Πανεπιστήμιο Πελοποννήσου
Σχολή Οικονομίας, Διοίκησης και Πληροφορικής
Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Μεταπτυχιακή Εργασία

Έλεγχος ασφάλειας ιστοχώρων: μεθοδολογίες και έλεγχος ευπαθειών

Ονοματεπώνυμο Φοιτητή: Μακρυπόδης Βασίλειος

ΑΜ: 2010016

Επιβλέπων Καθηγητής: Βασιλάκης Κωνσταντίνος

Τρίπολη, Οκτώβριος 2013

Στην κόρη μου Ιωάννα,

Ευχαριστίες

Κάθε εργαζόμενος που έχει συνειδητοποιήσει τις ανάγκες του χώρου όπου εργάζεται, συχνά αναζητεί επιστημονικά αντικείμενα στο χώρο δράσης του, που επαυξάνουν τις αρχικές του γνώσεις. Στο πλαίσιο αυτό, αναζήτησα κάποιες επιπρόσθετες γνώσεις που αφορούν την επιστήμη των υπολογιστών. Συμμετέχοντας στο πρόγραμμα μεταπτυχιακών σπουδών «Πληροφορική και Εφαρμογές» κατέληξα στη συγγραφή της διπλωματικής μου εργασίας με τίτλο «Έλεγχος ασφάλειας ιστοχώρων: μεθοδολογίες και έλεγχος ευπαθειών».

Θα ήθελα να ευχαριστήσω τον κο. Βασιλάκη Κωνσταντίνο για την υποστήριξη του, την ενθάρρυνσή του και την καθοδήγηση του, καθ' όλη τη διάρκεια εκπόνησης της εν λόγω διπλωματικής εργασίας.

Επίσης οφείλω ένα μεγάλο ευχαριστώ στη σύζυγό μου Κατερίνα για την υπομονή και τη κατανόηση που έδειξε, καθώς και για τις συνθήκες εκείνες που διαμόρφωσε και βοήθησε έμμεσα για την ολοκλήρωση της εργασίας μου. Στην κόρη μου Ιωάννα που μου έδινε χαρά με την παρουσία της, καθ' όλη τη διάρκεια συγγραφής της εργασίας και στην αγέννητη ακόμη κόρη μου που μου έδινε χαρά με την… «αισθητή» παρουσία της και με την αναμονή του ερχομού της.

Τέλος να ευχαριστήσω τον αδερφό μου Άγγελο, για την πολύτιμη βοήθεια που μου πρόσφερε.

Περίληψη

Στη σημερινή εποχή, η τεχνολογία μπορεί να θεωρηθεί ο καλύτερος σύμμαχος του τρόπου ζωής του σύγχρονου ανθρώπου. Μπορούμε να κάνουμε σχεδόν τα πάντα πάνω από δικτυωμένα συστήματα ή το διαδίκτυο, από απλές εργασίες όπως η κράτηση θέσης σε μια πτήση αεροπορικής εταιρείας, μέχρι σύνθετες εργασίες εμβάσματα σημαντικών ποσών μεταξύ μεγάλων χρηματοπιστωτικών οργανισμών, σχεδιασμό ταξιδιών κ.λπ. Η ασφάλεια των επικοινωνιών και των πληροφοριακών συστημάτων ωστόσο δεν μπορούμε να την θεωρήσουμε δεδομένη και όσο η τεχνολογία εξελίσσεται τόσο το πρόβλημα της ασφάλειας γίνεται πιο σύνθετο. Είναι εξαιρετικά σημαντικό να γνωρίσουμε νέες τεχνικές και μεθοδολογίες για την καταπολέμηση των κυβερνοεπιθέσεων σε κρίσιμα συστήματα πληροφοριών, των κλοπών δεδομένων και των διεισδύσεων σε δίκτυα δεδομένων. Οι οργανισμοί και οι διαχειριστές συστημάτων πρέπει να εκπαιδευτούν για να είναι σε θέση να επιλέξουν την κατάλληλη τεχνολογία ασφαλείας, τα κατάλληλα εργαλεία και εκείνες τις μεθοδολογίες που θα τους βοηθήσουν στην πρόληψη ή στον μετριασμό των απειλών, πριν αυτές εκδηλωθούν. Πρέπει επίσης να αξιολογήσουν την ασφάλεια των συστημάτων που υποστηρίζουν, για να αναδείξουν τις ευπάθειες που πιθανά να υπάρχουν και να λάβουν τα κατάλληλα διορθωτικά μέτρα.

Αντικείμενο της παρούσας μεταπτυχιακής εργασίας αποτελεί η διερεύνηση των ευπαθειών και των υπαρχουσών μεθοδολογιών που αφορούν τον έλεγχο ασφαλείας των ιστοχώρων, όπως αυτές περιγράφονται από τρεις μεγάλους οργανισμούς. Επίσης πραγματοποιήθηκαν έλεγχοι ασφαλείας, μέσω δοκιμών διείσδυσης, σε ιστόχωρους του Πανεπιστημίου Πελοποννήσου

Η δομή της εργασίας έχει ως ακολούθως: στο **πρώτο κεφάλαιο** γίνεται μια εισαγωγή στην έννοια της ασφάλειας και στις μεθόδους ελέγχου ασφάλειας των εφαρμογών ιστού. Οι δέκα δημοφιλέστερες ευπάθειες που συναντώνται σε δικτυακές εφαρμογές όπως αυτές παρουσιάζονται από τον οργανισμό OWASP παρουσιάζονται στο **δεύτερο κεφάλαιο**. Στο **τρίτο κεφάλαιο** παρουσιάζεται η μεθοδολογία για την αξιολόγηση της ασφάλειας σύμφωνα με τον οργανισμό NIST. Οι ευπάθειες και οι πιθανοί κίνδυνοι που απειλούν ένα ιστόχωρο σύμφωνα με την κατηγοριοποίηση του οργανισμού Web Application Security παρατίθενται στο **τέταρτο κεφάλαιο**. Στο **πέμπτο κεφάλαιο** γίνεται μια επισκόπηση και αναλύονται τα χαρακτηριστικά των πιο σημαντικών εργαλείων που διατίθενται για τον έλεγχο των ευπαθειών. Στο **έκτο κεφάλαιο** παρουσιάζονται και αναλύονται διεξοδικά οι λειτουργίες του ανοιχτού λογισμικού Webscarab. Πραγματοποιήθηκαν μετρήσεις ασφαλείας με τη μέθοδο της δοκιμής διείσδυσης, σύμφωνα με τον οδηγό OWASP Testing Guide v.3 στους εξυπηρέτες του Τμήματος Κοινωνικής & Εκπαιδευτικής Πολιτικής του Πανεπιστημίου Πελοποννήσου με το συγκεκριμένο λογισμικό. Στο **έβδομο κεφάλαιο** παρουσιάζονται τα συμπεράσματα από τις πραγματοποιηθείσες μετρήσεις ασφαλείας.

Λέξεις - κλειδιά: Δοκιμή διείσδυσης, έλεγχος ευπάθειας, αξιολόγηση ασφάλειας, εργαλεία μέτρησης ασφάλειας

Abstract

Nowadays technology can be considered as the indispensable companion of people's modern way of life. Numerous tasks can be performed over networks or the internet, ranging from simple tasks such as booking a flight, to highly complex tasks such as transferring significant amounts of money through financial organizations or vacation planning. The security of communications and information systems, however, cannot be taken as granted. The more technology evolves, the more complicated the security issue becomes. It is extremely important to keep up-to-date with new technics and methods, in order to confront cyber-attacks against critical data systems, data-stealing and database intrusions. Organizations and system administrators must be trained to choose the more efficient security technologies, tools and methods so that to prevent or moderate the possible threats. The security of the systems supported must be evaluated, so as to identify possible vulnerabilities and take appropriate corrective measures.

This dissertation focuses on the study of the vulnerabilities and the existing methods concerning the web sites' security assessment, as they are described by three important organizations such as **OWASP**, **NIST** and **Web Application Security**. Security assessments were also conducted for the websites of the University of Peloponnese.

The structure of this dissertation is as follows: **first chapter** is an introduction to the concept of security and the methods for assessing the web applications' security. The **second chapter** investigates the ten most popular vulnerabilities that come across in the web applications as they are presented by the OWASP. **Chapter three** presents the security assessment method published by NIST. The **fourth chapter** examines vulnerabilities and possible threats for a website, according to the classification of the Web Application Security organization. **Chapter five** surveys and analyzes the most important tools available for vulnerability control. **Chapter six** presents and analyses the functionality provided by Webscarab, a piece of software for conducting penetration tests. Webscarab was used for conducting penetration testing, according to the OWASP Testing Guide v.3, against the servers of the Department of Social and Educational Policy of the University of Peloponnese. Finally, the **chapter seven** concludes the thesis and presents the outcomes of the conducted security assessment tests.

Key - words: Penetration Testing, vulnerability control, security Assessment, security measurement tools

Περιεχόμενα

Ευρετήριο Εικόνων

Ευρετήριο Πινάκων

1. Θεωρητικό υπόβαθρο

1.1. Ασφάλεια υπολογιστικών συστημάτων

Η ασφάλεια πληροφοριακών συστημάτων είναι κλάδος της επιστήμης της πληροφορικής που ασχολείται με την προστασία των υπολογιστών, των δικτύων που τους συνδέουν και των δεδομένων που εμπεριέχονται σε αυτά τα συστήματα. Η ασφάλεια μεριμνά για την προστασία των πόρων (δεδομένων και προγραμμάτων) από συμπτωματική ή κακόβουλη τροποποίηση, καταστροφή ή διαρροή, διασφαλίζοντας την ακεραιότητα, την αυθεντικότητα, την εγκυρότητα, την εμπιστευτικότητα, τη διαθεσιμότητα και τη μη αποποίηση ευθύνης. Ειδικότερα, οι διαστάσεις αυτές της ασφάλειας ορίζονται ως ακολούθως:

- *Εμπιστευτικότητα*: οι πληροφορίες είναι προσπελάσιμες μόνο από εξουσιοδοτημένους χρήστες
- *Ακεραιότητα*: τα δεδομένα και τα προγράμματα τροποποιούνται και καταστρέφονται μόνο με καλά καθορισμένους τρόπους και με κατάλληλη εξουσιοδότηση
- *Διαθεσιμότητα*: Οι εξουσιοδοτημένοι χρήστες θα μπορούν να χρησιμοποιήσουν δεδομένα, προγράμματα και υπηρεσίες όταν το επιθυμήσουν
- *Αυθεντικότητα*: εξασφάλιση ότι τα δεδομένα είναι απαλλαγμένα ατελειών και ανακριβειών κατά τις εξουσιοδοτημένες τροποποιήσεις
- *Εγκυρότητα*: εξασφάλιση ότι τα δεδομένα είναι ακριβή και πλήρη
- *Μη αποποίηση ευθύνης*: κανένα από τα συναλλασσόμενα μέρη δεν πρέπει να έχει τη δυνατότητα να αρνηθεί τη συμμετοχή του σε μία συναλλαγή.

Ανάμεσα στους συγγενικούς τομείς της ασφάλειας πληροφοριακών συστημάτων συμπεριλαμβάνονται η ψηφιακή εγκληματολογία και η εφαρμοσμένη κρυπτογραφία.

1.2. Σχεδιασμός Πολιτικής Ασφαλείας

Η πολιτική ασφάλειας των πληροφοριακών συστημάτων περιλαμβάνει τον σκοπό και τους στόχους της ασφάλειας, οδηγίες, διαδικασίες, κανόνες, ρόλους και υπευθυνότητες που αφορούν την προστασία των πληροφοριακών συστημάτων του οργανισμού. Ο σχεδιασμός πολιτικών ασφάλειας στα Πληροφοριακά Συστήματα, συνδέεται άμεσα τόσο με τεχνικές, διαδικασίες και διοικητικά μέτρα όσο και με ηθικό-κοινωνικές αντιλήψεις, αρχές και παραδοχές, προφυλάσσοντας από κάθε είδους απειλή τυχαία ή σκόπιμη. Οι διαδικασίες σχεδιασμού πολιτικών ασφαλείας, δεν θα πρέπει να παρεμβαίνουν στην απρόσκοπτη λειτουργία των πληροφοριακών συστημάτων, ενώ οφείλουν να τηρούν την αρχή της αποκέντρωσης, της ύπαρξης αντικατάστασης και την αρχή της άμυνας σε βάθος. Ως βάση μπορεί να οριστεί ο εντοπισμός, η αξιολόγηση και στη συνεχεία η διαμόρφωση ενός θεωρητικού πλαισίου για το σχεδιασμό πολιτικών σχεδιασμού ασφάλειας.

Το πιο βασικό σημείο στη διαδικασία σχεδιασμού πολιτικών ασφάλειας, είναι ο εντοπισμός και χαρακτηρισμός με την κατάλληλη διαβάθμιση των πληροφοριών που πρόκειται να χρησιμοποιηθούν και να προστατευθούν (εμπιστευτικές,

απόρρητές, άκρως απόρρητες κ.λπ.). Εκτός από τις αρχές της **Ακεραιότητας Πληροφοριών**, την **Εμπιστευτικότητα** και τη **Διαθεσιμότητα Πληροφοριών** οι πολιτικές ασφάλειας θα πρέπει να εμπεριέχουν και τους όρους αυθεντικότητα, εγκυρότητα, μοναδικότητα και μη αποποίηση.

Ωστόσο, οι πολιτικές ασφάλειας προϋποθέτουν την ύπαρξη μίας δέσμης βασικών αρχών, εκφρασμένων με σαφήνεια η οποία να τέμνει τους σχεδιαστικούς στόχους των λειτουργικών συστημάτων. Κάθε αντικείμενο του συστήματος θα πρέπει να μπορεί να αναγνωρισθεί μονοσήμαντα και να συνοδεύεται από μία ένδειξη του βαθμού εμπιστευτικότητας. Επιπλέον, η ισχύς των ασφαλιστικών μηχανισμών δεν θα πρέπει να βασίζονται στην άγνοια των χρηστών σχετικά με τις τεχνικές ασφαλείας οι οποίες χρησιμοποιούνται *(Security through obscurity)*, αλλά στην αποτελεσματική τους σχεδίαση.[1]

Στόχος μιας μεθοδολογίας ανάπτυξης πολιτικών ασφάλειας είναι ο περιορισμός της επικινδυνότητας σε αποδεκτό επίπεδο. Η μεθοδολογία περιλαμβάνει αξιολόγηση της επικινδυνότητας και καθορισμό του αποδεκτού επιπέδου ασφαλείας, ανάπτυξη και εφαρμογή μιας πολιτικής ασφαλείας καθώς και δημιουργία κατάλληλου οργανωτικού πλαισίου και εξασφάλιση των απαιτούμενων πόρων για την εφαρμογή της πολιτικής ασφάλειας. Η πολιτική ασφάλειας, μαζί με το σύνολο των μέτρων προστασίας, αποτελούν το σχέδιο ασφαλείας (security plan) για τα πληροφοριακά συστήματα ενός οργανισμού διότι χρειαζόμαστε ένα ολοκληρωμένο πλαίσιο με την καθοδήγηση των μέτρων ασφαλείας να λειτουργεί ως μέσο επικοινωνίας των εμπλεκομένων στα ζητήματα ασφαλείας.

Επιπλέον θεμελιώνεται η σημασία της ασφάλειας του πληροφοριακού συστήματος για τα μέλη του οργανισμού, δημιουργείται μια κουλτούρα ασφαλείας καθώς πολλές φορές αποτελεί νομική υποχρέωση και αποτελεί παράγοντα εμπιστοσύνης μεταξύ οργανισμού και πελατών. Τα είδη των πολιτικών ασφαλείας είναι:

- οι **τεχνικές** (computer oriented), οι οποίες περιλαμβάνουν Πολιτικές Ασφάλειας Πληροφοριών, Πολιτικές Ασφάλειας Λειτουργικών Συστημάτων και Πολιτικές Ασφάλειας Δικτύων Υπολογιστών και
- οι **οργανωτικές** (human oriented), όπου διαμορφώνονται *Πολιτικές Ασφάλειας Πληροφοριακών Συστημάτων,* οι οποίες άπτονται όλων των πτυχών των πληροφοριακών συστημάτων.

Από την άλλη πλευρά, οι πολιτικές ασφάλειας μπορούν να έχουν τις εξής μορφές:

- **ατομικές** (individual security policies). Οι πολιτικές αυτές αναπτύσσονται ανά σύστημα ή εφαρμογή (π.χ. Πολιτική Ασφάλειας για τη χρήση του e-mail). Παρέχουν αποσπασματική διαχείριση της ασφάλειας ΠΣ και έχουν μεγάλη πολυπλοκότητα στη συντήρηση των πολιτικών, είναι ωστόσο αποτελεσματικές όταν υπάρχουν αυτόνομες εφαρμογές και υπολογιστικά συστήματα που δε συνδέονται μεταξύ τους.
- **αναλυτικές** (Comprehensive Security Policies), όπου υπάρχει ένα ενιαίο έγγραφο που αναφέρεται σε όλα τα υπολογιστικά συστήματα, τις εφαρμογές και τις διαδικασίες του. Είναι μεγάλες σε όγκο, και κατά συνέπεια όχι πολύ εύχρηστες, ενώ οι οδηγίες και διαδικασίες που περιλαμβάνονται είναι σε γενικό επίπεδο, χωρίς λεπτομέρειες.

- **αρθρωτές** (Modular Security Policies). Σε αυτές υπάρχει ένα ενιαίο έγγραφο με παραρτήματα που περιγράφουν τις επιμέρους πολιτικές. Το έγγραφο αυτό μπορεί να είναι σε μορφή υπερκειμένου (hypertext).

Οι απαιτήσεις για την ασφάλεια του ΠΣ που πρέπει να ικανοποιεί η Πολιτική Ασφάλειας προέρχονται από όλους τους εμπλεκόμενους στη χρήση και λειτουργία του ΠΣ ενός οργανισμού, όπως είναι:

- Οι χρήστες και διαχειριστές των ΠΣ
- Η διοίκηση του οργανισμού
- Οι πελάτες του οργανισμού
- Οι νομικές και κανονιστικές διατάξεις που διέπουν τη λειτουργία του

Ο καθορισμός της πολιτικής ασφάλειας ενός πληροφοριακού συστήματος θα πρέπει να καλύπτει τις ακόλουθες κατηγορίες:

- Ζητήματα προσωπικού
- Φυσική ασφάλεια
- Έλεγχος πρόσβασης στο πληροφοριακό σύστημα
- Διαχείριση υλικών και λογισμικών
- Νομικές υποχρεώσεις
- Διαχείριση της πολιτικής ασφάλειας
- Οργανωτική δομή
- Σχέδιο συνέχισης λειτουργίας

Όταν εφαρμόζουμε μια πολιτική ασφαλείας επιδιώκουμε:

- οι οδηγίες και τα μέτρα προστασίας οφείλουν να καλύπτουν το σύνολο των αγαθών και όλες τις λειτουργίες (πληρότητα)
- να λάβουμε υπόψη τις τρέχουσες τεχνολογικές εξελίξεις (επικαιρότητα)
- με κάποιες τροποποιήσεις ή προσθήκες να μπορεί η πολιτική να καλύπτει μικρές αλλαγές ή επεκτάσεις στο πληροφοριακό σύστημα (γενικευσιμότητα). Επιπλέον πρέπει να υπάρχει σαφήνεια και εύκολη κατανόηση, τεχνολογική ανεξαρτησία και καταλληλότητα ανάλογα με τον οργανισμό που απευθύνεται.
- Μια Πολιτική Ασφάλειας ΠΣ επιτυγχάνει καλύτερα τους στόχους της όταν υποστηρίζει τους επιχειρηματικούς στόχους, η ανώτερη διοίκηση του οργανισμού υποστηρίζει και συμμετέχει ενεργά στην εφαρμογή της, είναι κατάλληλη για το συγκεκριμένο περιβάλλον όπου εφαρμόζεται (οργανωσιακή κουλτούρα) και οι χρήστες εκπαιδεύονται και ενημερώνονται κατάλληλα. Παράλληλα, πρέπει να υπάρχουν διαδικασίες αξιολόγησης της αποτελεσματικότητάς της, ώστε να αναθεωρείται αναλόγως, να εφαρμόζεται σταδιακά, ανάλογα με το βαθμό της αλλαγής που επιφέρει η εφαρμογή της Πολιτικής στις δραστηριότητες των χρηστών και τέλος να έχουν εύκολη και άμεση πρόσβαση σε αυτήν όλοι οι χρήστες του ΠΣ.

1.3. Βασικές αρχές ασφάλειας της πληροφορίας

Για να κατανοήσουμε καλύτερα την έννοια της ασφάλειας της πληροφορίας (information security) πρέπει να μπορέσουμε να διακρίνουμε τις τρεις συνεχείς και διαφορετικές μεταξύ τους δράσεις που αυτή απαιτεί:

- **Πρόληψη** (prevention): Είναι η λήψη μέτρων που μας επιτρέπουν να προλαβαίνουμε τη δημιουργία επικίνδυνων καταστάσεων.
- **Ανίχνευση** (detection): Είναι η λήψη μέτρων που μας επιτρέπουν να αντιληφθούμε πως, πότε και από ποιόν έχει προκληθεί κάποια ζημιά.
- **Αντίδραση** (reaction): Είναι η λήψη μέτρων που μας επιτρέπουν να αποκαταστήσουμε τις ζημιές που έχουν προκληθεί.

Έτσι συμπεραίνουμε πως, η πρόληψη αποτελεί την ουσία της ασφάλειας, καθώς χρησιμεύει ως μονάδα ποσοτικής μέτρησης έναντι της ανίχνευσης και αντίδρασης. Η ανίχνευση συνεπικουρεί στον εντοπισμό τυχόν κενών και προβλημάτων ασφάλειας μόλις τα προληπτικά μέτρα τεθούν σε εφαρμογή και η αντίδραση ανταποκρίνεται με τους κατάλληλους μηχανισμούς στις παραβιάσεις ασφάλειας.

Η ασφάλεια σε ένα σύστημα, κατά κύριο λόγο, συνεπάγεται την εξέταση των ευπαθειών, των απειλών, των αντιμέτρων και του αποδεκτού ρίσκου.

- **Ευπάθεια (vulnerability):** ονομάζεται μια αδυναμία της εφαρμογής, η οποία μπορεί να οφείλεται σε ένα τρωτό σημείο της κατά τη σχεδίαση της, ή σε ένα σφάλμα κατά την υλοποίηση της και η οποία μπορεί να επιτρέψει στην πραγματοποίηση μιας απειλής.
- **Απειλή (threat):** ονομάζουμε οποιαδήποτε πράξη ή γεγονός που θα μπορούσε να παραβιάσει την ασφάλεια μιας εφαρμογής ή ενός συστήματος γενικότερα και να προκαλέσει ζημιά υπό μορφή καταστροφής, κοινοποίησης, τροποποίησης των στοιχείων ή και άρνηση της υπηρεσίας.
- **Αντίμετρα (countermeasures):** ονομάζουμε τον μηχανισμό ή τη διαδικασία εκείνη (τεχνολογίες ή μοντέλα άμυνας) που αποσκοπεί στη μείωση ή αποτροπή των επιμέρους κινδύνων στους οποίους εκτίθεται το πληροφοριακό σύστημα. Τα αναγκαία αντίμετρα σε μια εφαρμογή πρέπει να αναγνωριστούν με την χρήση της ανάλυσης κινδύνου (risk analysis) έτσι ώστε να διασφαλιστεί ότι η εφαρμογή προστατεύεται από κοινούς τύπους επιθέσεων. Οποιαδήποτε αδυναμία ή ρωγμή στη σχεδίαση των αντίμετρων ή ακόμα και ή παράλειψη ενός συγκεκριμένου αντιμέτρου μπορεί να έχει ως αποτέλεσμα μια ευπάθεια, η οποία μπορεί να είναι ικανή να καταστήσει την εφαρμογή ευάλωτη σε επιθέσεις.

Η ασφάλεια πληροφοριακών συστημάτων στηρίζεται σε τρεις βασικές αρχές. Την **Ακεραιότητα**, την **Διαθεσιμότητα** και την **εμπιστευτικότητα**.

