K. (2010). CEH : certified ethical hacker study guide 1st ed. Wiley Inc., Indianapolis, Indiana (σελίδες 197-198)

[44] https://www.tripwire.com/state-of-security/featured/adultfriendfinder-data-breach-what-you-need-to-know/ (πρόσβαση 28/04/2017)

[45] https://www.arbornetworks.com/insight-into-the-global-threat-landscape (πρόσβαση 28/04/2017)

[46] http://www.verizonenterprise.com/verizon-insights-lab/dbir/2016/ (πρόσβαση 28/04/2017)

[47] https://www.symantec.com/security-center/threat-report (πρόσβαση 28/04/2017)

[48] Shah, S. & Mehtre, B.M. J Comput Virol Hack Tech (2015) An overview of vulnerability assessment and penetration testing techniques. doi:10.1007/s11416-014-0231-x

[49] https://saas.hpe.com/en-us/software/webinspect (πρόσβαση 28/04/2017)

Παράρτημα Α
Ορισμοί Βασικών Όρων

Προκειμένου να αποσαφηνιστεί η ορολογία που χρησιμοποιείται, δίνονται εδώ οι ορισμοί βασικών όρων.

Εξυπηρετητής Ιστού. Ένα πρόγραμμα που διαχειρίζεται την υπηρεσία HTTP για να λαμβάνει αιτήματα πελατών για περιεχόμενο στο διαδίκτυο. Ο διακομιστής χειρίζεται τα αιτήματα του πελάτη για να βεβαιωθεί ότι είναι σε έγκυρη μορφή και ότι υπάρχουν οι απαιτούμενοι πόροι πριν να μεταβιβάσει τις αιτήσεις στην εφαρμογή Ιστού για επεξεργασία. Τα πακέτα λογισμικού διακομιστή ιστού περιλαμβάνουν τον διακομιστή Apache και τον Microsoft IIS [30].

Εφαρμογή Ιστού. Μια εφαρμογή που χειρίζεται την λογική πλευρά του διακομιστή ενός διακομιστή ιστού και είναι προσπελάσιμη μέσω του Internet χρησιμοποιώντας το πρωτόκολλο HTTP. Οι τρεις βαθμίδες συσχετίζονται συνήθως με μια εφαρμογή ιστού είναι το επίπεδο παρουσίασης, η λογική βαθμίδα και το επίπεδο δεδομένων. Το επίπεδο παρουσίασης είναι υπεύθυνο για την εμφάνιση ή τη λήψη δεδομένων από τον πελάτη, το λογικό επίπεδο χειρίζεται την επεξεργασία εισερχόμενης εισόδου από την παρουσίαση ή τη βαθμίδα δεδομένων και η βαθμίδα δεδομένων παρέχει ένα μηχανισμό για την αποθήκευση και την ανάκτηση δεδομένων που απαιτούνται από τη λογική βαθμίδα [30, 31].

Σαρωτής ευπάθειας εφαρμογής Ιστού. Ένας τύπος εργαλείου που ανιχνεύει εφαρμογές ιστού για γνωστά ελαττώματα και ευπάθειες. Συνήθως ένα σύνολο υπογραφών χρησιμοποιείται από το σαρωτή σε δοκιμή έναντι διαφόρων στοιχείων εφαρμογών ιστού [32]. Αυτός ο τύπος σαρωτή χρησιμοποιεί αυτό που ονομάζεται τεχνική σάρωσης μαύρου κουτιού επειδή δεν έχει πρόσβαση στον πηγαίο κώδικα της εφαρμογής που σαρώνει. Αυτά τα εργαλεία μπορούν επίσης να αναφέρονται ως σαρωτές εφαρμογών ιστού, σαρωτές ασφάλειας ιστού, σαρωτές ευπάθειας ιστού ή απλά σαρωτές.

Web Crawling. Επίσης αναφέρεται ως web spidering, αυτή είναι η τεχνική της αυτόματης περιήγησης σε έναν ιστότοπο με μεθοδικό τρόπο για να αποκαλύψει τη δομή του. Σε πολλές

περιπτώσεις, τα αντίγραφα των σελίδων που επισκέπτονται γίνονται έτσι ώστε οι πόροι της εφαρμογής του web, όπως οι ενσωματωμένοι σύνδεσμοι και ο κώδικας HTML, να έχουν πρόσβαση πιο γρήγορα.

Ευπάθεια. Στις εφαρμογές ιστού, αυτό είναι μια περιοχή αδυναμίας που είναι επιρρεπής σε επίθεση. Για να είναι ευπαθές ένα στοιχείο εφαρμογής ιστού, πρέπει να είναι προσβάσιμο από έναν εισβολέα ο οποίος διαθέτει την ικανότητα να εκμεταλλευτεί το σφάλμα.

Εκμετάλλευση ευπαθειών. Στις εφαρμογές ιστού, πρόκειται για ένα κομμάτι κώδικα υπολογιστή που προσβάλλει μια ευπάθεια για να προκαλέσει ακούσια ή απρόβλεπτη συμπεριφορά. Μπορεί να λάβει τη μορφή λογισμικού ή ακολουθιών εντολών και αποσκοπεί συνήθως να αποκτήσει τον έλεγχο ενός συστήματος, κλιμακώνοντας τα δικαιώματα χρήστη ή να αρνηθεί την υπηρεσία στην εφαρμογή [37].