Εικόνα 1. Οι τρεις βασικές αρχές της ασφάλειας

Ακεραιότητα (integrity): αναφέρεται στη διατήρηση των δεδομένων ενός πληροφοριακού συστήματος σε μια γνωστή κατάσταση χωρίς ανεπιθύμητες τροποποιήσεις, αφαιρέσεις ή προσθήκες από μη εξουσιοδοτημένα άτομα, καθώς και την αποτροπή της πρόσβασης ή χρήσης των υπολογιστών και δικτύων του συστήματος από άτομα χωρίς άδεια. Στην περίπτωση αυτή αναφερόμαστε είτε στην πληροφορία η οποία υπάρχει μέσα στα δεδομένα είτε στην μεταπληροφορία η οποία συνοδεύεται από τα δεδομένα. Ένα παράδειγμα μεταπληροφορίας σε μία βάση δεδομένων θα ήταν τα ονόματα των πινάκων, το σχήμα τους και το πλήθος των εγγραφών τους. Ο επιτιθέμενος μπορεί να μην καταφέρει να διαβάσει τις εγγραφές της βάσης δεδομένων αλλά ακόμη και η γνώση του σχήματος (π.χ. αν υπάρχει στήλη «πολιτικές πεποιθήσεις») ή η γνώση του πλήθους των εγγραφών (π.χ. αριθμός πελατών μιας εταιρείας) μπορεί να είναι πολύτιμες πληροφορίες για τον επιτιθέμενο, ή –αντίστροφα- πληροφορίες που δεν πρέπει να διαρρεύσουν, σε ό,τι αφορά τον ιδιοκτήτη του πληροφοριακού συστήματος.

Διαθεσιμότητα (availability): Η διαθεσιμότητα των δεδομένων και των υπολογιστικών πόρων, είναι η εξασφάλιση ότι οι υπολογιστές, τα δίκτυα και τα δεδομένα θα είναι στη διάθεση των νομότυπων χρηστών όποτε απαιτείται η χρήση τους.

Μία τυπική απειλή που αντιμετωπίζουν τα σύγχρονα πληροφοριακά συστήματα είναι η επίθεση άρνησης υπηρεσιών (DOS attack), που έχει ως σκοπό να τεθούν εκτός λειτουργίας οι στοχευόμενοι πόροι είτε προσωρινά, είτε μόνιμα. Η άρνηση υπηρεσιών δεν προκαλείται αναγκαία από εχθρική επίθεση. Το φαινόμενο Slashdot, κατά το οποίο δημοσιεύεται σε δημοφιλή ιστοχώρο ένας σύνδεσμος προς μια ιστοσελίδα φιλοξενούμενη σε εξυπηρέτη με σύνδεση χαμηλής χωρητικότητας, με συνέπεια εκατοντάδες χιλιάδες αναγνώστες να υπερφορτώσουν τη σύνδεση της αναφερομένης ιστοσελίδας (του εξυπηρέτη δηλαδή με τη χαμηλής χωρητικότητας σύνδεση), προκαλεί το ίδιο αποτέλεσμα.

Εμπιστευτικότητα (confidentiality): Είναι η προστασία της πληροφορίας και των πληροφοριακών συστημάτων. Αυτό σημαίνει ότι ευαίσθητες πληροφορίες δεν

θα έπρεπε να αποκαλύπτονται σε μη εξουσιοδοτημένα άτομα. Τα δεδομένα τα οποία είτε διακινούνται είτε αποθηκεύονται πρέπει να είναι προσβάσιμα μόνο από τα άτομα που πρέπει να έχουν πρόσβαση σε αυτά. Δεν θέλουμε κανένας τρίτος με κανένα τρόπο να μπορεί να διαβάσει αυτά τα δεδομένα και να τα χρησιμοποιήσει κατά το δοκούν.

Η διαρροή ευαίσθητων πληροφοριών μπορεί να γίνει με πιο παραδοσιακές μεθόδους από την ψηφιακή υποκλοπή, π.χ. με την κλοπή φορητών υπολογιστών από το κατάλληλο τμήμα μιας εταιρίας. Το 2006 μια μελέτη με τη συνεργασία 480 εταιριών έδειχνε ότι 80% των εταιριών είχε πρόβλημα με διαρροή πληροφοριών λόγω κλοπής φορητού υπολογιστή.

1.4. Βασικές μέθοδοι ελέγχου ασφάλειας των εφαρμογών ιστού.

Λόγω της μεγάλης εξάπλωσης που έχει γνωρίσει ο παγκόσμιος ιστός και οι εφαρμογές του, η ασφάλεια των εφαρμογών ιστού έχει αποκτήσει ιδιαίτερη σημασία. Ο συνεχής έλεγχος και η μέτρηση της ασφάλειας των εφαρμογών ιστού είναι απαραίτητη προϋπόθεση για τη διατήρηση της ασφάλειας ενός συστήματος, το οποίο συνδέεται στο Web. Όμως σε έναν ιστότοπο είναι δυνατό να φιλοξενείται ένα μεγάλο πλήθος εφαρμογών ιστού διαφορετικού τύπου με διαφορετικό λογισμικό. Οι εφαρμογές ιστού κατασκευάζονται σε επίπεδα, από προγράμματα και δεδομένα τα οποία φιλοξενούνται σε πολλαπλούς εξυπηρέτες (web servers, application servers, database servers) και για τον λόγο αυτό υπάρχουν διάφορες μέθοδοι ελέγχου της ασφάλειας των εφαρμογών ιστού. Οι σημαντικότερες και ευρέως χρησιμοποιούμενες μέθοδοι είναι οι ακόλουθες:

- **Επιθεώρηση ασφαλείας (security audit):** Είναι η διαδικασία όπου ένα σύστημα ελέγχεται με βάση ένα σύνολο από λίστες ελέγχου (checklists), οι οποίες διαμορφώνονται με βάση διεθνή πρότυπα σχετικά με την ασφάλεια, καθώς και κατάλληλες πολιτικές ασφάλειας του οργανισμού, που χρησιμοποιεί την εφαρμογή ιστού. Οι ελεγκτές εκτελούν την εργασία τους μέσα από προσωπικές συνεντεύξεις, ανιχνεύσεις αδυναμιών, επιθεωρήσεις των ρυθμίσεων, αναλύσεις των διαμοιρασμένων πόρων δικτύου και μελέτες των αρχείων καταγραφής (log files).
- **Αυτοαξιολόγηση ασφάλειας (security self-assement):** Εδώ δεν υπάρχουν συγκεκριμένα standards ως προς τα οποία θα αξιολογηθεί το σύστημα, αλλά ο στόχος προσδιορίζεται από την περιοχή που χρειάζεται διερεύνηση και βελτίωση στη θωράκισή της. Ξεπερνά τους πίνακες ελέγχου (checklists) και επεκτείνεται σε ένα πιο λεπτομερή έλεγχο για εντοπισμό αδυναμιών, αλλά και σε συστάσεις για επιδιορθώσεις και βελτιώσεις. Το πλεονέκτημα της μεθόδου είναι η δυνατότητα να οριστούν επίπεδα προτεραιότητας σε κάθε συστατικό που αξιολογείται, έτσι ώστε με την ολοκλήρωσή της να δοθεί μια κατάταξη προτεραιοτήτων στην επιδιόρθωση των ευπαθειών που ανιχνεύθηκαν.
- **Δοκιμή διείσδυσης (penetration testing ή ethical hacking):** Είναι η ελεγχόμενη προσομοίωση μιας επίθεσης, προκειμένου να επιτευχθεί ένας προκαθορισμένος στόχος. Επίσης είναι γνωστή και ως εσωτερική επιθεώρηση ασφάλειας (internal security auditing). Ο σκοπός της είναι να εντοπιστούν συγκεκριμένες πληροφορίες σχετικές με την ύπαρξη γνωστών

ευπαθειών και να διερευνηθεί κατά πόσο είναι δυνατόν ένας εξωτερικός χρήστης, κάνοντας χρήση αυτών των πληροφοριών, να είναι σε θέση να δημιουργήσει προβλήματα στην εφαρμογή ιστού. Δεν έχει σκοπό να εντοπίσει όλες τις ευπάθειες, αλλά να αποδείξει ότι η ασφάλεια του συστήματος μπορεί να διακυβευτεί. Η δοκιμή μπορεί να πραγματοποιηθεί στη βάση μηδενικής γνώσης (zero knowledge) ή με πλήρη γνώση (full knowledge) του συστήματος, που δοκιμάζεται. Χρησιμοποιείται για να καθορίσει την αξιοπιστία και την ισχύ των μέτρων ασφάλειας που έχουν ληφθεί. Οι "ethical hackers" προσπαθούν να υιοθετήσουν τις τεχνικές επιθέσεων των hackers, ώστε να μπορέσουν να μετρήσουν το επίπεδο ασφάλειας της εφαρμογής.

1.4.1. Σύγκριση των μεθόδων ελέγχου της ασφάλειας

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

- Για τον έλεγχο της ασφάλειας μιας εφαρμογής ιστού με τις μεθόδους της επιθεώρησης ασφάλειας και της αυτοαξιολόγησης απαιτείται η φυσική παρουσία μιας μεγάλης ομάδας ειδικών ασφαλείας στον τόπο που λειτουργεί ο οργανισμός, του οποίου η ασφάλεια της εφαρμογής ελέγχεται.
- Η ομάδα αυτή πρέπει να έχει στη διάθεσή της τα κατάλληλες λίστες ελέγχου, να έχει υψηλή τεχνογνωσία και να είναι άρτια συντονισμένη.
- Για τη διενέργεια των ελέγχων απαιτείται πολύς χρόνος, ώστε να ολοκληρωθούν οι συνεντεύξεις, οι επιθεωρήσεις, οι αξιολογήσεις και οι έρευνες στη διάρκεια των οποίων αποκαλύπτεται και διαταράσσεται η λειτουργία του οργανισμού.

Όλα τα ανωτέρω, σε συνάρτηση με την ανάγκη για συνεχείς και επαναλαμβανόμενους ελέγχους καθιστούν τη δοκιμή διείσδυσης ιδιαίτερα ελκυστική. Επιπλέον, η δοκιμή διείσδυσης έχει τα ακόλουθα πλεονεκτήματα:

- απαιτείται ελάχιστο προσωπικό, του οποίου δεν είναι αναγκαία η μετακίνηση
- παρέχει τη δυνατότητα πλήρους αυτοματοποίησης
- διαρκεί ελάχιστο χρόνο και είναι εύκολα επαναλαμβανόμενη
- δεν απαιτεί τη σε βάθος γνώση της ελεγχόμενης εφαρμογής
- δεν διαταράσσει τη λειτουργία της εφαρμογής ή του οργανισμού
- είναι πολύ οικονομικότερη από τις δύο άλλες μεθόδους.

1.4.2. Τεχνικές δοκιμής διείσδυσης

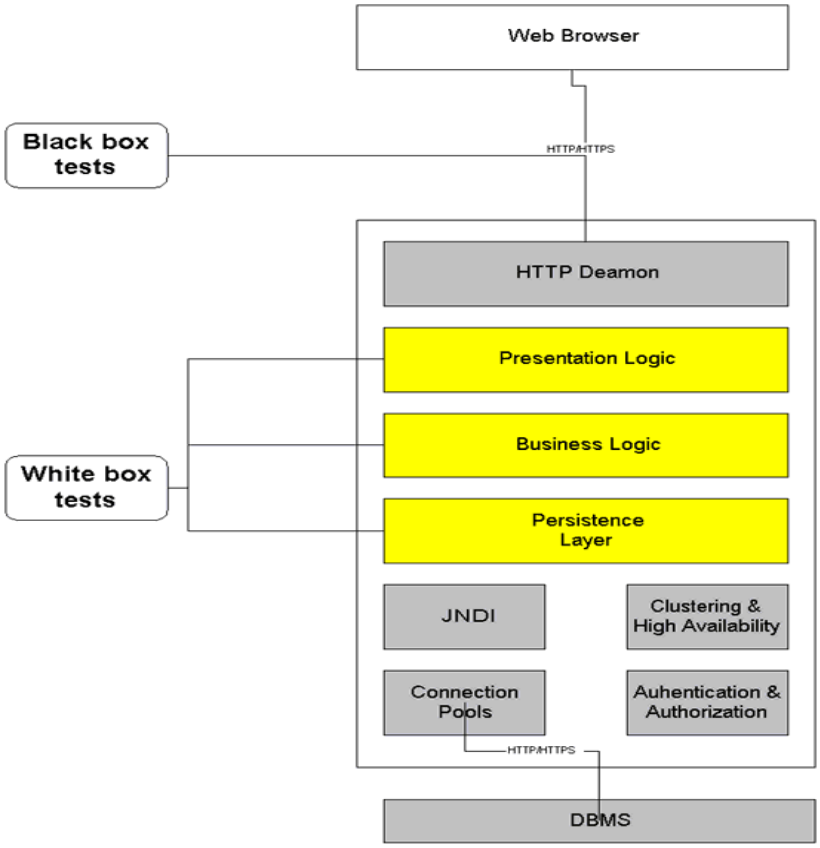
Υπάρχουν δύο κύριες κατηγορίες τεχνικών ελέγχου ασφάλειας των εφαρμογών ιστού με τη μέθοδο της δοκιμής διείσδυσης:

- Χειροκίνητη (manual), στην οποία όλη η διαδικασία ελέγχου γίνεται βήμα - βήμα χωρίς την ύπαρξη αυτοματισμών επανάληψης παρόμοιων βημάτων.
- Αυτοματοποιημένη (automated), στην οποία με τη χρήση εργαλείων, αυτοματοποιούνται μερικοί ή όλοι οι έλεγχοι και οι διαδικασίες ελέγχου.

Μία διαφορετική κατάταξη των τεχνικών δοκιμής διείσδυσης εξετάζει τον βαθμό γνώσης που ο «επιτιθέμενος» διαθέτει για το σύστημα-στόχο:

- **Black Box:** ονομάζεται ο τύπος της δοκιμής διείσδυσης όπου ο επιτιθέμενος δεν διαθέτει καμία γνώση για το υπό εξέταση σύστημα, πέραν της δημόσιας διεπαφής του. Η εφαρμογή εξετάζεται χρησιμοποιώντας την εξωτερική της διεπαφή, αυτή, που χρησιμοποιούν οι απλοί χρήστες.
- **White Box:** ονομάζεται ο τύπος της δοκιμής διείσδυσης όπου ο επιτιθέμενος διαθέτει πλήρη γνώση για το υπό εξέταση σύστημα, δηλαδή όχι μόνο γνώση της εξωτερικής διεπαφής του αλλά και της εσωτερικής δομής του, συμπεριλαμβάνοντας την αρχιτεκτονική του λογισμικού, τις δομές δεδομένων, τις βιβλιοθήκες τρίτων κατασκευαστών, το λειτουργικό σύστημα και τις βάσεις δεδομένων.

Στην (Εικόνα 2) παρουσιάζεται η διαφορά στη διεπαφή που χρησιμοποιεί ο ελεγκτής ασφαλείας για την πραγματοποίηση των ελέγχων του με τις δυο αυτές τεχνικές.



Εικόνα 2: Σχηματική αναπαράσταση της διεπαφής (interface) των Black Box και White Box τεχνικών δοκιμής διείσδυσης σε μια εφαρμογή ιστού.

(Πηγή: HTTP://www.onjava.com/pub/a/onjava/2003/05/07/blackboxwebtest.html**)**

Στην συνέχεια της εργασίας γίνεται μια επισκόπηση των μεθοδολογιών για τον έλεγχο ασφάλειας ιστοχώρων όπως αυτές περιγράφονται από τρεις μεγάλους οργανισμούς:

- OWASP (www.owasp.org)
- NIST (www.nist.gov)

- Web Application Security Consortium (www.webappsec.org)

Επίσης γίνεται μια επισκόπηση των διαθέσιμων εργαλείων για τον έλεγχο των κυριότερων ευπαθειών.

Τέλος, για το πεδίο εφαρμογής της εργασίας αυτής, επιλέχθηκαν μερικές από τις κυριότερες ευπάθειες, προκειμένου να ελεγχθεί το δίκτυο και οι εξυπηρέτες του Τμήματος Κοινωνικής & Εκπαιδευτικής Πολιτικής του Πανεπιστημίου Πελοποννήσου από πιθανά τρωτά σημεία και συντάχθηκε η σχετική αναφορά ευπαθειών των εξυπηρετών του συγκεκριμένου Τμήματος.

2. Η προσέγγιση ελέγχου του OWASP

2.1. Τι είναι το OWASP

Το Open Web Application Security Project (OWASP) είναι μια ανοιχτή κοινότητα αφιερωμένη στην ενημέρωση των οργανισμών για το πώς μπορούν να αναπτύξουν, προμηθευτούν και συντηρήσουν ασφαλείς εφαρμογές. Ως μη κερδοσκοπικός οργανισμός ακολουθεί την ιδεολογία του ελεύθερου – ανοιχτού λογισμικού και παρέχει δωρεάν, όλα τα πρότυπα, έγγραφα και εργαλεία που αναπτύσσονται. Παράλληλα διοργανώνει διεθνή συνέδρια για την ενημέρωση των επαγγελματιών του χώρου. Έχει έδρα την Αμερική και συνεργάζεται με προγραμματιστές από όλο τον κόσμο.

Το OWASP είναι ένα νέο είδος οργανισμού το οποίο είναι ελεύθερο από τις πιέσεις της αγοράς γιατί δεν σχετίζεται με καμία εταιρεία τεχνολογίας, γεγονός που επιτρέπει να παρέχει ανεπηρέαστες, πρακτικές και οικονομικές πληροφορίες για την ασφάλεια των εφαρμογών. Όπως και άλλα έργα ελεύθερου και ανοιχτού λογισμικού, έτσι και το OWASP παράγει υλικό με έναν ανοικτό και συνεργατικό τρόπο.

Η Ελληνική ομάδα εργασίας δημιουργήθηκε το 2005 με κύριο στόχο την ενημέρωση και την ευαισθητοποίηση της Ελληνικής κοινότητας αναφορικά με τους κινδύνους ασφαλείας στις διαδικτυακές εφαρμογές. Η Ελληνική ομάδα δραστηριοποιείται σε προγράμματα ελεύθερου/ανοιχτού λογισμικού, σε μεταφράσεις κειμένων του OWASP στα Ελληνικά, ενώ συμμετέχει σε συνέδρια και ημερίδες προωθώντας το OWASP σε τοπικό επίπεδο. Παράλληλα, με εργαλείο τη λίστα διανομής ηλεκτρονικού ταχυδρομείου (mailing list) που διαθέτει ενημερώνει και προκαλεί συζητήσεις σχετικά με επίκαιρα θέματα ασφαλείας στο διαδίκτυο, εκδίδει newsletter, ενώ διατηρεί και το πρώτο Ελληνικό ιστολόγιο (blog) για την ασφάλεια των πληροφοριών στη διεύθυνση: HTTP://blog.owasp.gr/

2.2. OWASP TOP 10

Οι ευπάθειες που μπορεί να παρουσιάζει ένας ιστοχώρος ή μια εφαρμογή στο διαδίκτυο είναι πολυάριθμες. Το OWASP Top 10 είναι μια λίστα που μας περιγράφει τα κυριότερα τρωτά σημεία που πρέπει να γνωρίζουν και να λαμβάνουν υπόψη τους όσοι αναπτύσσουν λογισμικό τέτοιου είδους.

Το OWASP Top 10 εκδόθηκε πρώτη φορά το 2003, ενώ μικρές ενημερώσεις έγιναν το 2004 και 2007. Η τελευταία έκδοση του 2010 σηματοδοτεί τον όγδοο χρόνο του έργου αυτού αλλά και της προσπάθειας για την ευαισθητοποίηση σχετικά με τη σημασία των κυριότερων κινδύνων στην ασφάλεια των εφαρμογών. Για καθέναν από τους κινδύνους που αναφέρονται στην λίστα γίνεται αναφορά στη δυσκολία πραγματοποίησης μιας επίθεσης, το πόσο είναι διαδεδομένη η συγκεκριμένη ευπάθεια, το πόσο εύκολα μπορεί να αντιμετωπιστεί, καθώς και τον αντίκτυπο των επιπτώσεων που έχει στην εταιρεία ή οργανισμό.

Στη συνέχεια παρουσιάζονται συνοπτικά, οι 10 κυριότερες, σύμφωνα με το OWASP, ευπάθειες που απαντώνται στις διαδικτυακές εφαρμογές. Για το 2010 και με φθίνουσα σειρά σημασίας αυτές είναι:

2.2.1. A1 – Επιθέσεις τύπου έγχυσης (Injection)

Επιτιθέμενοι: Στην πιο διαδεδομένη ευπάθεια της λίστας, μπορούμε να θεωρήσουμε ως επιτιθέμενο οποιονδήποτε χρήστη συμπεριλαμβανομένων των εξωτερικών, των εσωτερικών χρηστών καθώς και των διαχειριστών, που μπορεί να στείλει, μη αξιόπιστα δεδομένα στον διερμηνευτή (interpreter) μας προκειμένου να εκτελεστούν ως κώδικας.

Είδος επίθεσης: Ο επιτιθέμενος στέλνει απλές επιθέσεις βασισμένες σε κείμενο, εκμεταλλευόμενος τη σύνταξη της γλώσσας που υποστηρίζει ο διερμηνευτής (interpreter) στο σύστημα-στόχο. Σχεδόν οποιαδήποτε πηγή δεδομένων μπορεί να γίνει φορέας για την επίθεση τύπου έγχυσης (injection), συμπεριλαμβανομένων και των εσωτερικών πηγών.

Αδυναμία ασφαλείας: Ελαττώματα τύπου έγχυσης (injection) μπορούν να συμβούν όταν μια εφαρμογή στέλνει μη αξιόπιστα δεδομένα στον διερμηνευτή (interpreter). Αυτή η ευπάθεια είναι πολύ διαδεδομένη ιδιαίτερα στον κώδικα legacy[1], που συχνά βρίσκονται σε επερωτήσεις SQL, LDAP, Xpath, εντολές λειτουργικών συστημάτων παραμέτρους προγραμμάτων κ.ά. Η ευπάθεια αυτή είναι εύκολο να εντοπιστεί κατά την εξέταση του κώδικα αλλά δυσκολότερο να εντοπιστεί μέσω του ελέγχου διείσδυσης (penetration testing). Συνήθως εφαρμογές της κατηγορίας “scanners” και ”fuzzers”[2] μπορούν να βοηθήσουν τους επιτιθέμενους να τα εντοπίσουν.

Επιπτώσεις: Η ευπάθεια μπορεί να έχει ως αποτέλεσμα την απώλεια ή καταστροφή των δεδομένων μέχρι και άρνηση εξυπηρέτησης ή ακόμα την απόκτηση δικαιωμάτων πλήρους διαχείρισης του συστήματος-στόχου.

Επιχειρηματικές επιπτώσεις: Πρέπει να ληφθεί σοβαρά υπόψη η επιχειρηματική αξία των επηρεαζόμενων στοιχείων ή των λειτουργιών της εφαρμογής, όπως και ο αντίκτυπος στη φήμη της επιχείρησης κατά τη δημοσιοποίηση της ευπάθειας, καθώς είναι πιθανό όλα τα δεδομένα να κλαπούν, να παραποιηθούν ή να διαγραφούν.

Τρόποι εντοπισμού της ευπάθειας: Ο καλύτερος τρόπος για να βρεθεί αν μια εφαρμογή είναι ευάλωτη στην ευπάθεια τύπου έγχυσης (injection) είναι να επιβεβαιώσουμε ότι η χρήση όλων των διερμηνευτών διαχωρίζει με σαφήνεια τα μη έμπιστα δεδομένα από μια εντολή ή μια επερώτηση (query). Για τις ερωτήσεις SQL αυτό σημαίνει, χρήση μεταβλητών δέσμευσης (bind variables) σε όλες τις προετοιμασμένες ερωτήσεις (prepared statements) και στις αποθηκευμένες διαδικασίες, με παράλληλη αποφυγή δυναμικά γραφόμενων ερωτημάτων.

Ο έλεγχος του κώδικα είναι ένας γρήγορος και αποτελεσματικός τρόπος για να καταλάβουμε αν η εφαρμογή χρησιμοποιεί τους διερμηνευτές με ασφαλή τρόπο.

[1] Με τον όρο «Legacy» αναφερόμαστε στον κώδικα που είτε προέρχεται από παλαιότερα συστήματα είτε έχει ως στόχο την επικοινωνία με αυτά

[2] Το Web Fuzzing είναι ένας από του πλέον αποτελεσματικούς τρόπους ανακάλυψης σφαλμάτων (bugs) που προκαλούν οι επιθέσεις τύπου άρνησης υπηρεσίας (DoS attacks), κυρίως XSS αλλά και SQL εκγύσεων (SQL Injections). Fuzzers είναι τα εργαλεία εκείνα που χρησιμοποιούνται για να ανιχνεύσουν τις ευπάθειες αυτές.

Τα εργαλεία ανάλυσης κώδικα μπορούν να βοηθήσουν έναν αναλυτή ασφαλείας να εντοπίσει την χρήση των διερμηνευτών και να παρακολουθήσει τη ροή των δεδομένων μέσω της εφαρμογής. Οι ελεγκτές εισβολής (penetration testers) μπορούν να επικυρώσουν αυτά τα προβλήματα δημιουργώντας exploits[3] που επιβεβαιώνουν την ευπάθεια.

Επίσης, η αυτοματοποιημένη δυναμική σάρωση που ελέγχει την εφαρμογή μπορεί να παράσχει πληροφόρηση σχετικά με το αν υπάρχουν εκμεταλλεύσιμα λάθη τύπου έγχυσης (injection). Οι σαρωτές δεν μπορούν πάντα να προσεγγίσουν τους διερμηνευτές και αντιμετωπίζουν δυσκολία στο να ανιχνεύσουν πότε μια επίθεση ήταν επιτυχής και πότε όχι. Ο κακός χειρισμός των σφαλμάτων κάνει την ευπάθεια τύπου έγχυσης ευκολότερη στην ανίχνευσή της.

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

1. Η προτεινόμενη επιλογή είναι η χρήση ενός ασφαλούς API το οποίο αποφεύγει τελείως τη χρήση του διερμηνευτή (interpreter) ή απλά παρέχει μια παραμετροποιημένη διεπαφή (interface). Εδώ πρέπει να δώσουμε προσοχή σε API's τύπου «αποθηκευμένες διαδικασίες» τα οποία είναι μεν παραμετροποιημένα αλλά μπορούν να παρουσιάσουν και αυτά ευπάθεια τύπου έγχυσης, με όχι ορατό τρόπο.
2. Αν δεν είναι διαθέσιμο ένα παραμετροποιημένο API, θα πρέπει προσεκτικά να αναιρέσουμε την ειδική σημασία των ειδικών χαρακτήρων χρησιμοποιώντας το ειδικό συντακτικό διαφυγής (escaping) για τον διερμηνευτή (interpreter).
3. Επίσης προτείνεται η θετική ή διαφορετικά «λευκή λίστα» επικύρωσης εισόδου, π.χ. με χρήση κανονικών εκφράσεων – με άλλα λόγια η θεώρηση εισόδων που περιέχουν ειδικούς χαρακτήρες ως μη παραδεκτές. Η μέθοδος αυτή ωστόσο δεν θεωρείται πλήρης άμυνα, καθώς αρκετές εφαρμογές μπορεί να απαιτούν τη δυνατότητα χρήσης ειδικών χαρακτήρων στην είσοδό τους. Το ESAPI του OWASP παρέχει μια εκτεταμένη βιβλιοθήκη, με αλγόριθμους επικύρωσης εισόδου λευκής λίστας.

2.2.2. A2 – Χρήση Scripts Μεταξύ Πολλαπλών Ιστοχώρων (Cross-Site Scripting, XSS)

Επιτιθέμενοι: Στην ευπάθεια του τύπου XSS, ο επιτιθέμενος μπορεί να είναι ο οποιοσδήποτε χρήστης συμπεριλαμβανομένων των εξωτερικών, των εσωτερικών χρηστών καθώς και των διαχειριστών, που επιδιώκει να στείλει, μη αξιόπιστα δεδομένα στον διερμηνευτή μας προκειμένου να εκτελεστούν ως κώδικας.

Είδος επίθεσης: Ο επιτιθέμενος στέλνει στην εφαρμογή πλοήγησης (browser) απλά scripts τύπου απλού κειμένου (text-based) τα οποία εκμεταλλεύονται τον διερμηνευτή. Σχεδόν οποιαδήποτε πηγή δεδομένων μπορεί να είναι φορέας της

[3] Είναι η μέθοδος με την οποία επιτυγχάνεται η εκμετάλλευση μίας αδυναμίας ή ενός bug του λογισμικού, προκειμένου να προκληθεί μια επίθεση. Από την στιγμή που ανακαλύπτεται μια αδυναμία, δημιουργείται και το ανάλογο exploit που μπορεί να την εκμεταλλευτεί.

επίθεσης, συμπεριλαμβανομένων των εσωτερικών πηγών όπως τα δεδομένα από τη βάση δεδομένων.

Αδυναμία ασφαλείας: Η ευπάθεια XSS είναι το πιο διαδεδομένο ελάττωμα ασφαλείας εφαρμογών ιστού (web). Οι ευπάθειες XSS μπορούν να συμβούν όταν μια εφαρμογή περιλαμβάνει σε μια σελίδα που αποστέλλεται στην εφαρμογή πλοήγησης (browser) δεδομένα που παρέχονται από τον χρήστη, χωρίς την κατάλληλη επικύρωση ή την χρήση διαφυγής (escaping) του περιεχομένου. Υπάρχουν τρία γνωστά είδη ευπάθειας XSS:

- Stored
- Reflected
- DOM based XSS

Η ανίχνευση της ευπάθειας είναι αρκετά εύκολη μέσω ελέγχων (testing) ή μέσω ανάλυσης κώδικα (code analysis).

Επιπτώσεις: Οι επιτιθέμενοι μπορούν να εκτελέσουν scripts στην εφαρμογή πλοήγησης (browser) του θύματος, να υποκλέψουν συνεδρίες (sessions) του χρήστη, να αλλοιώσουν ιστοσελίδες (web sites), να εισάγουν εχθρικό περιεχόμενο, να ανακατευθύνουν τους χρήστες, να ενσωματώσουν στην εφαρμογή πλοήγησης του χρήστη κακόβουλο λογισμικό (malware) κ.ά.

Επιχειρηματικές επιπτώσεις: Για την πλήρη αποτίμηση των επιχειρηματικών επιπτώσεων, θα πρέπει να αξιολογηθεί η επιχειρηματική αξία του πληγέντος συστήματος και όλων των δεδομένων που επηρεάζονται από αυτό, καθώς και την επίπτωση στην επιχείρηση από τη δημοσιοποίηση της ευπάθειας.

Τρόποι εντοπισμού της ευπάθειας: Για την αντιμετώπιση της ευπάθειας θα πρέπει να εξασφαλιστεί ότι όλα τα δεδομένα εισόδου του χρήστη τα οποία στέλνονται πίσω στην εφαρμογή πλοήγησης, έχουν επαληθευτεί (μέσω επαλήθευσης εισόδου) και ότι στα δεδομένα αυτά έχει γίνει χρήση διαφυγής (escaping) του περιεχομένου πριν αυτά να περιληφθούν στη σελίδα εξόδου. Η κατάλληλη κωδικοποίηση στην έξοδο διασφαλίζει ότι παρόμοιες είσοδοι πάντα αντιμετωπίζονται από την εφαρμογή πλοήγησης ως «κείμενο» (browser) και όχι ως ενεργό περιεχόμενο το οποίο μπορεί να είναι εκτελέσιμο με πιθανώς κακόβουλο σκοπό.

Τόσο τα στατικά όσο και τα δυναμικά εργαλεία μπορούν να ανακαλύψουν κάποια προβλήματα XSS αυτόματα. Παρόλα αυτά, κάθε εφαρμογή δημιουργεί σελίδες εξόδου με διαφορετικό τρόπο και χρησιμοποιεί διαφορετικούς διερμηνευτές (interpreters) στην πλευρά της εφαρμογής πλοήγησης, όπως JavaScript, activeX, Flash και Silverlight, κάτι το οποίο καθιστά τον αυτόματο εντοπισμό δύσκολο. Έτσι, μια ολοκληρωμένη κάλυψη του προβλήματος απαιτεί συνδυασμό από τη χειροκίνητη μελέτη του κώδικα και χειροκίνητες δοκιμές διείσδυσης, επιπρόσθετα σε κάθε αυτόματη προσέγγιση που επιλέγεται.

Οι τεχνολογίες Web2.0, όπως η AJAX, κάνουν την ευπάθεια XSS πολύ πιο δύσκολη στον εντοπισμό της, μέσω αυτοματοποιημένων εργαλείων εντοπισμού της ευπάθειας.

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

Η επιλογή που προτείνεται είναι να γίνει κατάλληλη χρήση διαφυγής (escaping) όλων των μη έμπιστων δεδομένων τα οποία βασίζονται στο πλαίσιο του προτύπου HTML (σώμα, ιδιότητες, Javascript, CSS ή URL) όπου τα δεδομένα μπορούν να τοποθετηθούν σε αυτό. Οι κατασκευαστές λογισμικού πρέπει να περιλαμβάνουν τη χρήση διαφυγής (escaping) στις εφαρμογές τους, εκτός αν το πλαίσιο διεπαφής χρήστη αναλαμβάνει τη διαδικασία αυτή με πλήρως αξιόπιστο τρόπο.

Η θετική ή λευκή λίστα (whitelist) επικύρωσης εισόδου επίσης προτείνεται, καθώς βοηθάει στην προστασία από την ευπάθεια XSS, αλλά αυτή δεν αποτελεί ολοκληρωμένη προστασία, καθώς αρκετές εφαρμογές μπορεί να απαιτούν τη δυνατότητα χρήσης ειδικών χαρακτήρων στην είσοδό τους. Μια τέτοια επικύρωση πρέπει να επικυρώνει το μήκος, τους χαρακτήρες και τη μορφή των δεδομένων εισόδου, προτού να κάνει αποδεκτή την είσοδο.

2.2.3. A3 – Επισφαλής Αυθεντικοποίηση και διαχείριση Συνόδου (Broken Authentication and Session Management)

Επιτιθέμενοι: Σε αυτή την περίπτωση οι επιτιθέμενοι μπορεί να είναι ανώνυμοι εξωτερικοί χρήστες ή χρήστες που διαθέτουν δικό τους λογαριασμό και προσπαθούν να κλέψουν λογαριασμούς από άλλους. Στην τελευταία κατηγορία περιλαμβάνονται οι χρήστες εκείνοι που προσπαθούν να αποκρύψουν τις ενέργειες τους, ενεργώντας κάτω από κλεμμένους λογαριασμούς.

Είδος επίθεσης: Οι επιτιθέμενοι χρησιμοποιούν κενά ασφαλείας στις λειτουργίες διαχείρισης ταυτότητας για να παραστήσουν κάποιον άλλο χρήστη.

Αδυναμία ασφαλείας: Οι κατασκευαστές λογισμικού συχνά φτιάχνουν δικά τους σχήματα αυθεντικοποίησης και ελέγχου της συνεδρίας (session) αλλά τα σχήματα αυτά δεν είναι εύκολο να μην έχουν λάθη. Το αποτέλεσμα είναι συνήθως να έχουν κενά ασφαλείας, όπως η έξοδος από την σελίδα, η διαχείριση κωδικών, η λήξη της συνόδου, οι κρυφές ερωτήσεις για είσοδο σε περίπτωση απώλειας συνθηματικού, ενημέρωση λογαριασμού κ.ά. Η εύρεση των ελαττωμάτων αυτών είναι συνήθως δύσκολη, καθώς κάθε εφαρμογή είναι μοναδική.

Επιπτώσεις: Ένα τέτοιο ελάττωμα του λογισμικού μπορεί να έχει ως συνέπεια την πρόσβαση του επιτιθέμενου σε έναν, σε μερικούς ακόμα και σε όλους τους λογαριασμούς. Συνήθως, οι λογαριασμοί που γίνονται στόχος είναι λογαριασμοί με υψηλά δικαιώματα ώστε να μπορεί ο επιτιθέμενος να έχει πλήρη πρόσβαση στην εφαρμογή.

Επιχειρηματικές επιπτώσεις: Πρέπει να ληφθεί σοβαρά υπόψη η επιχειρηματική αξία των επηρεαζόμενων στοιχείων ή των λειτουργιών της εφαρμογής, καθώς και ο αντίκτυπος στην φήμη της επιχείρησης κατά την δημοσιοποίηση της ευπάθειας.

Τρόποι εντοπισμού της ευπάθειας: Για να εντοπίσει κάποιος κατασκευαστής λογισμικού εάν είναι ασφαλής ως προς την ευπάθεια αυτή θα πρέπει να είναι σίγουρος ότι μπορεί να απαντήσει στις παρακάτω ερωτήσεις:

1. Είναι ασφαλή τα διαπιστευτήρια (credentials) σύνδεσης όταν αυτά αποθηκεύονται χρησιμοποιώντας κρυπτογράφηση (encryption) ή κερματισμό (hashing);

2. Μπορεί κάποιος να μαντέψει ή να αλλάξει τα διαπιστευτήρια κάνοντας χρήση αδύναμων, από άποψης ασφάλειας, λειτουργιών ελέγχου του λογαριασμού, όπως δημιουργία, αλλαγή κωδικού πρόσβασης ή ανάκτηση κωδικού του λογαριασμού;
3. Είναι τα αναγνωριστικά συνεδρίας (sessions ids) εκτεθειμένα στο URL;
4. Είναι τα αναγνωριστικά συνεδρίας (sessions id) ευάλωτα σε επιθέσεις υποκλοπής συνεδρίας;
5. Γίνεται σωστή λήξη της συνεδρίας; Μπορούν οι χρήστες να πραγματοποιήσουν έξοδο;
6. Δρομολογούνται οι κωδικοί, τα αναγνωριστικά συνεδρίας, και άλλα πιστοποιητικά μόνο πάνω από συνδέσεις TLS/SSL ώστε να εξασφαλίζεται ότι δεν μπορεί κάποιος κακόβουλος χρήστης να δει τι πληροφορίες μεταδίδονται;

Τρόποι αποτροπής της ευπάθειας: Η κυριότερη σύσταση για έναν οργανισμό προκειμένου να αποτρέψει αυτή την ευπάθεια είναι να θέσει στην διάθεση των κατασκευαστών λογισμικού ένα σύνολο ισχυρών εργαλείων ελέγχου αυθεντικοποίησης της συνεδρίας. Αυτά τα εργαλεία ελέγχου θα πρέπει να:

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

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

2.2.4. A4 – Επισφαλής άμεση αναφορά αντικειμένου (Insecure Direct Object References)

Επιτιθέμενοι: Κατηγορίες χρηστών οι οποίες έχουν δικαιώματα μερικής μόνο πρόσβασης σε ορισμένα είδη των δεδομένων του συστήματος.

Είδος επίθεσης: Στην περίπτωση αυτή ο επιτιθέμενος, ο οποίος είναι ένας εξουσιοδοτημένος χρήστης του συστήματος, απλά αλλάζει την τιμή μιας παραμέτρου η οποία αναφέρεται άμεσα σε ένα αντικείμενο, βάζοντας την τιμή ενός άλλου αντικειμένου στο οποίο δεν είναι εξουσιοδοτημένος να έχει πρόσβαση.

Αδυναμία ασφαλείας: Οι εφαρμογές συχνά χρησιμοποιούν το πραγματικό όνομα ή το κλειδί ενός αντικειμένου κατά τη δημιουργία κάποιων ιστοσελίδων. Όμως οι εφαρμογές δεν μπορούν πάντα να ελέγξουν εάν ο χρήστης έχει εξουσιοδότηση και αν πρέπει να έχει πρόσβαση στις σελίδες που αυτός στοχεύει. Αυτό οδηγεί σε μια επισφαλή άμεση αναφορά αντικειμένου (Insecure Direct Object Reference). Οι επιτιθέμενοι μπορούν εύκολα να αλλάξουν την τιμή μιας παραμέτρου και να εντοπίσουν προβλήματα αυτού του είδους, ενώ και η ανάλυση του κώδικα δείχνει άμεσα αν η αυθεντικοποίηση μπορεί να επαληθευτεί.

Επιπτώσεις: Αυτού του είδους ατέλειες μπορούν να θέσουν σε κίνδυνο όλα τα δεδομένα που σχετίζονται με την παράμετρο και να δώσουν στον εισβολέα πρόσβαση σε όλα τα διαθέσιμα στοιχεία αυτού του τύπου.

Επιχειρηματικές επιπτώσεις: Πρέπει να ληφθεί σοβαρά υπόψη η επιχειρηματική αξία των επηρεαζόμενων στοιχείων ή των λειτουργιών της εφαρμογής, καθώς και ο αντίκτυπος στην φήμη της επιχείρησης κατά την δημοσιοποίηση της ευπάθειας.

Τρόποι εντοπισμού της ευπάθειας: Ο καλύτερος τρόπος για να διαπιστώσουμε αν μια εφαρμογή είναι ευάλωτη στην συγκεκριμένη ευπάθεια είναι να εξακριβώσουμε ότι όλες οι αναφορές διαθέτουν την κατάλληλη άμυνα. Για να επιτευχθεί αυτό πρέπει να εξετάσουμε τα ακόλουθα:

1. Σχετικά με τις άμεσες αναφορές σε απαγορευμένους πόρους, η εφαρμογή θα πρέπει να ελέγχει αν ο χρήστης έχει το δικαίωμα πρόσβασης στους πόρους αυτούς.
2. Σχετικά με τις έμμεσες αναφορές, η απεικόνιση προς την άμεση αναφορά θα πρέπει να εκτελείται μόνο για δεδομένα στα οποία έχει πρόσβαση ο τρέχων χρήστης. Η επιθεώρηση του κώδικα της εφαρμογής μπορεί γρήγορα να εξακριβώσει εάν η προσέγγιση υλοποιείται με ασφάλεια. Οι δοκιμές είναι επίσης αποτελεσματικές για τον εντοπισμό άμεσων αναφορών αντικειμένων και για το κατά πόσο, είναι αυτές ασφαλείς. Τα αυτοματοποιημένα εργαλεία συνήθως δεν ερευνούν για ευπάθειες τέτοιου είδους διότι δεν μπορούν να αναγνωρίσουν τι χρειάζεται προστασία ή πιο συγκεκριμένα τι θεωρείται ασφαλές και τι όχι.

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

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

2.2.5. A5 – Παραχάραξη Αιτήσεων μεταξύ Πολλαπλών Ιστοχώρων (Cross-Site Request Forgery, CSRF)

Επιτιθέμενοι: Θεωρείται ο οποιοσδήποτε που θα μπορούσε να φορτώσει με πλάγια μέσα περιεχόμενο στις εφαρμογές πλοήγησης των χρηστών προκειμένου να τους οδηγήσει να υποβάλουν (παρά τη θέλησή τους) ένα αίτημα σε ιστοσελίδα. Η αίτηση αυτή θα μπορούσε να γίνει σε οποιαδήποτε ιστοσελίδα του φορέα στην οποία οι χρήστες έχουν πρόσβαση.

Είδος επίθεσης: Ο κακόβουλος χρήστης καθώς επιτίθεται, δημιουργεί παραχαραγμένα αιτήματα HTTP, και ξεγελά το θύμα με τρόπο που οδηγεί στην υποβολή αυτών των αιτημάτων Συνήθεις τρόποι για να ξεγελαστούν τα θύματα περιλαμβάνουν τα image tags, XSS, ή άλλες τεχνικές. Εάν ο χρήστης έχει περάσει τον έλεγχο αυθεντικοποίησης, τότε η επίθεση πετυχαίνει.

Αδυναμία ασφαλείας: Η Παραχάραξη Αιτήσεων μεταξύ Πολλαπλών Ιστοχώρων εκμεταλλεύεται εφαρμογές ιστού που επιτρέπουν στους επιτιθέμενους να προβλέψουν όλες τις λεπτομέρειες μιας συγκεκριμένης δράσης. Καθώς τα προγράμματα πλοήγησης (browsers) στέλνουν διαπιστευτήρια όπως cookies συνεδρίας αυτόματα, ο επιτιθέμενος μπορεί να δημιουργήσει κακόβουλες ιστοσελίδες που παράγουν πλαστά αιτήματα πανομοιότυπα με τα κανονικά. Η ανίχνευση των λαθών του CSRF είναι αρκετά εύκολη στον εντοπισμό της μέσω των δοκιμών διείσδυσης (penetration testing) ή μέσω της ανάλυσης του κώδικα (code analysis).

Επιπτώσεις: Ο κακόβουλος χρήστης καθώς επιτίθεται, μπορεί να αναγκάσει το θύμα να αλλάξει οποιαδήποτε πληροφορία που του επιτρέπεται να αλλάξει ή να πραγματοποιήσει οποιαδήποτε ενέργεια που το θύμα έχει την εξουσιοδότηση να κάνει.

Επιχειρηματικός αντίκτυπος: Πρέπει να ληφθεί σοβαρά υπόψη η επιχειρηματική αξία των επηρεαζόμενων στοιχείων ή των λειτουργιών της εφαρμογής καθώς και ο αντίκτυπος στην φήμη της επιχείρησης κατά την δημοσιοποίηση της ευπάθειας.

Τρόποι εντοπισμού της ευπάθειας: Ο ευκολότερος τρόπος να ελέγξει κάποιος αν η εφαρμογή είναι ευάλωτη σε αυτού του είδους την ευπάθεια, είναι να δει αν κάθε υπερσύνδεση (link) ή και φόρμα περιέχει ένα μη προβλέψιμο διακριτικό (token) για τον κάθε χρήστη. Χωρίς την ύπαρξη αυτού του απρόβλεπτου token ο επιτιθέμενος μπορεί να δημιουργήσει κακόβουλα αιτήματα. Επίσης, έμφαση πρέπει να δοθεί σε υπερσυνδέσεις (links) και φόρμες, που καλούν λειτουργίες οι οποίες αλλάζουν την κατάσταση της εφαρμογής, αφού αυτές οι υπερσυνδέσεις είναι κυρίως οι στόχοι μιας CSRF επίθεσης.

Ένας άλλος τρόπος εντοπισμού της ευπάθειας είναι να ελεγχθούν οι συναλλαγές πολλαπλών βημάτων, καθώς αυτές δεν είναι εγγενώς άτρωτες. Ο επιτιθέμενος μπορεί εύκολα να πραγματοποιήσει μια σειρά από αιτήματα, χρησιμοποιώντας πολλαπλά tags ή ενδεχομένως και javascript.

Σημειώνεται εδώ, ότι τα cookies συνεδρίας, η διεύθυνση IP και πιθανώς άλλες πληροφορίες που αποστέλλονται αυτόματα από το πρόγραμμα πλοήγησης (browser), δεν παρέχουν καμία προστασία έναντι επιθέσεων αυτής της κατηγορίας, καθώς αυτές οι πληροφορίες περιλαμβάνονται επίσης στα αιτήματα τα οποία είναι πλαστά.

Το CSRF tester tool του OWASP μπορεί να συμβάλει στη δημιουργία ελέγχων και να μας καταδείξει τους κινδύνους των ατελειών CSRF.

Τρόποι αποτροπής της ευπάθειας: Για την αποτροπή του CSRF απαιτείται να συμπεριλάβουμε ένα μη προβλέψιμο διακριτικό (Token) στο σώμα ή το URL κάθε HTTP αίτησης. Τέτοια διακριτικά πρέπει να είναι, τουλάχιστον μοναδικά ανά συνεδρία χρήστη, αλλά μπορεί να είναι μοναδικά και ανά αίτηση.

Η προτιμώμενη επιλογή είναι να συμπεριλάβουμε το μοναδικό διακριτικό σε ένα κρυφό πεδίο. Αυτό έχει ως συνέπεια, η τιμή να στέλνεται στο σώμα του αιτήματος HTTP, αποφεύγοντας την περίληψη του στο URL το οποίο εκτίθεται.

Το μοναδικό διακριτικό μπορεί ωστόσο να συμπεριληφθεί στο ίδιο το URL ή σε μια παράμετρο αυτού. Μια τέτοια τοποθέτηση όμως, διατρέχει τον κίνδυνο της έκθεσης του URL στον επιτιθέμενο, με αποτέλεσμα να τίθεται σε κίνδυνο το μυστικό διακριτικό.

Το **CSRF guard** του OWASP μπορεί να χρησιμοποιηθεί για να συμπεριλάβει αυτόματα τέτοια διακριτικά σε εφαρμογές java,.NET ή PHP.

Επίσης το **ESAPI** του OWASP περιλαμβάνει γεννήτριες (generators) και επαληθευτές (validators) ούτως ώστε οι προγραμματιστές να μπορούν να χρησιμοποιούν τα εργαλεία αυτά για την προστασία των συναλλαγών τους.

2.2.6. A6 – Επισφαλείς ρυθμίσεις ασφαλείας (Security Misconfiguration)

Επιτιθέμενοι: Μπορούν να θεωρηθούν ανώνυμοι εξωτερικοί χρήστες καθώς και χρήστες οι οποίοι διαθέτουν δικούς τους λογαριασμούς και επιχειρούν να θέσουν σε κίνδυνο το σύστημα αποκρύπτοντας τις κινήσεις τους.

Είδος επίθεσης: Ο επιτιθέμενος αποκτά πρόσβαση σε λογαριασμούς, σε αχρησιμοποίητες σελίδες, σε μη προστατευμένα αρχεία ή φακέλους προκειμένου να αποκτήσει μη εξουσιοδοτημένη πρόσβαση ή γνώση του συστήματος.

Αδυναμία ασφαλείας: Οι επισφαλείς ρυθμίσεις ασφαλείας μπορούν να συμβούν σε οποιοδήποτε σημείο της εφαρμογής συμπεριλαμβανομένης της πλατφόρμας, του εξυπηρέτη ιστού (web server), του εξυπηρέτη εφαρμογών (application server), του framework, όπως επίσης και του προσαρμοσμένου κώδικα. Οι προγραμματιστές και οι διαχειριστές δικτύου οφείλουν να συνεργαστούν για να εξασφαλίσουν ότι η εφαρμογή στο σύνολό της λειτουργεί σωστά. Οι αυτοματοποιημένοι ελεγκτές είναι χρήσιμοι για τον εντοπισμό κρίσιμων αναβαθμίσεων που απουσιάζουν, για τον εντοπισμό προγραμματιστικών λαθών, αχρησιμοποίητων υπηρεσιών καθώς και λογαριασμούς χρηστών με χαμηλό επίπεδο ασφαλείας.

Επιπτώσεις: Τέτοια λάθη στις ρυθμίσεις ασφαλείας, συχνά δίνουν στον επιτιθέμενο μη εξουσιοδοτημένη πρόσβαση σε ορισμένα δεδομένα ή σε συγκεκριμένες λειτουργείες του συστήματος ή ακόμα και στην ίδια την λειτουργικότητα του συστήματος. Υπάρχουν φυσικά και περιπτώσεις που τέτοια λάθη μπορούν να οδηγήσουν σε πλήρη παραβίαση του πληροφοριακού συστήματος.

Επιχειρηματικός αντίκτυπος: Το σύστημα θα μπορούσε να παραβιαστεί ολοκληρωτικά χωρίς να το γνωρίζει ο διαχειριστής. Όλα τα δεδομένα πιθανά να κλαπούν ή να τροποποιηθούν κατά τη βούληση του επιτιθέμενου και το κόστος επαναφοράς θα μπορούσε να κοστίσει αρκετά.

Τρόποι εντοπισμού της ευπάθειας: Για τον εντοπισμό της ευπάθειας αυτής όπως και οποιασδήποτε άλλης ευπάθειας, είναι αναγκαίο να υπάρχει η κατάλληλη τήρηση της ασφαλείας σε ολόκληρη τη δομή της εφαρμογής. Τα ερωτήματα που έχουμε να κάνουμε για να εντοπίσουμε την ευπάθεια πρέπει να απαντηθούν παρακάτω:

1. Υπάρχει η διαθέσιμη διαδικασία για έχουμε ενημερωμένο το λογισμικό, συμπεριλαμβανομένου του λειτουργικού συστήματος, του εξυπηρέτη ιστού (web server), του εξυπηρέτη εφαρμογών (application server) και όλων των βιβλιοθηκών κώδικα (code libraries);
2. Είναι απενεργοποιημένο ή απεγκατεστημένο οτιδήποτε περιττό στο εν λόγω σύστημα, όπως υπηρεσίες, θύρες, σελίδες, λογαριασμοί και προνόμια;
3. Έχουν αλλάξει ή είναι απενεργοποιημένοι οι προκαθορισμένοι λογαριασμοί των χρηστών;
4. Ο χειρισμός λαθών έχει παραμετροποιηθεί κατάλληλα έτσι ώστε να αποτρέπεται η διαρροή πολύτιμων πληροφοριών του συστήματος από μηνύματα λάθους;
5. Οι ρυθμίσεις ασφαλείας κατά την ανάπτυξη του συστήματος, έχουν κατανοηθεί σωστά και έχουν διαμορφωθεί κατάλληλα;

Οι αυστηρή και επαναλαμβανόμενη ρύθμιση ασφαλείας του συστήματος ενισχύει τη συνολική του ασφάλεια.

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

1. Μια επαναλαμβανόμενη διαδικασία ενίσχυσης της ασφάλειας που καθιστά γρήγορη και εύκολη την ανάπτυξη άλλου περιβάλλοντος που είναι κατάλληλα διασφαλισμένο. Το περιβάλλον ανάπτυξης, διασφάλισης ποιότητας και παραγωγής θα πρέπει να είναι επίσης διασφαλισμένο, με τρόπο αντίστοιχο με εκείνον του συστήματος παραγωγής. Η διαδικασία αυτή θα πρέπει να είναι αυτοματοποιημένη έτσι ώστε να ελαχιστοποιήσει την προσπάθεια που απαιτείται προκειμένου να εγκατασταθεί ένα νέο ασφαλές περιβάλλον.
2. Η διαδικασία για την ενημέρωση και εγκατάσταση των ενημερώσεων ασφαλείας και επιδιορθώσεων του λογισμικού, σχετικά σε εύθετο χρόνο σε όλα τα λειτουργούντα περιβάλλοντα. Η διαδικασία αυτή θα πρέπει να περιλαμβάνει οπωσδήποτε όλες τις βιβλιοθήκες κώδικα, οι οποίες συχνά παραλείπονται.
3. Μια ισχυρή αρχιτεκτονική της εφαρμογής η οποία παρέχει ασφαλή ασφάλεια ανάμεσα στα στοιχεία της.
4. Μια διαδικασία περιοδικής εκτέλεσης σαρώσεων και ελέγχων ασφάλειας βοηθά στον εντοπισμό των επισφαλών ρυθμίσεων ασφαλείας ή στον εντοπισμό μη εγκατεστημένων επιδιορθώσεων ασφαλείας.

2.2.7. A7 – Επισφαλής κρυπτογραφημένη αποθήκευση (Insecure Cryptographic Storage)

Επιτιθέμενοι: Σε αυτή την επισφάλεια πρέπει να δώσουμε σημασία στους εσωτερικούς χρήστες ή ακόμα και στους διαχειριστές ενός συστήματος οι οποίοι μπορεί να θέλουν να αποκτήσουν πρόσβαση σε προστατευμένα δεδομένα στα οποία οι ίδιοι δεν έχουν πρόσβαση.

Είδος επίθεσης: Στην περίπτωση αυτή, οι επιτιθέμενοι συνήθως δεν «σπάνε» την κρυπτογράφηση. Πράττουν κάτι άλλο, όπως το να επιδιώξουν να αποκτήσουν

κλειδιά, να αποκτήσουν αντίγραφα δεδομένων τα οποία είναι αποθηκευμένα σε μη κρυπτογραφημένο κείμενο (cleartext), ή να αποκτήσουν πρόσβαση σε δεδομένα που διέρχονται μέσω μη κρυπτογραφημένων καναλιών.

Αδυναμία ασφαλείας: Το ποιο κοινό λάθος που μπορεί να γίνει είναι, η μη κρυπτογράφηση δεδομένων που κανονικά θα έπρεπε να είναι κρυπτογραφημένα. Όταν όμως χρησιμοποιείται η κρυπτογράφηση, η μη ασφαλής αποθήκευση και διαδικασία παραγωγής κλειδιών και η χρήση αδύναμων από άποψης κρυπτογραφικής ισχύος αλγορίθμων είναι συνήθεις αδυναμίες. Συνήθης αδυναμία είναι ειδικότερα η χρήση αδύναμων από άποψης κρυπτογραφικής ισχύος συναρτήσεων κατακερματισμού (hashes) για την προστασία των κωδικών. Οι επιτιθέμενοι οι οποίοι είναι εξωτερικοί έχουν δυσκολίες στην ανίχνευση τέτοιων λαθών λόγων της περιορισμένης πρόσβασης στην εφαρμογή. Αυτοί συνήθως πρώτα πρέπει να εκμεταλλευτούν κάποια άλλη αδυναμία για να αποκτήσουν την απαραίτητη πρόσβαση.

Επιπτώσεις: Η αποτυχία στην ανίχνευση της ευπάθειας συχνά θέτει σε κίνδυνο όλα τα δεδομένα που θα έπρεπε να είχαν κρυπτογραφηθεί. Συνήθως αυτές οι πληροφορίες περιλαμβάνουν ευαίσθητα δεδομένα όπως ιατρικά δεδομένα, πιστοποιητικά, προσωπικά δεδομένα, δεδομένα πιστωτικών καρτών κ.ά.

Επιχειρηματικός αντίκτυπος: Πρέπει να λάβουμε υπόψη τον επιχειρηματικό αντίκτυπο που θα υπάρξει στην φήμη του οργανισμού, αν ο επιτιθέμενος καταφέρει και κλέψει τα δεδομένα. Επίσης σημασία πρέπει να δώσουμε στις νομικές συνέπειες που θα υπάρξουν όταν αποκαλυφθούν τα δεδομένα αυτά.

Τρόποι εντοπισμού της ευπάθειας: Το πρώτο πράγμα που πρέπει να κάνουμε είναι να καθορίσουμε ποια δεδομένα είναι αρκετά ευαίσθητα ώστε να απαιτούν κρυπτογράφηση. Ευαίσθητα δεδομένα μπορούν να θεωρηθούν οι κωδικοί πρόσβασης, πιστωτικές κάρτες, ιατρικά δεδομένα, προσωπικά δεδομένα και ό,τι άλλο κρίνουν οι υπεύθυνοι του συστήματος. Για τα δεδομένα αυτά πρέπει να διασφαλιστεί ότι:

1. Είναι κρυπτογραφημένα οπουδήποτε στο σύστημα, είναι αποθηκευμένα μακροπρόθεσμα και ιδιαίτερα στα αντίγραφα ασφαλείας μας.
2. Μέσω ελέγχου πρόσβασης, μόνο οι εξουσιοδοτημένοι χρήστες μπορούν να έχουν πρόσβαση στα αποκρυπτογραφημένα αντίγραφα των δεδομένων αυτών.
3. Χρησιμοποιείται ένας ισχυρός αλγόριθμος κρυπτογράφησης.
4. Χρησιμοποιείται ένα ισχυρό κλειδί, το οποίο προστατεύεται από μη εξουσιοδοτημένη πρόσβαση. Η αλλαγή του κλειδιού πρέπει να πραγματοποιείται σε προκαθορισμένα χρονικά παράθυρα συντήρησης.

Τρόποι αποτροπής της ευπάθειας: Για όλα τα ευαίσθητα δεδομένα που αξίζουν να είναι κρυπτογραφημένα θα πρέπει κατ’ ελάχιστο να γίνουν τα εξής:

1. Να ληφθούν υπόψη οι απειλές από τις οποίες κρίνεται σκόπιμο να προστατέψουμε τα δεδομένα, όπως από επίθεση εκ των έσω, ή επίθεση από εξωτερικούς χρήστες, και να βεβαιώσουμε ότι έχουμε κρυπτογραφήσει όλα τα δεδομένα κατά τέτοιο τρόπο ώστε να αμυνόμαστε ενάντια των απειλών αυτών.

2. Να σιγουρευτούμε πως έχουμε, εκτός συστήματος, εφεδρικά κρυπτογραφημένα αντίγραφα των δεδομένων αυτών με τα κλειδιά τους αποθηκευμένα ξεχωριστά.
3. Να βεβαιωθούμε πως χρησιμοποιούνται ισχυροί αλγόριθμοι και ισχυρά κλειδιά και ότι αυτά υπόκεινται σε ορθή διαχείριση.
4. Να διασφαλίσουμε ότι οι κωδικοί ασφαλείας κερματίζονται με έναν ισχυρό αλγόριθμο.
5. Όλα τα κλειδιά και οι κωδικοί πρόσβασης προστατεύονται από μη εξουσιοδοτημένη πρόσβαση.

2.2.8. A8 - Ανεπαρκής περιορισμός πρόσβασης URL– Failure to Restrict URL Access

Επιτιθέμενοι: Θεωρείται οποιοσδήποτε ο οποίος έχει πρόσβαση στο δίκτυο και μπορεί να στείλει στην εφαρμογή ένα αίτημα (request).

Είδος επίθεσης: Σε αρκετές περιπτώσεις, η προστασία των πόρων έγκειται μόνο στο ότι οι σύνδεσμοι προς τις λειτουργίες παρουσιάζονται μόνο στους χρήστες που δεν έχουν το δικαίωμα εκτέλεσης των λειτουργιών. Οι πιστοποιημένοι χρήστες αιτούνται το URL μιας σελίδας με αυξημένα δικαιώματα, ενώ οι ανώνυμοι (μη συνδεδεμένοι) χρήστες μπορούν να αποκτήσουν πρόσβαση σε ιδιωτικές σελίδες οι οποίες δεν προστατεύονται.

Αδυναμία ασφαλείας: εφαρμογές δεν προστατεύουν πάντα τις αιτήσεις των σελίδων κατάλληλα. Μερικές φορές τα URL προστατεύονται μέσω κάποιων ρυθμίσεων και το σύστημα δεν έχει ρυθμιστεί σωστά. Σε άλλες περιπτώσεις, οι κατασκευαστές λογισμικού ενώ πρέπει αν συμπεριλάβουν τους κατάλληλους ελέγχους στον κώδικα, αυτό δεν πραγματοποιείται από αμέλεια. Η ανίχνευση τέτοιων λαθών είναι εύκολη. Το πιο δύσκολο κομμάτι είναι η αναγνώριση των σελίδων (URL’s) που υπάρχουν για επίθεση.

Επιπτώσεις: Τέτοια λάθη επιτρέπουν στους επιτιθέμενους να αποκτήσουν πρόσβαση λειτουργίες για τις οποίες δεν είναι εξουσιοδοτημένοι. Οι λειτουργίες του διαχειριστή είναι στόχοι κλειδιά σε τέτοιου είδους επιθέσεις.

Επιχειρηματικός αντίκτυπος: Πρέπει να ληφθεί σοβαρά υπόψη η επιχειρηματική αξία των επηρεαζόμενων στοιχείων ή των λειτουργιών της εφαρμογής καθώς και ο αντίκτυπος στη φήμη της επιχείρησης κατά τη δημοσιοποίηση της ευπάθειας.

Τρόποι εντοπισμού της ευπάθειας: Ο καλύτερος τρόπος για να διαπιστώσει κάποιος ότι η εφαρμογή έχει αποτύχει να περιορίσει την πρόσβαση σε URL είναι να ελέγξει κάθε σελίδα της εφαρμογής. Ο διαχειριστής αποφασίζει αν η κάθε σελίδα πρέπει να είναι δημόσια ή ιδιωτική. Στην περίπτωση που η σελίδα είναι ιδιωτική τότε:

1. Απαιτείται πιστοποίηση για όποιον θέλει να έχει πρόσβαση στην σελίδα;
2. Θα έπρεπε αυτή να είναι προσβάσιμη σε οποιονδήποτε εξουσιοδοτημένο χρήστη; Αν όχι, με έναν έλεγχο εξουσιοδότησης μπορούμε να διασφαλίσουμε ότι ο χρήστης έχει δικαίωμα να προσπελάσει την σελίδα;

Εξωτερικοί μηχανισμοί ασφαλείας συχνά μας παρέχουν έλεγχο ταυτότητας και εξουσιοδότηση για την πρόσβαση σε σελίδες. Έτσι πρέπει να βεβαιωθούμε ότι οι έλεγχοι έχουν ρυθμιστεί σωστά για την κάθε σελίδα. Αν χρησιμοποιείται προστασία σε επίπεδο κώδικα, πρέπει να βεβαιωθούμε ότι αυτό συμβαίνει σε κάθε σελίδα. Ο έλεγχος ασφαλείας (Penetration Testing) μπορεί ωστόσο να επιβεβαιώσει κατά πόσο τα μέτρα ασφαλείας που λαμβάνονται είναι επαρκή ή όχι.

Τρόποι αποτροπής της ευπάθειας: Για την αποτροπή της ευπάθειας αυτής χρειάζεται η επιλογή μιας προσέγγισης που να απαιτεί κατάλληλη αυθεντικοποίηση και εξουσιοδότηση πιστοποίησης των χρηστών για την κάθε σελίδα. Συχνά, η προστασία αυτή παρέχεται από ένα ή περισσότερα στοιχεία, εξωτερικά του κώδικα της εφαρμογής. Ανεξάρτητα από τον μηχανισμό ή τους μηχανισμούς που χρησιμοποιούνται, συνίστανται τα παρακάτω:

1. Οι πολιτικές αυθεντικοποίησης και εξουσιοδότησης της ταυτότητας του χρήστη να είναι βασισμένες σε ρόλους, προκειμένου να ελαχιστοποιηθεί η προσπάθεια που απαιτείται για τη διατήρηση αυτών των πολιτικών.
2. Μηχανισμοί ελέγχου πρόσβασης θα πρέπει να απαγορεύουν συνολικά την πρόσβαση από προεπιλογή, απαιτώντας ρητές εξουσιοδοτήσεις σε συγκεκριμένους χρήστες για την πρόσβαση σε κάθε σελίδα.
3. Σε περίπτωση που η σελίδα εμπλέκεται σε μια ροή εργασίας, πρέπει να ελέγξουμε ότι οι συνθήκες είναι σε κατάσταση τέτοια ώστε να επιτρέπουν την πρόσβαση.
4. Πρέπει να υπάρχει επαρκής προστασία στο δικτυακό επίπεδο μεταφοράς.

2.2.9. A9 – Ανεπαρκής προστασία του επιπέδου μεταφοράς (Insufficient Transport Layer Protection)

Επιτιθέμενοι: Στην συγκεκριμένη ευπάθεια ο επιτιθέμενος μπορεί να είναι κάποιος που παρακολουθεί την δικτυακή κίνηση των χρηστών του δικτύου. Αν η εφαρμογή είναι στο διαδίκτυο, κανείς δεν ξέρει με ποιον τρόπο οι χρήστες την προσπελαύνουν.

Είδος επίθεσης: Το να παρακολουθεί κάποιος την δικτυακή κίνηση των χρηστών είναι κάτι το δύσκολο, αλλά όχι πάντοτε. Η κύρια δυσκολία έγκειται στην καταγραφή της δικτυακής κίνησης των χρηστών όταν αυτοί επισκέπτονται την ευάλωτη εφαρμογή.

Αδυναμία ασφαλείας: Οι εφαρμογές συχνά δεν προστατεύουν την κίνηση του δικτύου. Ένας τρόπος για να προστατευτούν τα δεδομένα που ανταλλάσσονται σε μια εφαρμογή είναι η χρήση SSL/TLS. Όμως πολλές εφαρμογές παραλείπουν να τα χρησιμοποιήσουν σε σημεία εκτός από την αυθεντικοποίηση των χρηστών, με αποτέλεσμα να εκτίθενται χρήσιμες πληροφορίες όπως δεδομένα και αναγνωριστικά συνεδρίας (session ids).

Η ανίχνευση των βασικών ελαττωμάτων μιας εφαρμογής είναι εύκολη. Είναι αρκετή, η παρατήρηση από τον επιτιθέμενο, της δικτυακής κίνησης και η υποκλοπή των κατάλληλων πληροφοριών.

Επιπτώσεις: Τέτοιου είδους σφάλματα της εφαρμογής εκθέτουν τα δεδομένα των χρηστών και μπορεί να οδηγήσουν σε κλοπή των λογαριασμών. Αν όμως ένας από τους παραβιασμένους λογαριασμούς είναι του διαχειριστή, τότε βρίσκεται

εκτεθειμένη ολόκληρη η ιστοσελίδα. Επίσης, μια ανίσχυρη εγκατάσταση SSL μπορεί να διευκολύνει επιθέσεις τύπου phishing [4]και MITM[5] (man in the middle).

Επιχειρηματικός αντίκτυπος: Πρέπει να ληφθεί σοβαρά υπόψη η επιχειρηματική αξία των επηρεαζόμενων στοιχείων που εκτίθενται στον επικοινωνιακό δίαυλο καθώς και η ανάγκη για τον έλεγχο της ταυτότητας των συμμετεχόντων.

Τρόποι εντοπισμού της ευπάθειας: Ο καλύτερος τρόπος για να εξακριβωθεί ότι μια εφαρμογή έχει επαρκή προστασία στο επίπεδο μεταφοράς (TLS) είναι πιστοποιήσουμε τις παρακάτω παραμέτρους:

1. Το SSL να χρησιμοποιείται για την προστασία όλης της κίνησης που σχετίζεται με την αυθεντικοποίηση.
2. Το SSL να χρησιμοποιείται για όλους τους πόρους, σε όλες τις σελίδες και τις υπηρεσίες. Αυτή η τακτική προστατεύει όλα τα δεδομένα που εκτίθενται. Η μικτή SSL σε μια ιστοσελίδα θα πρέπει να αποφεύγεται, δεδομένου ότι προκαλεί προειδοποιήσεις στους χρήστες μέσω των εφαρμογών πλοήγησης (browsers), που πιθανά θα ανησυχήσουν για την ασφάλειά τους και επίσης μπορεί να εκθέσει τα αναγνωριστικά συνεδρίας τους.
3. Να υποστηρίζονται μόνο ισχυροί αλγόριθμοι κρυπτογράφησης
4. Όλα τα cookies της συνεδρίας να έχουν ένα secure flag, έτσι ώστε η εφαρμογή πλοήγησης να μην μπορεί να τα μεταδίδει απροστάτευτα.
5. Το πιστοποιητικό του εξυπηρέτη (Server Certificate) να είναι νόμιμο και σωστά διαμορφωμένο για τον συγκεκριμένο εξυπηρέτη. Αυτό σημαίνει πως πρέπει να έχει εκδοθεί από εξουσιοδοτημένο εκδότη, να μην έχει λήξει, να μην έχει ανακληθεί και να ταιριάζει σε όλα τα πεδία (domains) που χρησιμοποιεί η ιστοσελίδα.

Τρόποι αποτροπής της ευπάθειας: Η παροχή κατάλληλης προστασίας στο επίπεδο μεταφοράς ίσως επηρεάσει τον σχεδιασμό της ιστοσελίδας. Είναι πιο εύκολο να απαιτηθεί SSL σε ολόκληρη την ιστοσελίδα. Για λόγους απόδοσης όμως, ορισμένες ιστοσελίδες χρησιμοποιούν SSL μόνο στις ιδιωτικές ιστοσελίδες. Κάποιοι άλλοι το χρησιμοποιούν μόνο σε σελίδες που θεωρούνται κρίσιμες (critical), αλλά αυτή η τακτική μπορεί να εκθέσει τα αναγνωριστικά εισόδου και άλλα ευαίσθητα δεδομένα.

Για να αποτραπεί η ευπάθεια, το ελάχιστο που μπορεί να γίνει είναι να τηρηθούν τα παρακάτω:

1. Να απαιτηθεί SSL για όλες τις ευαίσθητες σελίδες. Μη SSL αιτήματα σε αυτές τις σελίδες θα πρέπει να ανακατευθύνονται σε SSL σελίδες.
2. Να γίνει καθορισμός “secure flag” για όλα τα ευαίσθητα cookies.

[4] Μια μορφή επίθεσης κοινωνικής μηχανικής, με την οποία ο επιτιθέμενος δολίως, προσπαθεί να αποσπάσει από το θύμα προσωπικές πληροφορίες όπως κωδικούς πρόσβασης, λεπτομέρειες πιστωτικών καρτών κ.ά. υποδυόμενος κάποιον αξιόπιστο φορέα, όπως για παράδειγμα μια τράπεζα.

[5] Επίθεση κατά την οποία ο επιτιθέμενος μπορεί να υποκλέψει τα μηνύματα μεταξύ δυο σημείων τα οποία επικοινωνούν μεταξύ τους. Τα θύματα δεν είναι σε θέση να γνωρίζουν ότι η μεταδιδόμενη κίνηση (traffic) έχει παραβιαστεί.

3. Να γίνει διαμόρφωση του παρόχου SSL έτσι ώστε να υποστηρίζει μόνο ισχυρούς αλγόριθμους.
4. Να διασφαλιστεί ότι το πιστοποιητικό είναι έγκυρο, δεν έχει λήξει, δεν έχει ανακληθεί και ταιριάζει σε όλα τα πεδία (domains) που χρησιμοποιούνται από την ιστοσελίδα.
5. Συνδέσεις που γίνονται από την πλευρά του εξυπηρέτη (Backend) καθώς και άλλες συνδέσεις, θα πρέπει να χρησιμοποιούν και αυτές SSL ή άλλες τεχνολογίες κρυπτογράφησης.

2.2.10. A10 – Μη επικυρωμένες ανακατευθύνσεις και προωθήσεις (Unvalidated Redirects and Forwards)

Επιτιθέμενοι: Ο επιτιθέμενος συνήθως να είναι κάποιος ο οποίος μπορεί να παραπλανήσει τους χρήστες προκειμένου να υποβάλουν αίτηση (request) στην ιστοσελίδα. Αυτό μπορεί να συμβεί σε κάθε ιστοσελίδα που ο χρήστης έχει την δυνατότητα να αιτηθεί (request).

Είδος επίθεσης: Ο επιτιθέμενος, μέσω μη νόμιμων συνδέσμων προσπαθεί μέσω διαφόρων τρόπων να πείσει τους χρήστες να τους χρησιμοποιήσουν και να τους ανακατευθύνει εκεί όπου αυτός θέλει. Τα θύματα είναι πιθανό να κάνουν click στον σύνδεσμο, δεδομένου ότι ο σύνδεσμος αυτός ανήκει σε μια έγκυρη τοποθεσία. Ο στόχος του επιτιθέμενου είναι, μέσω των μη επικυρωμένων προωθήσεων να παρακάμψει τους ελέγχους ασφαλείας.

Αδυναμία ασφαλείας: Είναι συνηθισμένο φαινόμενο οι εφαρμογές να ανακατευθύνουν τους χρήστες σε άλλες σελίδες ή να χρησιμοποιούν εσωτερικές προωθήσεις με παρόμοιο τρόπο. Πολλές φορές δε, η σελίδα-στόχος καθορίζεται ως μία παράμετρος, η οποία δεν ελέγχεται ωστόσο επαρκώς, επιτρέποντας τους επιτιθέμενους να επιλέξουν την σελίδα στην οποία θα γίνει η προώθηση.

Η ανίχνευση των μη ελεγμένων ανακατευθύνσεων είναι εύκολη υπόθεση. Αρκεί να αναζητήσουμε ανακατευθύνσεις όπου μπορούμε να ορίσουμε την πλήρη διεύθυνση (URL). Όμως μη ελεγμένες προωθήσεις είναι δυσκολότερες στον εντοπισμό τους, δεδομένου ότι στοχεύουν σε εσωτερικές σελίδες.

Επιπτώσεις: Αν οι ανακατευθύνσεις έχουν ως σκοπό την εγκατάσταση κακόβολου λογισμικού (malware) ή να ξεγελάσουν τα θύματα στο να αποκαλύψουν κωδικούς ή άλλα ευαίσθητα δεδομένα, τότε η επισφάλεια αυτή έχει πολύ σημαντικές επιπτώσεις, διότι μπορεί να επιτρέψει την παράκαμψη του ελέγχου ασφαλείας του συστήματος.

Επιχειρηματικός αντίκτυπος: Πρέπει να ληφθεί σοβαρά υπόψη η επιχειρηματική αξία που θα είχε η επισφάλεια αυτή λόγω της απώλειας της εμπιστοσύνης των χρηστών, καθώς και των συνεπειών όσων χρηστών πληγούν από το κακόβουλο λογισμικό.

Επίσης, σοβαρά πρέπει να ληφθεί υπ’ όψη η φήμη της επιχείρησης κατά την δημοσιοποίηση της ευπάθειας.

Τρόποι εντοπισμού της ευπάθειας: Ο καλύτερος τρόπος για να εξακριβώσουμε ότι η εφαρμογή μας έχει μη επικυρωμένες ανακατευθύνσεις και προωθήσεις είναι να ελέγξουμε τις παρακάτω παραμέτρους.

1. Να εξετάσουμε τον κώδικα για όλες τις λειτουργίες των ανακατευθύνσεων ή των προωθήσεων. Για κάθε λειτουργία πρέπει να εξετασθεί αν το URL-στόχος υποδεικνύεται από κάποια παράμετρο που δίνεται από τον χρήστη. Αν ισχύει το παραπάνω, τότε πρέπει να επιβεβαιωθεί ότι η παράμετρος είναι επικυρωμένες έναντι μίας «λευκής λίστας» και συνεπώς υποδεικνύει μόνο έναν επιτρεπόμενο προορισμό.
2. Να κάνουμε πλήρη έλεγχο της σελίδας για να εξακριβωθεί αν δημιουργεί ανακατευθύνσεις. Στην συνέχεια, πρέπει να γίνει έλεγχος στις παραμέτρους πριν από την ανακατεύθυνση, για να φανεί αν εμφανίζονται ως ένα URL στόχος ή ένα κομμάτι του από μια τέτοια διεύθυνση URL. Αν το παραπάνω είναι αληθές, τότε πρέπει να αλλαχθεί το URL στόχος και να παρατηρήσουμε αν η ιστοσελίδα ανακατευθύνει στην νέα σελίδα στόχο.
3. Σε περίπτωση που ο κώδικας δεν είναι διαθέσιμος πρέπει να γίνει έλεγχος όλων των παραμέτρων για να φανεί αν αποτελούν μέρος μιας ανακατεύθυνσης ή προώθησης και στην συνέχεια να δοκιμαστεί η λειτουργία του.

Τρόποι αποτροπής της ευπάθειας: Ασφαλής χρήση των ανακατευθύνσεων ή των προωθήσεων σε μια εφαρμογή μπορεί να πραγματοποιηθεί με τους παρακάτω τρόπους.

1. Απλά, να σταματήσουμε να χρησιμοποιούμε αυτές τις μεθόδους.
2. Αν όμως, δεν μπορούμε να τις αποφύγουμε τότε, καλό είναι να μην χρησιμοποιούμε τους παραμέτρους των χρηστών στον υπολογισμό της σελίδας προορισμού.
3. Αν πάλι δεν μπορούμε να αποφύγουμε την εισαγωγή των παραμέτρων των χρηστών, πρέπει να βεβαιωθούμε ότι οι τιμές που δίνουν οι χρήστες στο σύστημα είναι έγκυρες και εξουσιοδοτημένες για τον χρήστη.

Συνίσταται δε, πως οι τιμές των παράμετροι προορισμού να μην είναι το ίδιο το URL αλλά ένας κωδικός, ο οποίος μέσω απεικόνισης στην πλευρά του εξυπηρέτη να οδηγεί στο URL στόχο.

Μπορούμε να χρησιμοποιήσουμε το ESAPI του OWASP για να κάνουμε έλεγχο όλων των ανακατευθύνσεων και προωθήσεων που έχει η εφαρμογή που ελέγχουμε, προκειμένου να σιγουρευτούμε ότι είμαστε ασφαλείς.

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

3. NIST National Institute of Standards and Technology (NIST)

3.1. Γενικά για το Ίδρυμα NIST

Το Εργαστήριο Πληροφορικής (ITL) στο Εθνικό Ινστιτούτο Προτύπων και Τεχνολογίας (National Institute of Standards and Technology), αναπτύσσει δοκιμές, τις μεθόδους αυτών, τα δεδομένα αναφοράς και την τεχνική ανάλυση που αφορούν την προώθηση και την ανάπτυξη της παραγωγικότητας της τεχνολογίας των πληροφοριών. Στις ευθύνες του εργαστηρίου περιλαμβάνονται, η ανάπτυξη τεχνικών προτύπων καθώς και η διαχείριση αυτών.

Η συλλογή με τα πρότυπα της σειράς 800 (800-series), σχετίζεται με την διαχείριση της ασφάλειας των πληροφοριών. Ο οδηγός NIST SP 800-115 “Technical Guide to Information Security Testing and Assessment” που σε αυτόν βασίζεται κατά κύριο λόγο η συγγραφή του κεφαλαίου αυτού, ασχολείται με τον έλεγχο της ασφάλειας των δικτύων και αναφέρεται στις βασικές τεχνικές πτυχές σχετικά με την διαχείριση των αξιολογήσεων της ασφάλειας των πληροφοριών. Παρουσιάζει μεθόδους και τεχνικές ελέγχου που ένας οργανισμός μπορεί να ακολουθήσει προκειμένου να αξιολογήσει τα συστήματα και τα δίκτυα του. Πραγματοποιεί πρακτικές συστάσεις σχετικά με τον σχεδιασμό, την υλοποίηση των ελέγχων ασφαλείας και των διαδικασιών αξιολόγησης για την εύρεση ευπαθειών σε ένα σύστημα ή ένα δίκτυο.

3.2. Μεθοδολογία και τεχνικές αποτίμησης της ασφάλειας των πληροφοριών

Η αποτίμηση της ασφάλειας των πληροφοριών είναι η διαδικασία που μας επιτρέπει να καθορίσουμε κατά πόσο αποτελεσματικό είναι το σύστημα που αξιολογούμε, καθώς και το αν πληροί συγκεκριμένους στόχους ασφάλειας που έχουμε θέσει για αυτό. Οι μέθοδοι που μπορούν να χρησιμοποιηθούν για την επίτευξη της αποτίμησης περιλαμβάνουν τρία στάδια. Τον **έλεγχο (testing)**, την **εξέταση (examination)** και την **συνέντευξη (interviewing)**.

Ο έλεγχος είναι η διαδικασία αξιολόγησης ενός ή περισσοτέρων αντικειμένων υπό συγκεκριμένες συνθήκες, με σκοπό τη σύγκριση μεταξύ πραγματικής και αναμενόμενης συμπεριφοράς. **Η εξέταση** είναι η διαδικασία ελέγχου μέσω της επιθεώρησης, της επανεξέτασης, της παρατήρησης, της μελέτης ή της ανάλυσης ενός ή περισσοτέρων αντικειμένων αξιολόγησης, για τη διευκόλυνση της κατανόησης, την επίτευξη της διευκρίνισης, ή τη συγκέντρωσης αποδεικτικών στοιχείων. **Η συνέντευξη** είναι η διαδικασία της διεξαγωγής των συζητήσεων με άτομα ή ομάδες σε έναν οργανισμό, για να διευκολύνει την κατανόηση, να επιτευχθεί η διευκρίνιση και να προσδιοριστεί η θέση των αποδεικτικών στοιχείων.

Τα αποτελέσματα της αξιολόγησης χρησιμοποιούνται από τον οργανισμό ή την εταιρεία για τον προσδιορισμό της αποτελεσματικότητας του ελέγχου.

3.2.1. Μεθοδολογία για την αξιολόγηση της ασφάλειας

Η μεθοδολογία της αξιολόγησης της ασφάλειας των πληροφοριών που προτείνεται από το NIST είναι οργανωμένη σε στάδια, διάρθρωση που προσφέρει

μια σειρά από πλεονεκτήματα και επιτυγχάνει αποδοτική χρήση του προσωπικού και των πόρων του οργανισμού, κατά την πραγματοποίηση της αξιολόγησης αυτής. Τα στάδια που περιλαμβάνονται είναι τα ακόλουθα:

- **Σχεδιασμός.** Το στάδιο αυτό είναι ζωτικής σημασίας για την επιτυχή αξιολόγηση της ασφάλειας. Χρησιμοποιείται για τη συλλογή των πληροφοριών που απαιτούνται για την αποτίμηση της αξιολόγησης, καθώς και για τους ελέγχους που πρέπει να γίνουν για τον εντοπισμό των απειλών και την ανάπτυξη της προσέγγισης της αξιολόγησης. Οι πληροφορίες αυτές είναι οι ακριβείς πόροι για τους οποίους θα πραγματοποιηθεί ο έλεγχος, οι συνήθεις ευπάθειες που τους διέπουν και οι δικλείδες ασφαλείας που τους προστατεύουν. Στην φάση αυτή περιλαμβάνονται και οι απαιτήσεις που διέπουν την διαχείριση ενός έργου όπως είναι ο σκοπός των δοκιμών, το πεδίο εφαρμογής, οι λειτουργικές απαιτήσεις, οι ρόλοι και οι ευθύνες της ομάδας, οι αναγκαίοι περιορισμοί των ελέγχων, τα παραδοτέα και το χρονοδιάγραμμα των εργασιών.
- **Εκτέλεση.** Οι πρωταρχικοί στόχοι για το στάδιο της εκτέλεσης είναι η αναγνώριση των ευπαθειών και η επικύρωσή τους, όταν αυτό απαιτείται. Η φάση αυτή εξετάζει τις δραστηριότητες εκείνες, που συνδέονται με τις μεθόδους αξιολόγησης που αποφασίστηκαν στην φάση του σχεδιασμού. Μολονότι οι συγκεκριμένες δραστηριότητες μπορεί να διαφέρουν ανάλογα με τον τύπο αξιολόγησης, μετά την ολοκλήρωση αυτής της φάσης, οι αξιολογητές θα έχουν εντοπίσει το σύστημα, το δίκτυο, και τις οργανωτικές αδυναμίες της διαδικασίας. Αν μια ευπάθεια ανιχνευτεί κατά τη διάρκεια της διαδικασίας αυτής, οι αξιολογητές πρέπει να ακολουθήσουν τις διαδικασίες που προβλέπει ο οργανισμός για την συγκεκριμένη περίπτωση. Με τον τρόπο αυτό μας επιτρέπει να εξετάσουμε και να αναλύσουμε τα αίτια της ευπάθειας. Σύμφωνα με τις ορθές προβλεπόμενες πρακτικές ασφαλείας τα δεδομένα που συλλέχθηκαν με την διαδικασία αυτή πρέπει να καταστραφούν.
- **Μετά την εκτέλεση**. Η φάση «μετά την εκτέλεση» επικεντρώνεται στην ανάλυση των ευπαθειών που εντοπίστηκαν ώστε να καθοριστούν οι βαθύτερες αιτίες, η διατύπωση των συστάσεων για τις προτάσεις αντιμετώπισης των ευπαθειών και η ανάπτυξη μιας έκθεσης η οποία θα ενσωματώνει τις προτάσεις αυτές.

Υπάρχουν αρκετές μεθοδολογίες σχετικά με την διεξαγωγή αξιολογήσεων της ασφάλειας των πληροφοριών. Η NIST έχει δημιουργήσει μια μεθοδολογία που έχει αναπτυχθεί στην ειδική Έκδοση SP 800-53A, με τίτλο “Guide for Assessing the Security Controls in Federal Information Systems”, η οποία προσφέρει προτάσεις για την αξιολόγηση της αποτελεσματικότητας των ελέγχων ασφαλείας που περιγράφονται στο NIST SP 800-53. Μια άλλη ευρέως χρησιμοποιούμενη μεθοδολογία αξιολόγησης, είναι το Open Source Security Testing Methodology Manual (OSSTMM).

Είναι συνηθισμένη τακτική, όταν ένας Οργανισμός θέλει να κάνει τις εκτιμήσεις του όσον αφορά την ασφάλεια των υπολογιστικών του συστημάτων, να χρησιμοποιεί πολλαπλές μεθοδολογίες.

3.2.2. Τεχνικές αξιολόγησης της ασφάλειας

Για την αξιολόγηση της ασφάλειας των συστημάτων και των δικτύων υπολογιστών, υπάρχουν δεκάδες τεχνικές που μπορούν να χρησιμοποιηθούν. Οι πιο συχνά χρησιμοποιούμενες, ομαδοποιούνται στις ακόλουθες τρεις κατηγορίες:

3.2.2.1. Τεχνικές αξιολόγησης

Είναι τεχνικές ελέγχου που χρησιμοποιούνται για την αξιολόγηση των συστημάτων, των εφαρμογών, των δικτύων, των πολιτικών και των διαδικασιών που χρειάζονται για να εντοπιστούν υπάρχουσες ευπάθειες. Περιλαμβάνονται οι παρακάτω επιμέρους τεχνικές:

- **Επισκόπηση τεκμηρίωσης (Document Review)**. Η επισκόπηση της τεκμηρίωσης εξετάζει εάν οι τεχνικές προδιαγραφές των πολιτικών και των διαδικασιών είναι ενημερωμένες και πλήρεις. Τα έγγραφα αυτά μπορεί να είναι ουσιαστικά τα θεμέλια για την ασφάλεια ενός οργανισμού, συχνά όμως παραβλέπονται κατά τη διάρκεια της τεχνικής αξιολόγησης. Η επισκόπηση της τεκμηρίωσης μπορεί να ανακαλύψει κενά και αδυναμίες που πιθανά να οδηγούν σε κακές πρακτικές στους ελέγχους ασφαλείας. Οι αξιολογητές πρέπει να βεβαιωθούν ότι τα έγγραφα της οργάνωσης είναι συμβατά με τα πρότυπα και τους κανονισμούς, και να αναζητήσει τις πολιτικές αυτές που είναι ελλιπείς ή ανενημέρωτες. Η επισκόπηση της τεκμηρίωσης δεν διασφαλίζει ότι οι έλεγχοι εφαρμόζονται ορθά. Δίνει την κατεύθυνση και την καθοδήγηση για να στηριχθούν οι υπάρχουσες υποδομές ασφαλείας.
- **Επισκόπηση καταγραφής (Log review).** Εξετάζει εάν έλεγχοι ασφαλείας καταγράφουν τις κατάλληλες πληροφορίες και εάν ο οργανισμός ακολουθεί τις κατάλληλες πολιτικές διαχείρισης των αρχείων καταγραφής. Τα αρχεία καταγραφής, ως πηγή ιστορικών πληροφοριών, μπορούν να χρησιμοποιηθούν για να επιβεβαιώσουν ότι το σύστημα λειτουργεί σύμφωνα με τις πολιτικές που έχουν τεθεί. Η επισκόπηση των αρχείων καταγραφής μπορεί να αποκαλύψει προβλήματα όπως υπηρεσίες που εμπεριέχουν λάθη στον προγραμματισμό και στον έλεγχο ασφαλείας, προσβάσεις στο σύστημα χωρίς εξουσιοδότηση, καθώς και επιτυχημένες ή αποτυχημένες απόπειρες εισβολής. Παραδείγματα πληροφοριών καταγραφής, που μπορεί να μας είναι χρήσιμα κατά την διεξαγωγή των τεχνικών αξιολογήσεων ασφαλείας είναι:
    - Αρχεία καταγραφής του εξυπηρέτη αυθεντικοποίησης (authentication server), που περιλαμβάνουν επιτυχημένες και αποτυχημένες αιτήσεις αυθεντικοποίησης.
    - Αρχεία καταγραφής του συστήματος, που περιλαμβάνουν πληροφορίες για την εκκίνηση και τον τερματισμό υπηρεσιών (services) ή του συστήματος (system), πληροφορίες για εγκατάσταση μη εξουσιοδοτημένου λογισμικού, για προσβάσεις αρχείων, αλλαγές στην πολιτική ασφαλείας καθώς και

συνδέσεις λογαριασμών με υψηλά δικαιώματα (root, administrator).

- Αρχεία καταγραφής από τείχη προστασίας (firewalls), δρομολογητές (routers) και συστήματα ανίχνευσης επιθέσεων (Intrusion Detection Systems) που μπορεί να περιλαμβάνουν ενδείξεις για κίνδυνο, από εσωτερικές συσκευές του δικτύου (rootkits, bots, Trojan horses, spyware) και ύποπτη δραστηριότητα από εξωτερικές συσκευές.
- Αρχεία καταγραφής από τείχη προστασίας (firewalls), που μπορεί να περιλαμβάνουν τις προσπάθειες μη εξουσιοδοτημένης σύνδεσης καθώς και ακατάλληλη χρήση.
- Αρχεία καταγραφής εφαρμογών, που μπορεί να περιλαμβάνουν προσπάθειες μη εξουσιοδοτημένης σύνδεσης, αλλαγές των λογαριασμών των χρηστών, χρήση προνομίων και πληροφορίες για τη χρήση των βάσεων δεδομένων.
- Αρχεία καταγραφής λογισμικού αντιμετώπισης ιών (antivirus), που μπορεί να περιλαμβάνουν αποτυχημένες ενημερώσεις και άλλες ενδείξεις για μη ενημερωμένο λογισμικό.
- Αρχεία καταγραφής συστημάτων IDS/IPS[6], που μπορεί να καταγράψει πληροφορίες σχετικά με ύποπτη δραστηριότητα υπηρεσιών και εφαρμογών.

- **Επισκόπηση συνόλου κανόνων.** Είναι μια συλλογή κανόνων ή υπογραφών κατά την οποία η κίνηση του δικτύου, ή η δραστηριότητα του συστήματος, συγκρίνεται, για να καθοριστεί ποια μέτρα πρέπει να ληφθούν για την αποδοχή ή την απόρριψη κάποιων συγκεκριμένων συμβάντων. Η επισκόπηση του συνόλου των κανόνων, γίνεται για να εξασφαλιστεί η πληρότητα, να εντοπιστούν τα κενά και οι αδυναμίες των συστημάτων ασφαλείας, να φανερωθούν τα τρωτά σημεία του δικτύου και οι παραβιάσεις στην πολιτικής ασφαλείας.

  Η επισκόπηση του συνόλου των κανόνων σε ένα δίκτυο συνήθως περιλαμβάνει: τους κανόνες του firewall, τους κανόνες του συστήματος

[6] **IDS (Instruction Detection System)**. Είναι ένα σύστημα ανίχνευσης εισβολής. Παρακολουθεί και αναλύει συμβάντα, τα οποία λαμβάνουν χώρα στα συστήματα και στο δίκτυο των υπολογιστών. Ο στόχος του είναι ο εντοπισμός ενδείξεων για πιθανές προσπάθειες εισβολής, κατά τις οποίες συχνά εντοπίζονται ίχνη παραβίασης της ακεραιότητας, της εμπιστευτικότητας και της διαθεσιμότητας των πληροφοριακών πόρων. Οι προσπάθειες παράκαμψης των μηχανισμών ασφαλείας μπορεί να προέρχονται από εξωτερικούς χρήστες, προς το εσωτερικό εταιρικό δίκτυο, στους οποίους δεν επιτρέπεται η πρόσβαση στο υπάρχον πληροφοριακό σύστημα, καθώς και από εσωτερικούς χρήστες, με περιορισμένα δικαιώματα πρόσβασης.

IPS (Instruction Prevention System). Είναι ένα σύστημα πρόληψης εισβολής. Παρακολουθεί και ελέγχει κάθε πρόσβαση στο δίκτυο για να προστατεύσει το σύστημα από κακόβουλους χρήστες. Ο έλεγχος της πρόσβασης γίνεται με γνώμονα την εφαρμογή και όχι τη διεύθυνση IP ή τη θύρα (port), που είναι τρόπος ελέγχου των firewall.

IDS/IPS και τις λίστες ελέγχου πρόσβασης (access–lists[7]) του δρομολογητή. Στη συνέχεια, φαίνονται τα παραδείγματα των ελέγχων που εκτελούνται πιο συχνά κατά την επισκόπηση του συνόλου των κανόνων.

Για λίστες ελέγχου πρόσβασης δρομολογητών (routers). Κάθε κανόνας που έχουμε θέσει, να είναι χρήσιμος και επίκαιρος και να έχουν απομακρυνθεί οι προσωρινοί κανόνες ή αυτοί που δεν χρειάζονται πλέον. Να είναι επιτρεπτή η κίνηση (traffic) η οποία είναι εγκεκριμένη από την πολιτική ασφαλείας και όλη η υπόλοιπη να απορρίπτεται από προεπιλογή.

Για σύνολο κανόνων τείχους προστασίας (firewalls).

- Κάθε κανόνας να εξακολουθεί να απαιτείται.
- Οι κανόνες να εφαρμόζουν την αρχή του ελάχιστου προνομίου. επιτρέποντας την πρόσβαση σε συγκεκριμένες διευθύνσεις IP και σε συγκεκριμένες θύρες.
- Να μην υπάρχουν ανοικτές θύρες οι οποίες να μην χρειάζεται να είναι ανοικτές. Αν ισχύει κάτι τέτοιο πρέπει να κλείσουν αμέσως.
- Το σύνολο κανόνων δεν επιτρέπει στη δικτυακή κίνηση να παρακάμψει άλλες άμυνες ασφαλείας
- Για τα τείχη προστασίας που εκτελούνται σε προσωπικούς υπολογιστές, να μην υπάρχουν κανόνες που να παραπέμπουν σε παρουσία κερκόπορτων (backdoors), δραστηριότητα spyware και τη λειτουργία εφαρμογών ανταλλαγής αρχείων ή απαγορευμένων εφαρμογών.

Για το σύνολο κανόνων των IDS / IPS. Να ενεργοποιηθούν οι απαραίτητες υπογραφές, να απενεργοποιηθούν οι περιττές υπογραφές και να γίνει σωστή συντήρηση και λεπτομερής ρύθμιση του συστήματος.

- **Επισκόπηση της διαμόρφωσης του συστήματος.** Είναι η διαδικασία του εντοπισμού των αδυναμιών που πιθανά να υπάρχουν, κατά τον έλεγχο των ρυθμίσεων ασφαλείας, στα αρχεία διαμόρφωσης του συστήματος. Αυτό το είδος ελέγχου μπορεί να αποκαλύψει περιττές υπηρεσίες και εφαρμογές, ακατάλληλους λογαριασμούς χρηστών και ρυθμίσεις κωδικών πρόσβασης, ακατάλληλες ρυθμίσεις καταγραφής και ακατάλληλες ρυθμίσεις στη διαδικασία αποθήκευσης των αντιγράφων ασφαλείας.

[7] **Access Control List (ACL)**: Λίστα ελέγχου πρόσβασης για δρομολογητητές (routers) και τείχη προστασίας (firewalls) με σκοπό την προστασία των υποδομών του δικτύου. Χρησιμοποιείται για την προστασία και την ελαχιστοποίηση του κινδύνου σε ένα δίκτυο, καθώς και για την προστασία των υποδομών, επιτρέποντας μεν τη δικτυακή κίνηση (traffic) από και προς τις υποδομές που θέλουμε, απαγορεύοντας δε, όλη την υπόλοιπη κίνηση.

Οι αξιολογητές, χρησιμοποιούν χειροκίνητες τεχνικές επισκόπησης και στηρίζονται σε οδηγούς διαμόρφωσης ασφάλειας ή σε λίστες ελέγχου, για να εξακριβώσουν ότι οι ρυθμίσεις του αξιολογηθέντος συστήματος, έχουν ελαχιστοποιημένο ρίσκο ασφαλείας. Κατά τη διαδικασία της επισκόπησης του συστήματος διαμόρφωσης, οι αξιολογητές αποκτούν πρόσβαση σε διάφορες ρυθμίσεις ασφαλείας της συσκευής που αξιολογούν και τις συγκρίνουν με τις προτεινόμενες ρυθμίσεις των λιστών ελέγχου. Οι ρυθμίσεις που δεν πληρούν τις ελάχιστες προδιαγραφές ασφαλείας, πρέπει να επισημαίνονται και αναφέρονται.

Η μέθοδος SCAP (System Content Automation Protocol) κάνει χρήση συγκεκριμένων προτύπων, ώστε να πραγματοποιεί αυτοματοποιημένη διαχείριση ευπάθειας, μέτρηση, καθώς και αποτίμηση του βαθμού συμμόρφωσης με τις πολιτικές. Φυσικά, υπάρχουν και άλλα αυτοματοποιημένα εργαλεία, με τα οποία οι αξιολογητές εκτελούν ελέγχους απευθείας στη συσκευή που είναι προς αξιολόγηση ή και κατευθείαν σε ένα σύστημα το οποίο ανήκει στο δίκτυο.

Οι αυτοματοποιημένες μέθοδοι μπορούν να εκτελεστούν ταχύτερα και δίνουν συνεπή αποτελέσματα. Ο χειροκίνητος έλεγχος είναι κουραστικός και επιρρεπής σε λάθη. Παρόλα αυτά, υπάρχουν ρυθμίσεις που πρέπει να ελέγχονται χειροκίνητα. Και οι δυο μέθοδοι απαιτούν αυξημένα δικαιώματα (administrator ή root) για τη διαχείριση των ρυθμίσεων ασφαλείας. Όμως, σε γενικές γραμμές, όπου αυτό είναι εφικτό, είναι προτιμότερο να χρησιμοποιούμε τους αυτόματους ελέγχους αντί των χειροκίνητων.

- **Παρακολούθηση, καταγραφή δικτύου.** Η παρακολούθηση της δικτυακής κίνησης ενός δικτύου (sniffing), είναι μια παθητική τεχνική που παρακολουθεί την επικοινωνία του δικτύου, αποκωδικοποιεί πρωτόκολλα, και εξετάζει τις κεφαλίδες και το περιεχόμενο των πακέτων που παρουσιάζουν ενδιαφέρον για αυτόν που παρακολουθεί. Εκτός από την χρησιμότητα του ως τεχνική επισκόπησης, χρησιμοποιείται και για αναγνώριση στόχου και τεχνική ανάλυση. Μερικοί από τους λόγους που μας ωθούν να κάνουμε χρήση του sniffing δίνονται παρακάτω:
    - Η σύλληψη (capturing) και αναπαραγωγή (replaying) της δικτυακής κίνησης.
    - Η ανίχνευση και προσδιορισμός των ενεργών συσκευών στο δίκτυο.
    - Ο προσδιορισμός των λειτουργικών συστημάτων, των εφαρμογών, των υπηρεσιών, και των πρωτοκόλλων που υφίστανται και μεταδίδονται στο υπό παρακολούθηση δίκτυο. Επίσης ανακάλυψη μη εγκεκριμένων και μη ασφαλών πρωτοκόλλων όπως peer-to peer και telnet, αντίστοιχα.

- Ο προσδιορισμός μη εξουσιοδοτημένων και ακατάλληλων δραστηριοτήτων, όπως η μη κρυπτογραφημένη μετάδοση ευαίσθητων πληροφοριών.
- Η συλλογή πληροφοριών, όπως τα μη κρυπτογραφημένα ονόματα χρηστών καθώς και οι κωδικοί πρόσβασης αυτών.

Η παρακολούθηση του δικτύου έχει μικρό αντίκτυπο στα συστήματα και τα δίκτυα, με πιο αξιοσημείωτη επίπτωση, στο εύρος ζώνης (bandwidth) και στην αυξημένη υπολογιστική ισχύ που απαιτείται στις εμπλεκόμενες ενεργές συσκευές του δικτύου. Το εργαλείο που χρησιμοποιείται για τη διεξαγωγή της παρακολούθησης του δικτύου (sniffing tool) απαιτεί την ύπαρξη σύνδεσης με το δίκτυο, όπως η σύνδεση σε hub ή μεταγωγέα (switch) με λειτουργία port spanning[8]. Συνήθως οι οργανισμοί αναπτύσσουν sniffers στο δίκτυο τους σε συγκεκριμένες θέσεις του δικτύου. Συνήθως είναι οι εξής:

- Στην περίμετρο, για την πρόσβαση στην κυκλοφορία που εισέρχεται και εξέρχεται από το δίκτυο.
- Πίσω από τείχη προστασίας (firewalls), για πρόσβαση στην επιτρεπόμενη κυκλοφορία από το firewall και την εκτίμηση των κανόνων φιλτραρίσματος καθώς και την ορθή λειτουργία τους.
- Πίσω IDS / IPS, προκειμένου να διαπιστωθεί εάν οι υπογραφές είναι οι κατάλληλες και ανταποκρίνονται ορθά.
- Μπροστά από ένα κρίσιμο για το δίκτυο σύστημα ή μια εφαρμογή και την εκτίμηση της δραστηριότητας του.
- Σε ένα συγκεκριμένο τμήμα του δικτύου (segment) όπως vlan[9] ή τομέα, για την επικύρωση των διακινούμενων κρυπτογραφημένων πρωτοκόλλων.

Η τεχνική του sniffing ενός δικτύου έχει και κάποιους περιορισμούς. Ένας από τους περιορισμούς που πιθανά να έχουμε, είναι η χρήση της κρυπτογράφησης. Πολλοί επιτιθέμενοι εκμεταλλεύονται τεχνικές κρυπτογράφησης για να κρύψουν τις δραστηριότητες τους. Στην περίπτωση αυτή, ενώ οι αξιολογητές μπορούν να δουν την ύπαρξη επικοινωνίας, δεν είναι σε θέση να δουν το περιεχόμενο αυτής.

- **Έλεγχος ακεραιότητας αρχείων.** Ο έλεγχος αυτός, μας παρέχει έναν τρόπο για να διαπιστώσουμε εάν τα αρχεία ενός συστήματος έχουν

[8] Port span: Σε περίπτωση που στο δίκτυο μας κάνουμε χρήση μεταγωγέων (switches), για να πραγματοποιήσουμε sniffing, πρέπει να ενεργοποιήσουμε την υπηρεσία “SPAN” στην θύρα (Ethernet port) που θα βάλουμε την συσκευή sniffing. Με την χρήση της υπηρεσίας αυτής, προωθείται η δικτυακή κίνηση όλου του δικτύου (ή των επιλεγμένων θυρών) στην θύρα του μεταγωγέα που έχουμε τοποθετήσει την sniffing συσκευή. Σε περίπτωση χρήσης hub, δεν χρειάζεται η συγκεκριμένη διαδικασία διότι η κίνηση μεταδίδεται σε όλες τις θύρες λόγω αρχιτεκτονικής.

[9] VLAN (Virtual LAN): Μια ομάδα συσκευών οι οποίες βρίσκονται σε ένα ή περισσότερα φυσικά LAN (broadcast domain) και μπορούν να επικοινωνούν μεταξύ τους σαν να βρίσκονται στο ίδιο broadcast domain.

τροποποιηθεί, υπολογίζοντας και αποθηκεύοντας το άθροισμα ελέγχου (checksum) κάθε επιτηρούμενου αρχείου και δημιουργώντας παράλληλα μια βάση δεδομένων για αυτά. Τα αποθηκευμένα αθροίσματα ελέγχου, επανυπολογίζονται, προκειμένου να συγκριθεί η τιμή που έχουν την δεδομένη στιγμή της σύγκρισης με την αποθηκευμένη τιμή. Οποιαδήποτε μεταβολή στην τιμή αυτή, υποδεικνύει ότι το αρχείο έχει τροποποιηθεί. Οι ελεγκτές ακεραιότητας των αρχείων συνήθως συμπεριλαμβάνονται στα συστήματα IDS, αλλά διατίθενται και ως αυτόνομα εργαλεία.

Τα εργαλεία ελέγχου της ακεραιότητας των αρχείων δεν απαιτούν υψηλό βαθμό ανθρώπινης αλληλεπίδρασης, όμως, για να διασφαλιστεί η αποτελεσματικότητά τους, θα πρέπει να χρησιμοποιούνται με προσοχή. Για να είναι πιο αποτελεσματικός ο έλεγχος, τα αρχεία του συστήματος συγκρίνονται με μια βάση δεδομένων αναφοράς, η οποία όταν δημιουργήθηκε, είχε χρησιμοποιήσει το σύστημα αρχείων σε μία στιγμή όπου είμαστε βέβαιοι ότι το σύστημα είναι ασφαλές. Με αυτόν τον τρόπο διασφαλίζουμε ότι η βάση δεδομένων αναφοράς δεν δημιουργήθηκε με αλλοιωμένα αρχεία. Η βάση αυτή, θα πρέπει να διατηρείται σε τοποθεσία όπου δεν υπάρχει πρόσβαση μέσω δικτύου, ούτως ώστε να μην είναι εφικτό σε επίδοξους εισβολείς, να θέσουν σε κίνδυνο το σύστημα με πιθανή τροποποίηση της. Επίσης, επειδή οι ενημερώσεις των αρχείων τους συστήματος (patches) τροποποιούν τα ελεγχόμενα αρχεία, τα αθροίσματα ελέγχου (checksum) της βάσης θα πρέπει να ενημερώνονται και αυτά.

3.2.2.2. Τεχνικές αναγνώρισης και ανάλυσης στόχων.

Με τις τεχνικές αναγνώρισης στόχου οι ελεγκτές ενός συστήματος μπορούν να εντοπίσουν ενεργά συστήματα, θύρες, υπηρεσίες, καθώς και πιθανές αδυναμίες του ελεγχόμενου συστήματος. Οι ελεγκτές χρησιμοποιούν τις πληροφορίες που αποκομίζουν από την αναγνώριση, για περεταίρω έρευνα η οποία πιθανά να επικυρώσει την ύπαρξη τρωτών σημείων. Ο συνηθέστερος τρόπος δοκιμής είναι μέσω αυτοματοποιημένων εργαλείων, αλλά φυσικά, έλεγχος μπορεί να γίνει και με χειροκίνητο τρόπο. Μερικοί από τους τρόπους που έχουμε για την αναγνώριση στόχου περιγράφονται παρακάτω.

- **Χαρτογράφηση δικτύου (Network Discovery).** Η χαρτογράφηση δικτύου είναι μια τεχνική που χρησιμοποιείται για να την ανακάλυψη των ενεργών και αποκρινόμενων συσκευών ενός δικτύου. Υπάρχουν δύο είδη χαρτογράφησης: η **παθητική** (passive) εξέταση και η **ενεργητική** (active) εξέταση.

  Στην **παθητική εξέταση** ο ελεγκτής χρησιμοποιεί έναν καταγραφέα πακέτων (network sniffer) για την παρακολούθηση της κίνησης του δικτύου και την καταγραφή των διευθύνσεων IP των ενεργών συσκευών που μετέχουν σε αυτό. Επίσης, ο καταγραφέας μπορεί να συνάγει το λειτουργικό σύστημα των συσκευών, τις ανοικτές θύρες, το είδους της κίνησης και άλλες λεπτομέρειες. Για τη συγκέντρωση των

απαραίτητων πληροφοριών, η εξέταση αυτή είναι πιο χρονοβόρα σε σχέση με την ενεργητική, ενώ επίσης δεν μπορεί να εντοπίσει συσκευές που δεν μεταδίδουν ή δεν δέχονται δικτυακή κίνηση κατά τη διάρκεια της εξέτασης. Η παθητική εξέταση λειτουργεί καλύτερα όταν η καταγραφή των πακέτων γίνεται εσωτερικά στο δίκτυο του φορέα.

Στην **ενεργητική εξέταση**, ο ελεγκτής, μέσω αυτοματοποιημένων εργαλείων, στέλνει διάφορα είδη πακέτων όπως ICMP[10] περιμένοντας απόκριση από τους ενεργούς υπολογιστές του δικτύου. Έτσι, μπορεί να αναγνωρίσει τα λειτουργικά συστήματα των ενεργών συσκευών ή με την αποστολή κατάλληλων πακέτων σε συγκεκριμένες θύρες και τις απαντήσεις που θα λάβει πίσω, μπορεί να καταγράψει τις υφιστάμενες ενεργές θύρες καθώς και την κατάσταση αυτών των θυρών, όπως για παράδειγμα αν είναι ανοιχτές ή κλειστές. Αυτές οι πληροφορίες που συλλέχθηκαν, μπορούν να χρησιμοποιηθούν για να γίνει μια χαρτογράφηση της τοπολογίας του δικτύου προκειμένου να πραγματοποιηθεί αργότερα μια προγραμματισμένη δοκιμή διείσδυσης για την ανακάλυψη των πιθανών τρωτών σημείων.

- **Αναγνώριση θυρών και υπηρεσιών δικτύου (Network Port and Service Identification):** Η τεχνική αυτή περιλαμβάνει τη χρήση ενός σαρωτή θυρών για να εντοπισθούν οι θύρες που είναι ανοικτές στο δίκτυο, οι υπηρεσίες που εκτελούνται στις ενεργές συσκευές, καθώς και οι εφαρμογές που λειτουργούν στα πλαίσια κάθε ταυτοποιημένης υπηρεσία, όπως π.χ. Microsoft IIS ή Apache για την υπηρεσία HTTP.

  Μερικοί από τους διαθέσιμους σαρωτές θυρών είναι σε θέση να παράσχουν πρόσθετες πληροφορίες σχετικά με τις ενεργές συσκευές που σαρώνουν. Αυτό γίνεται μέσα από πληροφορίες που συγκεντρώνονται κατά τη διάρκεια της σάρωσης. Οι πληροφορίες αυτές μπορούν να βοηθήσουν στον εντοπισμό του λειτουργικού συστήματος του στόχου, μια διαδικασία που ονομάζεται *OS fingerprinting*. Για παράδειγμα, εάν ένας υπολογιστής έχει ανοικτές τις θύρες TCP 135, 139, 445, τότε είναι πιθανώς μια συσκευή Windows, ή ενδεχομένως, μια συσκευή Unix που τρέχει Samba[11]. Βέβαια η μέθοδος αυτή έχει ένα σεβαστό ποσοστό σφάλματος, διότι υπάρχουν firewalls τα οποία μπλοκάρουν συγκεκριμένες θύρες και συγκεκριμένο είδος δικτυακής κίνησης (network traffic) και οι διαχειριστές των δικτύων μπορούν να διαμορφώσουν τα συστήματά τους με τέτοιο τρόπο, ώστε να συγκαλύπτουν το πραγματικό λειτουργικό σύστημα (OS).

[10] Internet Control Message Protocol

[11] Samba: Λογισμικό ανοιχτού κώδικα που προσφέρει εξυπηρέτηση αρχείων και εκτυπώσεων καθώς και ενσωμάτωση τομέων εξυπηρετών Windows σε εξυπηρέτες που λειτουργούν κάτω από UNIX ή UNIX-like (Linux, Solaris) λειτουργικό.

Ο κάθε σαρωτής υποστηρίζει διάφορες μεθόδους σάρωσης, οι οποίες έχουν αδυναμίες και πλεονεκτήματα. Υπάρχουν σαρωτές που αποδίδουν καλύτερα στην σάρωση διά μέσω firewalls, ενώ άλλοι είναι καταλληλότεροι για τις σαρώσεις στο εσωτερικό του δικτύου, όταν δηλαδή δεν παρεμβάλλεται τείχος προστασίας. Όπως είναι φυσικό, τα αποτελέσματα διαφέρουν ανάλογα με τον χρησιμοποιούμενο σαρωτή. Για αυτούς τους λόγους, ο ελεγκτής είναι αυτός που κάθε φορά επιλέγει το κατάλληλο εργαλείο, που θα τον βοηθήσει να αποκτήσει τις πιο χρήσιμες πληροφορίες για αυτόν στο υπό διερεύνηση δίκτυο.

Η προτεινόμενη μεθοδολογία για την ταυτοποίηση υπηρεσιών και θυρών είναι η εκτέλεση δυο διαφορετικών σαρώσεων, μίας που θα διενεργηθεί εσωτερικά της περιμέτρου ασφαλείας του δικτύου και μιας που θα διενεργηθεί εξωτερικά, με την εξωτερική σάρωση να διεξάγεται πρώτη σε σειρά, προκειμένου τα αρχεία καταγραφής να συγκριθούν με αυτά της εσωτερικής σάρωσης. Κατά την εκτέλεση των σαρώσεων στην εξωτερική περίμετρο, οι ελεγκτές μπορούν να χρησιμοποιήσουν οποιαδήποτε από τις υπάρχουσες τεχνικές απόκρυψης (stealth techniques) για να αποφύγουν την τυχόν ανίχνευση τους από τα συστήματα IDS και IPS. Επίσης μπορούν να χρησιμοποιήσουν τεχνικές κατακερματισμού (fragmentation), δημιουργίας διπλοτύπων (duplication), επικάλυψης (overlap), αναδιάταξης ακολουθίας (out-of-order), καθώς και τεχνικές που αλλάζουν τον χρονισμό των πακέτων, έτσι ώστε να φαίνονται ως κανονική κυκλοφορία και να διεισδύουν ευκολότερα. Ο εσωτερικός έλεγχος, χρησιμοποιεί λιγότερο επιθετικές μεθόδους σάρωσης, διότι αυτές οι σαρώσεις αναχαιτίζονται δυσκολότερα σε σχέση με τις εξωτερικές. Όμως, επειδή οι σαρωτές φτιάχνουν προσαρμοσμένα πακέτα για να πραγματοποιήσουν τη διαδικασία αυτή και τα πακέτα αυτά, μπορούν να έχουν ως παράπλευρη συνέπεια μια επίθεση άρνησης υπηρεσίας (Denial of Service (DoS) attack), αυτό το είδος του ελέγχου θα πρέπει να διεξάγεται κατά τη διάρκεια περιόδων χαμηλής κίνησης του δικτύου, όπως τις νυχτερινές ώρες ή το Σαββατοκύριακο.

- **Σάρωση εντοπισμού ευπαθειών (Vulnerability Scanning):** Η σάρωση εντοπισμού ευπαθειών, όπως και η σάρωση ταυτοποίησης θυρών και υπηρεσιών, προσδιορίζει τα χαρακτηριστικά των ενεργών συσκευών ενός δικτύου, όπως το λειτουργικό σύστημα, τις εφαρμογές που εκτελούνται, τις ανοιχτές θύρες που υπάρχουν σε αυτό κ.λπ. και επιχειρεί να εντοπίσει τα τρωτά σημεία που πιθανά υπάρχουν. Πολλοί σαρωτές εντοπισμού ευπαθειών, διαθέτουν τις κατάλληλες επιλογές για να δεχθούν έτοιμα αποτελέσματα, από ελέγχους που αναφέρθηκαν νωρίτερα, όπως την ανίχνευση δικτύου και την ταυτοποίηση θυρών και υπηρεσιών. Με αυτόν τον τρόπο μειώνεται η ποσότητα της εργασίας που απαιτείται για την ολοκλήρωση του συνόλου των εργασιών.

Οι έλεγχοι αυτοί μπορούν να βοηθήσουν στον εντοπισμό παρωχημένων εκδόσεων λογισμικού, παραλειπόμενων επιδιορθώσεων (patches) και απόκλιση ή μη συμμόρφωση από την πολιτική ασφαλείας του οργανισμού. Αυτό γίνεται με τον προσδιορισμό των λειτουργικών συστημάτων, τον προσδιορισμό των εκδόσεων των υπηρεσιών που εκτελούνται και την εύρεση πληροφοριών για γνωστές ευπάθειες των συγκεκριμένων εκδόσεων.

Οι σαρωτές εντοπισμού ευπαθειών μπορούν να:

- Ελέγξουν τη συμμόρφωση των ελεγχόμενων συσκευών με τις πολιτικές εκτέλεσης εφαρμογών και ασφάλειας
- Παράσχουν πληροφορίες για τους στόχους που πρόκειται να ελεγχθούν κατά τον έλεγχο διείσδυσης
- Παράσχουν πληροφορίες σχετικά με την αντιμετώπιση των ευπαθειών που ανακαλύφθηκαν

Χρησιμοποιώντας έναν σαρωτή ευπαθειών έχουμε έναν σχετικά γρήγορο και εύκολο τρόπο για να ανακαλύψουμε τις αδυναμίες που έχει ο υπό έλεγχο οργανισμός. Όμως, μια από τις υφιστάμενες δυσκολίες που έχουν να αντιμετωπίσουν οι ελεγκτές, σε ό,τι αφορά τον προσδιορισμό του επιπέδου του κινδύνου των ανακαλυπτόμενων τρωτών σημείων, είναι το γεγονός ότι τα τρωτά σημεία σπάνια είναι μεμονωμένα – συνηθέστατα υπάρχουν συνδυασμοί τρωτών σημείων. Για παράδειγμα, υπάρχει πιθανότητα να ανακαλύψουμε πολλές ευπάθειες χαμηλού κινδύνου, που όμως, δύναται να παρουσιάζουν υψηλό κίνδυνο, όταν αυτές συνδυάζονται. Τα εργαλεία αυτά δεν είναι σε θέση να κάνουν τέτοιου είδους συνδυασμούς που πιθανά να είναι και μέρος σχεδίου μιας επίθεσης. Ο πιο αξιόπιστος τρόπος για τον εντοπισμό και την αξιολόγηση των κινδύνων των ευπαθειών στο σύνολό τους, είναι μέσω δοκιμών διείσδυσης.

Η σάρωση εντοπισμού ευπαθειών είναι μια εργασία που απαιτεί υψηλό βαθμό ανθρώπινης παρέμβασης για την ερμηνεία των αποτελεσμάτων. Μπορεί επίσης να αποδιοργανώσει τις λειτουργίες του δικτύου και να τις επιβραδύνει, λόγω των αυξημένων αναγκών σε εύρος ζώνης (bandwidth) που απαιτεί. Παρόλα αυτά, η σάρωση εντοπισμού ευπαθειών είναι εξαιρετικά σημαντική γιατί με αυτόν τον τρόπο διορθώνουμε τα τρωτά σημεία και μετριάζουμε τις συνέπειες, πριν ανακαλυφθούν και αξιοποιηθούν από κακόβουλους χρήστες.

3.2.2.3. Τεχνικές επικύρωσης ευπαθειών του στόχου

Με την πραγματοποίηση αυτών των τεχνικών δοκιμών, οι ελεγκτές χρησιμοποιούν τις πληροφορίες που παράγονται από τον προσδιορισμό και την ανάλυση των στόχων, με σκοπό την εξακρίβωση και περαιτέρω διερεύνηση των πιθανών τρωτών σημείων. Οι έλεγχοι μπορούν να εκτελούνται χειροκίνητα ή με τη χρήση αυτόματων εργαλείων, ανάλογα με την τεχνική που χρησιμοποιείται και την ικανότητα της ομάδας δοκιμής. Οι τεχνικές επικύρωσης των ευπαθειών των στόχων,

περιλαμβάνουν αποκάλυψη συνθηματικών, ελέγχους διείσδυσης, κοινωνική μηχανική και δοκιμές ασφαλείας.

- **Αποκάλυψη συνθηματικών (Password Cracking).** Όταν ένας χρήστης πληκτρολογεί ένα συνθηματικό για να εισέλθει σε ένα σύστημα, παράγεται μία μορφή κερματισμού (”hash”) του εισαχθέντος κωδικού, και αυτή συγκρίνεται με τον αποθηκευμένο κερματισμό του πραγματικού συνθηματικού που βρίσκεται στο σύστημα. Αν ταιριάζουν οι δύο τιμές κερματισμού, ο χρήστης έχει πιστοποιηθεί. Η αποκάλυψη συνθηματικών είναι η διαδικασία ανάκτησης των συνθηματικών, από τις τιμές κερματισμού τους, που βρίσκονται αποθηκευμένες σε ένα σύστημα ή μεταδίδονται μέσα από το δίκτυο. Αυτό γίνεται όταν ο επιτιθέμενος, μέσω sniffer, υποκλέπτει την τιμή κερματισμού του κωδικού κατά τη μετάδοση του, ή την ανακτά από το σύστημα-στόχο, κάτι το οποίο κανονικά απαιτεί δικαιώματα διαχειριστή (administrator/root) ή/και φυσική πρόσβαση. Από τη στιγμή που ο επιτιθέμενος έχει λάβει τις τιμές κερματισμού που χρειάζεται, ένα αυτοματοποιημένο εργαλείο για αποκάλυψη συνθηματικών, παράγει και δοκιμάζει συνθηματικά, μέχρι να βρεθεί η απαιτούμενη αντιστοιχία στις τιμές κερματισμού και να σταματήσει η διαδικασία.

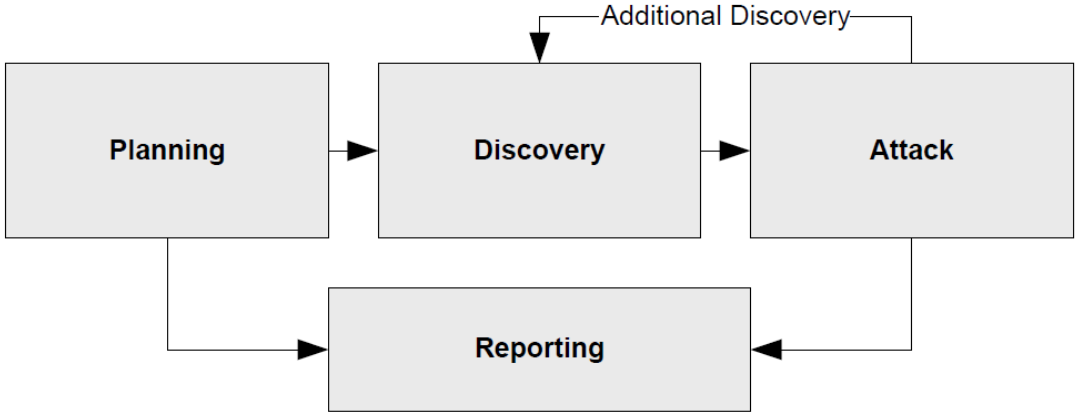
  Δυο είναι οι βασικές μέθοδοι για την τεχνική αυτή. Η μέθοδος που χρησιμοποιεί λεξικό για την επίθεση (**dictionary attack**), η οποία χρησιμοποιεί λέξεις από ένα αρχείο κειμένου και η υβριδική μέθοδος (**Hybrid attack**) η οποία βασίζεται στη μέθοδο του λεξικού με τη διαφορά ότι προσθέτει και αλφαριθμητικούς χαρακτήρες. Επίσης, μια άλλη μέθοδος είναι η μέθοδος της «ωμής βίας» (**brute force**). Η μέθοδος αυτή δημιουργεί πιθανά συνθηματικά μέχρι ένα ορισμένο μήκος συμβολοσειράς μέχρι να εντοπισθεί η κατάλληλη τιμή κερματισμού. Είναι φυσικό, βάσει των πιθανοτήτων, να χρειαστούν μήνες, μέχρι να πραγματοποιηθεί η αποκάλυψη (cracking) ενός κωδικού. Έτσι, οι αξιολογητές και οι επιτιθέμενοι χρησιμοποιούν παραπάνω από έναν υπολογιστές για το έργο τους με σκοπό να μειώσουν τον απαιτούμενο χρόνο.

  Τα εργαλεία σπασίματος κωδικών μπορούν να χρησιμοποιηθούν κατά τη διάρκεια μιας αξιολόγησης, με σκοπό να διασφαλίσουν τη συμμόρφωση των χρησιμοποιούμενων κωδικών με την πολιτική ασφάλειας του οργανισμού. Για παράδειγμα, εάν ο οργανισμός έχει μια πολιτική για την λήξη των κωδικών πρόσβασης, τα εργαλεία αυτά μπορούν να χρησιμοποιηθούν σε κατάλληλα χρονικά διαστήματα, για να ελέγξουν αν τηρούνται οι προβλεπόμενοι χρόνοι.
- **Έλεγχος διείσδυσης (Penetration Testing),** είναι δοκιμές ασφαλείας στο πλαίσιο των οποίων οι αξιολογητές μιμούνται πραγματικούς επιτιθέμενους, με σκοπό να εξακριβώσουν τις μεθόδους παράκαμψης ασφάλειας, μιας εφαρμογής, ενός συστήματος ή ενός δικτύου και να αποτιμήσουν την ασφάλεια των πληροφοριών. Περιλαμβάνει πραγματικές επιθέσεις, με εργαλεία και τεχνικές που χρησιμοποιούνται

συνήθως από επιτιθέμενους, σε συστήματα και δεδομένα. Οι περισσότερες δοκιμές διείσδυσης αναζητούν συνδυασμούς ευπαθειών στα ελεγχόμενα συστήματα, με σκοπό να αποκτήσουν μεγαλύτερο έλεγχο, σε σχέση με τον έλεγχο που θα αποκτούσαν κατά την εύρεση μιας μεμονωμένης ευπάθειας. Με τους ελέγχους αυτούς μπορούμε να προσδιορίσουμε για ένα σύστημα:

- Την αντοχή του σε πραγματικά σχέδια επίθεσης,
- Το επίπεδο πολυπλοκότητας της εισβολής,
- Τα πιθανά επιπρόσθετα αντίμετρα, που θα μπορούσαν να μετριάσουν τις ευπάθειες του συστήματος,
- Την ικανότητα του συστήματος να εντοπίσει την επίθεσης και να ανταποκριθεί κατάλληλα.

Οι δοκιμές ελέγχου διείσδυσης είναι μεγάλης σημασίας και απαιτούν από τους ελεγκτές μεγάλη εμπειρία για να ελαχιστοποιηθεί ο κίνδυνος των συστημάτων που εξετάζονται. Ο NIST εξειδικεύει τη μεθοδολογία εκτέλεσης των δοκιμών αυτών σε τέσσερεις φάσεις: (1) του **σχεδιασμού**, (2) της **αναγνώρισης**, (3) της **επίθεσης** και (4) της **αναφοράς**.



Εικόνα 3: Μεθοδολογία ελέγχου διείσδυσης τεσσάρων φάσεων κατά NIST

Στη φάση του σχεδιασμού των δοκιμών διείσδυσης, αποφασίζονται οι κανόνες καθώς και η διαδικασία των ελέγχων, εγκρίνονται από την διοίκηση του οργανισμού και τίθενται οι στόχοι των ελέγχων αυτών.

Η φάση της αναγνώρισης περιλαμβάνει δυο μέρη: στο πρώτο μέρος, που είναι και η αρχή της πραγματικής δοκιμής, συγκεντρώνονται οι απαραίτητες πληροφορίες του συστήματος στο οποίο θέλουμε να επιτεθούμε. Συγκεκριμένα, σαρώνεται το δίκτυο για να αναγνωριστούν οι υπηρεσίες που εκτελούνται καθώς και οι ανοιχτές θύρες, στις συσκευές στόχους. Οι πληροφορίες ονόματος και IP διεύθυνσης των στόχων μπορούν να συγκεντρωθούν μέσω πολλών μεθόδων, όπως από την υπηρεσία DNS, από ερωτήματα WHOIS, από παρακολούθηση δικτύου (sniffing) κ.ά. Ονόματα υπαλλήλων και στοιχεία επικοινωνίας μπορούν να συγκεντρωθούν με αναζήτηση στους εξυπηρέτες ιστού του οργανισμού, ή στους εξυπηρέτες καταλόγου. Πληροφορίες του

συστήματος, όπως ονόματα και κοινόχρηστοι φάκελοι μπορούν να βρεθούν μέσω της απαρίθμησης πόρων NetBIOS. Και τέλος, πληροφορίες εφαρμογών και υπηρεσιών, όπως εκδόσεις λογισμικών, μπορούν να καταγραφούν με μεθόδους όπως η αρπαγή πακέτων πληροφορίας (banner grabbing).

Το δεύτερο μέρος της φάσης διερεύνησης, περιλαμβάνει την σύγκριση των υπηρεσιών, των εφαρμογών και των λειτουργικών συστημάτων των στόχων, με βάσεις δεδομένων οι οποίες περιέχουν δημοσιευμένες ευπάθειες, για να αποκαλύψουν οι αξιολογητές τα τρωτά τους σημεία.

Η φάση της επίθεσης είναι και το επίκεντρο κάθε δοκιμής διείσδυσης. Σε αυτή τη φάση, διαφοροποιούνται οι δοκιμές ελέγχου διείσδυσης, από την αξιολόγηση των ευπαθειών του συστήματος επαληθεύοντας τις αδυναμίες του. Ο ελεγκτής, εφόσον εντοπίζει τις αδυναμίες του συστήματος, αποκτά πρόσβαση σε αυτό και προσπαθεί να αυξήσει τα προνόμια του για να εξερευνήσει περαιτέρω το σύστημα, ώστε να εγκαταστήσει τα κατάλληλα εργαλεία που θα τον βοηθήσουν να επαναλάβει την φάση της επίθεσης και να αποκτήσει πρόσβαση και σε άλλα συστήματα και πόρους του δικτύου.

Οι περισσότερες ευπάθειες που ανακαλύπτονται από με τους ελέγχους, μπορούν να ταξινομηθούν ως εξής:

- Λανθασμένες ρυθμίσεις ασφαλείας
- Ελαττώματα στον πυρήνα.
- Υπερχειλίσεις ενδιάμεσης μνήμης.
- Ανεπαρκής επικύρωση δεδομένων εισόδου.
- Συμβολικοί σύνδεσμοι (links).
- Επιθέσεις με χρήση περιγραφέων αρχείων.
- Συνθήκες ανταγωνισμού.
- Ακατάλληλα δικαιώματα αρχείων και καταλόγων.

Η φάση αναφοράς, γίνεται ταυτόχρονα με τις άλλες τρεις φάσεις του ελέγχου διείσδυσης. Στη φάση του σχεδιασμού, καταγράφεται και τεκμηριώνεται το πλάνο της αποτίμησης. Στην φάση της διερεύνησης καθώς και σε αυτό της επίθεσης, αρχειοθετούνται τα αρχεία καταγραφής και μεταφέρονται στους διαχειριστές του συστήματος. Μετά την ολοκλήρωση της δοκιμής, συντάσσεται η τελική αναφορά (το παραδοτέο) στο οποίο περιγράφονται οι ευπάθειες που εντοπίστηκαν, η επικινδυνότητα αυτών και οι συστάσεις προς τους διαχειριστές για τους τρόπους μετριασμού του κινδύνου.

Το συμπέρασμα που μπορούμε να εξάγουμε από όλα όσα αναφέρθηκαν παραπάνω είναι, ότι καμία τεχνική δεν μπορεί να προσφέρει μια ολοκληρωμένη εικόνα της ασφάλειας ενός συστήματος ή ενός δικτύου. Οι οργανισμοί θα πρέπει να συνδυάζουν τις κατάλληλες τεχνικές, προκειμένου να πραγματοποιηθεί μια καλή εκτίμηση ασφαλείας.

4. Web application security Consortium

Το Web Application Security Consortium (WASC)[12] είναι ένας μη κερδοσκοπικός οργανισμός που απαρτίζεται από μία διεθνή ομάδα ειδικών, ειδικών της βιομηχανίας και εκπροσώπων μεγάλων οργανισμών που παράγουν πρότυπα ασφάλειας βέλτιστων πρακτικών για τον ιστό, τα οποία είναι ευρέως αναγνωρισμένα και διατίθενται με τους όρους του ελεύθερου λογισμικού.

Η κατάταξη των ευπαθειών όπως αναφέρονται στο Web Application Security είναι μια συνεργατική προσπάθεια διευκρίνισης και οργάνωσης των πιθανών κινδύνων που απειλούν ένα ιστόχωρο. Τα μέλη του Web Application Security Consortium έχουν δημιουργήσει αυτό το πρόγραμμα για να προωθήσουν στην βιομηχανία της πληροφορικής, πρότυπη ορολογία για την περιγραφή αυτών των θεμάτων. Οι προγραμματιστές εφαρμογών ιστού, οι επαγγελματίες της ασφάλειας, οι προμηθευτές λογισμικού, έχουν τη δυνατότητα να αποκτήσουν πρόσβαση σε πληθώρα πληροφοριών, κατηγοριοποιημένα, για ζητήματα ασφάλειας των διαδικτυακών εφαρμογών.

4.1. Χρήση της Κατηγοριοποίησης Απειλών

Η δεύτερη έκδοση του Web Application Security v.2.0 (WASC) περιγράφει τις **επιθέσεις (attacks)** που δύναται να δεχθεί και τις **αδυναμίες (weaknesses)** που μπορεί να έχει μια εφαρμογή ή μια ιστοσελίδα, στα δεδομένα που περιέχει ή στους χρήστες της. Το έγγραφο αυτό χρησιμεύει κυρίως ως οδηγός αναφοράς, για κάθε δεδομένη επίθεση ή αδυναμία και παρέχει παραδείγματα για κάθε θέμα, καθώς και χρήσιμο υλικό αναφοράς. Η κατηγοριοποίηση χρησιμοποιείται από πολλούς οργανισμούς με τους ακόλουθους τρόπους.

Η κατηγοριοποίηση απειλών του WASC δημιουργήθηκε και αξιολογήθηκε από εμπειρογνώμονες του κλάδου με χρόνια εμπειρίας. Η κύρια χρήση του είναι να αποτελέσει οδηγό αναφοράς που μπορεί να περιλαμβάνεται σε εκθέσεις ασφαλείας, ελαττώματα ασφαλείας, σε παρουσιάσεις κ.ά. Εάν εργαζόμαστε ως ελεγκτές ενός συστήματος ασφαλείας μπορούμε να χρησιμοποιήσουμε την κατηγοριοποίηση απειλών του WASC για να απαριθμήσουμε και να κατηγοριοποιήσουμε τις πιθανές απειλές που μπορεί να αντιμετωπίσουμε, προκειμένου να δημιουργήσουμε ένα σχέδιο ελέγχου (test plan). Με αυτόν τον τρόπο μπορούμε να εντοπίσουμε τις πιθανές ελλείψεις ασφαλείας που πιθανώς να έχουμε και εφόσον γνωρίζουμε τις αδυναμίες μας μπορούμε να προβούμε σε αναβάθμιση της ασφάλειας του συστήματος μας ή της εφαρμογής μας.

Η δεύτερη έκδοση της κατηγοριοποίησης απειλών του WASC δημοσιεύτηκε σε δυο εκδοχές. Την εκδοχή που αδυναμίες και οι επιθέσεις προβάλλονται απαριθμημένα και την εκδοχή όπου προβάλλονται ανά φάση ανάπτυξης της εφαρμογής.

[12] HTTP://projects.webappsec.org/w/page/13246978/Threat%20Classification

4.2. Κατηγοριοποίηση απειλών, απαριθμημένη εκδοχή

Η εκδοχή αυτή απαριθμεί τις επιθέσεις, και τις αδυναμίες που δύναται να δεχθεί ή να έχει μια εφαρμογή ή μια ιστοσελίδα στα δεδομένα που περιέχει ή στους χρήστες της. Αυτή είναι και η βασική εκδοχή της κατηγοριοποίησης των απειλών, κατά WASC. Ο Πίνακας **1** παρουσιάζει επιγραμματικά τις πιο αντιπροσωπευτικές επιθέσεις και απειλές.

Επίθεση	**Ευπάθεια**
Κατάχρηση λειτουργικότητας (Abuse of Functionality)	Κακή διαμόρφωση εφαρμογής (Application Misconfiguration)
Ωμή βία (Brute Force)	Εμφάνιση λίστας περιεχομένων καταλόγων (Directory Indexing)
Υπερχείλιση ενδιάμεσης μνήμης (Buffer Overflow)	Ακατάλληλα δικαιώματα στο σύστημα αρχείων (Improper Filesystem Permissions)
Πλαστογράφηση περιεχομένου (Content Spoofing)	Ακατάλληλος χειρισμός εισόδου (Improper Input Handling)
Πρόβλεψη διαπιστευτηρίων/συνόδου (Credential/Session Prediction)	Ακατάλληλος χειρισμός εξόδου (Improper Output Handling)
Cross-Site Scripting	Διαρροή πληροφορίας (Information Leakage)
Πλαστογράφηση αιτήσεων μεταξύ ιστοχώρων (Cross-Site Request Forgery)	Μη ασφαλής ευρετηριασμός (Insecure Indexing)
Άρνηση παροχής υπηρεσίας (Denial of Service)	Ανεπαρκή μέτρα έναντι αυτοματοποιημένης εκτέλεσης (Insufficient Anti-automation)
Λήψη αποτυπωμάτων (Fingerprinting)	Ανεπαρκής αυθεντικοποίηση (Insufficient Authentication)
Συμβολοσειρά διαμόρφωσης (Format String)	Ανεπαρκής εξουσιοδότηση (Insufficient Authorization)
«Λαθραίες» απαντήσεις HTTP (HTTP Response Smuggling)	Ανεπαρκής διαδικασία ανάκτησης συνθηματικών (Insufficient Password Recovery)
Διάσπαση απαντήσεων HTTP (HTTP Response Splitting)	Ανεπαρκής επικύρωση διαδικασιών (Insufficient Process Validation)
«Λαθραία» αιτήματα HTTP (HTTP Request Smuggling)	Ανεπαρκής αυτόματη λήξη συνόδων (Insufficient Session Expiration)
Διάσπαση αιτήσεων HTTP (HTTP Request Splitting)	Ανεπαρκής προστασία επιπέδου μεταφοράς (Insufficient Transport Layer Protection)
Υπερχείλιση ακεραίων (Integer Overflows)	Επισφαλής διαμόρφωση εξυπηρέτη (Server Misconfiguration)

Επίθεση	Ευπάθεια
Έγχυση στον LDAP (LDAP Injection)	
Έγχυση στο σύστημα ηλεκτρονικής αλληλογραφίας (Mail Command Injection)	
Έγχυση μηδενικών bytes (Null Byte Injection)	
Εκτέλεση εντολών λειτουργικού συστήματος (OS Commanding)	
Διάσχιση διαδρομής (Path Traversal)	
Προβλέψιμη τοποθεσία πόρων (Predictable Resource Location)	
Συμπερίληψη απομακρυσμένου αρχείου (Remote File Inclusion - RFI)	
Παράκαμψη δρομολόγησης (Routing Detour)	
Κακόβουλος ορισμός αναγνωριστικών συνόδου (Session Fixation)	
Εκμετάλλευση πινάκων σε μηνύματα SOAP (SOAP Array Abuse)	
Έγχυση σε αρχεία συμπερίληψης εξυπηρέτη (SSI Injection)	
Έγχυση σε εντολές SQL (SQL Injection)	
Κατάχρηση ανακατεύθυνσης URL (URL Redirector Abuse)	
Έγχυση σε XPath (XPath Injection)	
Καταιγισμός γνωρισμάτων XML (XML Attribute Blowup)	
Εξωτερικές οντότητες XML (XML External Entities)	
Ανάπτυξη οντοτήτων XML (XML Entity Expansion)	
Έγχυση XML (XML Injection)	
Έγχυση XQuery (XQuery Injection)	

Πίνακας 1. αντιπροσωπευτικές επιθέσεις και απειλές, βάσει του WASC

4.3. Κατηγοριοποίηση απειλών, εκδοχή ανά φάση ανάπτυξης της εφαρμογής

Αυτού του είδους η περιγραφή δημιουργήθηκε έτσι ώστε να υποδείξει σε ποιο σημείο του κύκλου της εφαρμογής υπάρχει πιθανότητα να εμφανιστεί ένας συγκεκριμένος τύπος ευπάθειας. Αυτή η περιγραφή δημιουργήθηκε σε μια προσπάθεια να προσδιοριστούν ενδεχόμενα για εισαγωγή ευπάθειας που

λαμβάνουν χώρα στην ίδια φάση, και δεν προσπαθεί να περιγράψει ακατάλληλες ενημερώσεις λογισμικού ή να αριθμήσει ακραίες περιπτώσεις. Επίσης, κάνει χρήση πάρα πολλών συσχετισμών.

Οι φάσεις που διακρίνονται είναι οι ακόλουθες:

Σχεδιασμός: Καλύπτει ευπάθειες οι οποίες είναι πιθανό να εμφανιστούν εξαιτίας κενών σημείων στον σχεδιασμό του λογισμικού ή εξαιτίας ακατάλληλου ή και λάθους σχεδιασμού.

Υλοποίηση: καλύπτει ευπάθειες εξαιτίας ακατάλληλων επιλογών στην υλοποίηση του λογισμικού.

Ανάπτυξη: καλύπτει ευπάθειες οι οποίες είναι πιθανό να εμφανιστούν εξαιτίας ακατάλληλων διαδικασιών ανάπτυξης ή κακών εφαρμογών ή ακόμα και κακής παραμετροποίησης στον εξυπηρέτη.

Επίθεση	Σχεδιασμός	Υλοποίηση	Λειτουργία
Κατάχρηση λειτουργικότητας (Abuse of Functionality)	√		
Κακή διαμόρφωση εφαρμογής (Application Misconfiguration)		√	√
Ωμή βία (Brute Force)	√	√	
Υπερχείλιση ενδιάμεσης μνήμης (Buffer Overflow)		√	
Πλαστογράφηση περιεχομένου (Content Spoofing)		√	
Πρόβλεψη διαπιστευτηρίων/συνόδου (Credential/Session Prediction)		√	
Cross-Site Scripting		√	
Πλαστογράφηση αιτήσεων μεταξύ ιστοχώρων (Cross-Site Request Forgery)	√	√	
Άρνηση παροχής υπηρεσίας (Denial of Service)	√	√	
Εμφάνιση λίστας περιεχομένων καταλόγων (Directory Indexing)			√
Συμβολοσειρά διαμόρφωσης (Format String)		√	
«Λαθραίες» απαντήσεις HTTP (HTTP Response Smuggling)		√	
Διάσπαση απαντήσεων HTTP (HTTP Response Splitting)		√	
«Λαθραία» αιτήματα HTTP (HTTP Request Smuggling)		√	

Επίθεση	Σχεδιασμός	Υλοποίηση	Λειτουργία
Διάσπαση αιτήσεων HTTP (HTTP Request Splitting)		√	
Υπερχείλιση ακεραίων (Integer Overflows		√	
Ακατάλληλα δικαιώματα στο σύστημα αρχείων (Improper Filesystem Permissions)		√	√
Ακατάλληλος χειρισμός εισόδου (Improper Input Handling)		√	
Ακατάλληλος χειρισμός εξόδου (Improper Output Handling)		√	
Διαρροή πληροφορίας (Information Leakage)	√	√	√
Μη ασφαλής ευρετηριασμός (Insecure Indexing)		√	√
Ανεπαρκή μέτρα έναντι αυτοματοποιημένης εκτέλεσης (Insufficient Anti-automation)	√	√	
Ανεπαρκής αυθεντικοποίηση (Insufficient Authentication)	√	√	
Ανεπαρκής εξουσιοδότηση (Insufficient Authorization)	√	√	
Ανεπαρκής διαδικασία ανάκτησης συνθηματικών (Insufficient Password Recovery)	√	√	
Ανεπαρκής επικύρωση διαδικασιών (Insufficient Process Validation)	√	√	
Ανεπαρκής αυτόματη λήξη συνόδων (Insufficient Session Expiration)	√	√	√
Ανεπαρκής προστασία επιπέδου μεταφοράς (Insufficient Transport Layer Protection)	√	√	√
Έγχυση στον LDAP (LDAP Injection)		√	
Έγχυση στο σύστημα ηλεκτρονικής αλληλογραφίας (Mail Command Injection)		√	
Έγχυση μηδενικών bytes (Null Byte Injection)		√	
Εκτέλεση εντολών λειτουργικού συστήματος (OS Commanding)		√	
Διάσχιση διαδρομής (Path Traversal)		√	

Επίθεση	Σχεδιασμός	Υλοποίηση	Λειτουργία
Προβλέψιμη τοποθεσία πόρων (Predictable Resource Location)		√	√
Συμπερίληψη απομακρυσμένου αρχείου (Remote File Inclusion - RFI)		√	√
Παράκαμψη δρομολόγησης (Routing Detour)			√
Επισφαλής διαμόρφωση εξυπηρέτη (Server Misconfiguration)			√
Κακόβουλος ορισμός αναγνωριστικών συνόδου (Session Fixation)		√	√
Συμπερίληψη απομακρυσμένου αρχείου (Remote File Inclusion - RFI)		√	
Παράκαμψη δρομολόγησης (Routing Detour)		√	
Έγχυση σε εντολές SQL (SQL Injection)		√	
Κατάχρηση ανακατεύθυνσης URL (URL Redirector Abuse)	√	√	
Έγχυση σε XPath (XPath Injection)		√	
Καταιγισμός γνωρισμάτων XML (XML Attribute Blowup)		√	
Εξωτερικές οντότητες XML (XML External Entities)		√	
Ανάπτυξη οντοτήτων XML (XML Entity Expansion)		√	
Έγχυση XML (XML Injection)		√	
Έγχυση XQuery (XQuery Injection)		√	

Πίνακας 2. Αντιπροσωπευτικές επιθέσεις και απειλές, βάσει του WASC, ταξινομημένες κατά στάδιο κύκλου ζωής

4.4. Επιθέσεις (attacks)

4.4.1. Κατάχρηση της λειτουργικότητας (Abuse of Functionality)

Είναι μια τεχνική επίθεσης η οποία χρησιμοποιεί τα χαρακτηριστικά και τις λειτουργίες μιας ιστοσελίδας για να επιτεθεί είτε στην ίδια την ιστοσελίδα, είτε σε κάποια άλλη. Η τεχνική αυτή μπορεί να κάνει κατάχρηση κάποιων ιδιαίτερων χαρακτηριστικών μιας εφαρμογής με σκοπό να επιτύχει ένα επιθυμητό αποτέλεσμα από πλευράς επιτιθέμενου. Αυτές οι επιθέσεις έχουν ποικίλα αποτελέσματα όπως την κατανάλωση πόρων, παράκαμψη ελέγχου εισόδου ή διαρροή πληροφοριών. Η δυναμική και το επίπεδο της κατάχρησης αυτής ποικίλλει από σελίδα σε σελίδα και από εφαρμογή σε εφαρμογή. Οι επιθέσεις κατάχρησης λειτουργικότητας, γίνονται

συχνά σε συνδυασμό με επιθέσεις άλλου είδους ή άλλων συνιστωσών επίθεσης. Μερικά παραδείγματα που μπορούμε να παραθέσουμε είναι:

- Κατάχρηση λειτουργιών των εφαρμογών αποστολής e-mails
- Κατάχρηση διαδικασιών ανάκτησης κωδικών πρόσβασης
- Κατάχρηση λειτουργικότητας μέσω δημιουργίας υπερβολικού μεγέθους αιτημάτων αντιπροσώπευσης (proxy)

Οι εφαρμογές ιστού που έχουν την δυνατότητα να στέλνουν e-mails, όπως το "send-mail" πρέπει να δίνουν προσοχή ούτως ώστε να μην αφήνουν στον χρήστη πλήρη έλεγχο μέσω μηνυμάτων κεφαλίδας και περιεχομένου. Εάν ο εισβολέας καταφέρει να ελέγξει τα πεδία του μηνύματος (από, προς, θέμα και κυρίως σώμα) και δεν υπάρχουν αυτόματοι μηχανισμοί ελέγχου, τότε, οι λειτουργίες e-mail μπορούν να μετατραπούν σε μηχανισμούς μεταφοράς ανεπιθύμητης αλληλογραφίας.

Ανάλογες εφαρμογές που βασίζονται σε PERL, όπως το "FormMail" χρησιμοποιούνταν κανονικά για την μεταφορά δεδομένων χρηστών σε προκαθορισμένες διευθύνσεις e-mail. Και αυτές οι υψηλού επιπέδου λειτουργίες, έγιναν αντικείμενο κατάχρησης από απομακρυσμένους κακόβουλους χρήστες.

Οι μηχανισμοί ανάκτησης κωδικών πολλές φορές μπορούν να χρησιμοποιηθούν κακόβουλα για να γίνει διαρροή δεδομένων σχετικά με λογαριασμούς, οι οποίοι κανονικά θα έπρεπε να είναι ασφαλείς.

Τέλος υπάρχει η πιθανότητα, μερικές υπηρεσίες, όπως η μετάφραση της Google, να μπορεί να χρησιμοποιηθεί καταχρηστικά έτσι ώστε να λειτουργεί ως ένας ανοιχτός εξυπηρέτης αντιπροσώπευσης (proxy server).

4.4.2. Επιθέσεις ωμής βίας (Brute Force)

Είναι η μέθοδος επίθεσης η οποία υπολογίζει μια άγνωστη τιμή, χρησιμοποιώντας μια αυτόματη μέθοδο δοκιμής, μέσα από ένα πλήθος πιθανών επιλογών. Η επίθεση αυτή πλεονεκτεί από το γεγονός ότι, οι άγνωστες αυτές τιμές δεν είναι εύκολα αντιληπτές. Για παράδειγμα ενώ ένας κωδικός 8 αλφαριθμητικών χαρακτήρων μπορεί να έχει 2,8 τρισεκατομμύρια πιθανές τιμές, πολλοί άνθρωποι θα επιλέξουν κωδικούς από ένα πολύ μικρότερο υποσύνολο το οποίο αποτελείται από κοινές λέξεις και εκφράσεις.

Ο πιο κοινός τύπος τέτοιου είδους επίθεσης σε εφαρμογή ιστού είναι η επίθεση σε διαπιστευτήρια εισόδου. Λόγω του ότι οι χρήστες, θέλουν να θυμούνται τους κωδικούς τους, συνήθως επιλέγουν εύκολους κωδικούς, κάνοντας την επίθεση με χρήση λεξικού αρκετά εύκολη. Ο επιτιθέμενος που προσπαθεί μια τέτοια επίθεση επιδιώκει να εισέλθει στο σύστημα χρησιμοποιώντας μια μεγάλη λίστα από λέξεις ή φράσεις σαν πιθανούς κωδικούς. Οι κωδικοί μπορεί να περιλαμβάνουν μεταβολές στις λέξεις όπως για παράδειγμα την αντικατάσταση του «ο» με «0» κ.λπ.

Παράδειγμα τέτοιας επίθεσης είναι η περίπτωση που μια λειτουργία υπενθύμισης κωδικού ρωτάει το χρήστη μια πληροφορία η οποία είναι συνήθως γνωστή μόνο σε αυτόν. Παρ' όλα αυτά εάν η ερώτηση είναι του τύπου: «ποιο είναι το αγαπημένο σας χρώμα» τότε ο εισβολέας μπορεί εύκολα να βρει τον κωδικό, διότι ο αριθμός των χρωμάτων είναι περιορισμένος. Επίσης, από μελέτες έχει

βρεθεί πως οι περισσότεροι χρήστες σε αυτή την ερώτηση επιλέγουν το «μπλε» σαν αγαπημένο χρώμα, κάτι που κάνει την επίθεση από έναν εισβολέα αρκετά πιο εύκολη και προβλέψιμη.

Άλλες παρόμοιες επιθέσεις ωμής βίας είναι η επίθεση στα αναγνωριστικά εισόδου καθώς και η επίθεση υποκλοπής πληροφοριών πιστωτικών καρτών.

4.4.3. Υπερχείλιση ενδιάμεσης μνήμης (Buffer Overflow)

Η υπερχείλιση ενδιάμεσης μνήμης είναι ένα είδος επίθεσης που συμβαίνει όταν σε μία περιοχή μνήμης καταγράφονται περισσότερα δεδομένα από ότι η περιοχή μνήμης μπορεί να διαχειριστεί. Η τεχνική εκμετάλλευσης της υπερχείλισης ενδιάμεσης μνήμης επιτρέπει στον εισβολέα να τροποποιήσει τα σημεία που θέλει από τις διευθύνσεις της μνήμης που στοχεύει. Αυτή η ικανότητα μπορεί να χρησιμοποιηθεί για ένα πλήθος από σκοπούς όπως:

- Έλεγχο της εκτέλεσης της επεξεργασίας
- Κατάρρευση της επεξεργασίας
- Τροποποίηση των εσωτερικών μεταβλητών

Ο σκοπός του επιτιθέμενου είναι να μπορεί να ελέγχει τις διευθύνσεις μνήμης που στοχεύει. Οι πιο συχνές επιθέσεις αυτού του τύπου σχετίζονται με προγράμματα σχεδιασμένα με γλώσσα C και C++, λόγω του εύρους χρήσης τους και την ικανότητα τους να εκτελούν εντολές κατευθείαν στην μνήμη με κοινές προγραμματιστικές δομές.

4.4.4. Πλαστογράφηση περιεχομένου (Content Spoofing)

Είναι μια τεχνική επίθεσης, η οποία επιτρέπει στον εισβολέα να εγχύσει κακόβουλα πακέτα δεδομένων τα οποία αργότερα μπορεί να εκληφθούν σαν νόμιμα δεδομένα της εφαρμογής. Διακρίνεται σε δυο κύριες κατηγορίες:

- Πλαστογράφηση περιεχομένου που περιέχει μόνο κείμενο. Μία κοινή προσέγγιση για τη δημιουργία δυναμικών σελίδων, είναι να διαμορφώνεται το κύριο σώμα ή τμήματα του κειμένου μέσα στη σελίδα, αντιγράφοντας απλώς τιμές που έχει παραθέσει ο χρήστης στο αίτημά του. Αυτή η προσέγγιση είναι κοινή σε σελίδες σφάλματος (error pages) ή σε σελίδες που παρέχουν καταχωρήσεις ειδήσεων. Το περιεχόμενο που προσδιορίζεται με αυτή την παράμετρο αργότερα αντανακλάται μέσα στη σελίδα προκειμένου να παράσχει το περιεχόμενο αυτής.
- Πλαστογράφηση περιεχομένου των HTML markups. Αρκετές ιστοσελίδες λειτουργούν με πηγές περιεχομένου που ενημερώνονται δυναμικά. Ο εισβολέας μπορεί να αντικαταστήσει στον κώδικα, την τιμή της παραμέτρου η οποία καθορίζει την πηγή, π.χ. για ένα πλαίσιο (frame) ή μια εικόνα (img). Σε αντίθεση με την περίπτωση της ανακατεύθυνσης, όπου η ιστοσελίδα που προκύπτει φαίνεται ότι είναι παραποιημένη μέσω της γραμμής διευθύνσεων, εδώ, η ιστοσελίδα που προκύπτει φαίνεται πως έχει νόμιμο περιεχόμενο διότι είναι δυναμικά ανανεώσιμη και η πηγή του στοιχείου που έχει τροποποιηθεί φαίνεται μόνο στον κώδικα της σελίδας. Έτσι οι χρήστες της ιστοσελίδας εμπιστεύονται έμμεσα, πλαστό περιεχόμενο. Αυτή η τεχνική εκμεταλλεύεται την σχέση εμπιστοσύνης μεταξύ χρήστη και ιστοσελίδας και

χρησιμοποιείται για την δημιουργία ψεύτικων ιστοσελίδων και διασπορά ψευδών ειδήσεων.

4.4.5. Πρόβλεψη διαπιστευτηρίων / συνόδου - (Credential / Session Prediction)

Η επίθεση αυτή είναι μια μέθοδος πλαστοπροσωπίας του χρήστη ενός ιστότοπου. Εξάγοντας ή μαντεύοντας τη μοναδική τιμή η οποία προσδιορίζει μια συγκεκριμένη συνεδρία ή έναν χρήστη, ολοκληρώνεται η επίθεση. Η επίθεση αυτή είναι επίσης γνωστή και ως πειρατεία συνεδρίας (session highjacking).

Πολλές σελίδες είναι σχεδιασμένες για να αναγνωρίζουν τον χρήστη από το σημείο που επιτυγχάνεται η πρώτη συνεδρία και εφεξής. Για να γίνει αυτό, ο χρήστης αρχικά πρέπει να δώσει την ταυτότητά του στη σελίδα, συνήθως δίνοντας ένα όνομα χρήστη και ένα κωδικό. Έτσι, αντί να δίδονται αυτά τα δεδομένα πιστοποίησης σε κάθε προσπάθεια εισόδου στην σελίδα ή την εφαρμογή, η σελίδα δημιουργεί μια μοναδική ταυτότητα συνεδρίας “session ID” προκειμένου να αναγνωρίζει την συνεδρία του χρήστη ως αυθεντική. Το επακόλουθο αυτού του μηχανισμού είναι, η επικοινωνία ανάμεσα στο χρήστη και τη σελίδα, να σημειώνεται ως “έμπιστη”. Έτσι αν ένας κακόβουλος χρήστης, καταφέρει να προβλέψει ή να μαντέψει την ταυτότητα συνεδρίας ενός άλλου χρήστη, τότε είναι δυνατό να πιστοποιηθεί με τα στοιχεία του άλλου χρήση έτσι ώστε να κάνει πλαστοπροσωπία.

Πολλές σελίδες προσπαθούν να δημιουργήσουν “sessions id’s” χρησιμοποιώντας κατάλληλους αλγόριθμους. Αυτές οι μεθοδολογίες δημιουργίας ID συνεδρίας απλά προσαυξάνουν στατικούς αριθμούς ή μπορεί να ακολουθούν πιο σύνθετες διαδικασίες όπως ο συνυπολογισμός του χρόνου και άλλων εξειδικευμένων μεταβλητών. Στην συνέχεια το ID συνεδρίας αποθηκεύεται σε ένα cookie, σε μια κρυμμένη φόρμα ή σε ένα URL. Εάν ο εισβολέας μπορέσει να υπολογίσει τον αλγόριθμο που χρησιμοποιήθηκε για να δημιουργηθεί το ID συνεδρίας τότε η επίθεση που προετοιμάζει μπορεί να προχωρήσει όπως φαίνεται παρακάτω:

- Ο εισβολέας συνδέεται στην εφαρμογή ιστού αποκτώντας την συγκεκριμένη ID συνεδρίας.
- Ο εισβολέας υπολογίζει ή προχωρά σε επίθεση ωμής βίας (Brute Force) στην επόμενη συνεδρία ID.
- Ο εισβολέας αλλάζει την τιμή που έχει δοθεί στο cookie ή στα κρυμμένα πεδία φόρμας/URL και υποθέτει την ταυτότητα του επόμενου χρήστη.

4.4.6. Cross-Site Scripting

Το Cross-site Scripting (XSS) είναι μια τεχνική επίθεσης στην οποία ο επιτιθέμενος στέλνει τον δικό του τροποποιημένο κώδικα στην εφαρμογή πλοήγησης ιστού τους θύματος, προκειμένου να εκτελεστεί ως νόμιμος. Ένα παράδειγμα εφαρμογής του cross-site scripting μπορεί να είναι μία αυτόνομη εφαρμογή πλοήγησης (web browser), ή μία εφαρμογή πλοήγησης η οποία είναι ενσωματωμένη σε μια εφαρμογή όπως το Winamp ή ένας αναγνώστης RSS ή ένα πρόγραμμα διαχείρισης ηλεκτρονικής αλληλογραφίας (email client). Ο κώδικας

συνήθως είναι γραμμένος σε HTML/JAVAScript αλλά μπορεί να είναι γραμμένος και σε VBSScript, ActiveX, Java, Flash, ή οποιαδήποτε άλλη τεχνολογία υποστηρίζεται από μία εφαρμογή πλοήγησης ιστού.

Όταν ο επιτιθέμενος βρίσκει μία εφαρμογή πλοήγησης ιστού για να εκτελέσει τον κώδικά του, ο κώδικας αυτός θα τρέξει σύμφωνα με τους κανόνες ασφαλείας της εφαρμογής στην οποία φιλοξενείται. Σε αυτό το προνομιακό επίπεδο ο κώδικας έχει την ικανότητα να διαβάσει, να τροποποιήσει, ή ακόμα και να μεταδώσει ευαίσθητα δεδομένα στα οποία έχει πρόσβαση μέσα από την εφαρμογή πλοήγησης ιστού. Ένας χρήστης μπορεί να πέσει θύμα πειρατείας με “Cross-site Scripting” και η εφαρμογή πλοήγησης που χρησιμοποιεί μπορεί να τον μεταφέρει σε ιστοχώρους που έχει επιλέξει ο επιτιθέμενος χωρίς αυτό να γίνει αντιληπτό. Οι εισβολείς που κάνουν χρήση αυτής της τεχνικής στην ουσία εκθέτουν τη σχέση εμπιστοσύνης μεταξύ του χρήστη και του ιστότοπου.

4.4.7. Πλαστογράφηση αιτήσεων μεταξύ ιστοχώρων (Cross-Site Request Forgery)

Η πλαστογράφηση αιτήσεων μεταξύ ιστοχώρων (CSRF) είναι ένα είδος επίθεσης η οποία παροτρύνει το θύμα να στείλει ένα HTTP αίτημα σε ένα προορισμό στόχο, εν αγνοία του, με σκοπό να εκτελέσει μια ενέργεια. Η φύση της CSRF επίθεσης είναι ότι εκμεταλλεύεται την εμπιστοσύνη που έχει εγκαθιδρυθεί μεταξύ της ιστοσελίδας-στόχου και ενός χρήστη (πιθανότατα μετά από κάποια διαδικασία σύνδεσης), σε αντιδιαστολή με την επίθεση τύπου XSS, η οποία εκμεταλλεύεται την εμπιστοσύνη που έχει ο χρήστης για την ιστοσελίδα. Οι επιθέσεις τύπου CSRF είναι αποτελεσματικές σε ένα εύρος από καταστάσεις όπως:

- Το θύμα έχει μια ενεργή συνεδρία στην ιστοσελίδα-στόχο.
- Το θύμα είναι πιστοποιημένο μέσω HTTP πιστοποίησης στην ιστοσελίδα-στόχο.
- Το θύμα είναι στο ίδιο τοπικό δίκτυο με την ιστοσελίδα στόχο.

Το CSRF αρχικά χρησιμοποιήθηκε για να εκτελεί συγκεκριμένες ενέργειες σε ιστοσελίδες, χρησιμοποιώντας τα προνόμια του θύματος, αλλά με πιο πρόσφατες τεχνικές χρησιμοποιείται και για την αποκάλυψη πληροφοριών. Το ρίσκο της αποκάλυψης πληροφοριών αυξάνεται δραματικά όταν η ιστοσελίδα στόχος είναι ευάλωτη σε XSS, επειδή το XSS μπορεί να χρησιμοποιείται σαν πλατφόρμα για το CSRF, επιτρέποντας στην επίθεση να λειτουργήσει μέσα στα πλαίσια της ίδιας πολιτικής.

4.4.8. Άρνηση Υπηρεσίας (Denial of Service)

Η επίθεση (DoS) Είναι ένα είδος επίθεσης που έχει σκοπό να εμποδίζει μια εφαρμογή ιστού από το να εξυπηρετεί τη φυσιολογική δραστηριότητα του χρήστη. Αυτές οι επιθέσεις οι οποίες εφαρμόζονται εύκολα σε επίπεδο δικτύου είναι επίσης πιθανές και σε επίπεδο εφαρμογής. Μπορούν να επιτύχουν την κατασπατάληση των πόρων του συστήματος, εκμεταλλευόμενα την ευπάθεια του, ή κάνοντας κατάχρηση της λειτουργικότητάς του. Πολλές φορές οι επιθέσεις DoS προσπαθούν να καταναλώσουν όλους τους διαθέσιμους πόρους μιας εφαρμογής web ή μιας ιστοσελίδας όπως: τον επεξεργαστή, τη μνήμη τον χώρο στο δίσκο κ.λπ. Όταν

οποιοσδήποτε από αυτούς τους σημαντικούς πόρους εξαντληθεί, η εφαρμογή θα καταστεί μη προσβάσιμη.

Λόγω του ότι τα σύγχρονα περιβάλλοντα εφαρμογών ιστού περιλαμβάνουν έναν εξυπηρέτη ιστού (web server), έναν εξυπηρέτη βάσης δεδομένων, και έναν εξυπηρέτη αυθεντικοποίησης (authentication server), η επίθεση αυτή σε επίπεδο εφαρμογής μπορεί να στοχεύσει σε κάθε ένα από τα επιμέρους συστήματα. Σε αντιδιαστολή με την επίθεση DoS σε επίπεδο δικτύου, όπου ένας μεγάλος αριθμός από προσπάθειες σύνδεσης είναι απαραίτητος, η επίθεση αυτή σε επίπεδο εφαρμογής είναι μια πολύ πιο απλή διαδικασία.

4.4.9. Λήψη αποτυπωμάτων (Fingerprinting)

Η πιο γνωστή μέθοδος για τους εισβολείς σε ένα υπολογιστικό σύστημα είναι η μέθοδος όπου αρχικά γίνεται συλλογή όσο το δυνατό περισσότερων πληροφοριών για αυτό. Με τις πληροφορίες που θα συλλέξει ο εισβολέας, μπορεί να αναπτύξει ένα ακριβές σενάριο επίθεσης, με το οποίο θα εκμεταλλευτεί αποτελεσματικά τις αδυναμίες στο είδος ή στην έκδοση του λογισμικού το οποίο χρησιμοποιείται από τη σελίδα στόχο. Η μέθοδος αυτή ονομάζεται *λήψη αποτυπωμάτων* (fingerprinting).

Η μέθοδος της λήψης αποτυπωμάτων πολλαπλών επιπέδων είναι παρόμοια με την μέθοδο της λήψης αποτυπωμάτων TCP/IP, (π.χ. με σαρωτή nmap) με την διαφορά ότι είναι εστιασμένη στο επίπεδο εφαρμογής (Application Layer 7) του μοντέλου OSI αντί για το επίπεδο μεταφοράς (Transport Layer 4). Η θεωρία στην οποία βασίζεται η μέθοδος αυτή είναι η δημιουργία του ακριβούς προφίλ της υποδομής του στόχου, όπως για παράδειγμα η έκδοση του λογισμικού του εξυπηρέτη ιστού (web server), η έκδοση της βάσης δεδομένων (DB Server), της διαμόρφωσης αυτών ακόμα και αναγνώριση, αν μπορεί να καταστεί δυνατόν, της αρχιτεκτονικής του δικτύου.

Μερικές από τις μεθοδολογίες που ακολουθούνται σε αυτή τη μέθοδο συνοψίζονται παρακάτω.

- Προσδιορισμός της αρχιτεκτονικής/τοπολογίας ιστού
- Προσδιορισμός της έκδοσης του εξυπηρέτη ιστού
- Προσδιορισμός του λογισμικού της εφαρμογής του εξυπηρέτη ιστού
- Προσδιορισμός του λογισμικού της βάσης δεδομένων
- Προσδιορισμός των “services” των υπηρεσιών ιστού

4.4.10. Συμβολοσειρά μορφοποίησης (Format String)

Οι επιθέσεις που στοχεύουν στη συμβολοσειρά μορφοποίησης αλλάζουν τη ροή μιας εφαρμογής, χρησιμοποιώντας τα χαρακτηριστικά των συμβολοσειρών μορφοποίησης βιβλιοθηκών για να αποκτήσουν πρόσβαση σε άλλους χώρους μέσα στη μνήμη. Οι ευπάθειες δημιουργούνται όταν τα δεδομένα που παρέχει ο χρήστης χρησιμοποιούνται απευθείας ως συμβολοσειρές μορφοποίησης σε ορισμένες συναρτήσεις C/C ++ (π.χ., fprintf, printf, sprintf, setproctitle, syslog, κ.λπ.). Εάν ένας επιτιθέμενος κατορθώσει να μεταβιβάσει μία συμβολοσειρά που περιλαμβάνει χαρακτήρες μετατροπής (π.χ. για την printf "% f", "%" p, "% n", κ.λπ.) ως τιμή παραμέτρου σε μία εφαρμογή ιστού, και αυτή η παράμετρος χρησιμοποιηθεί ως συμβολοσειρά μορφοποίησης, τότε μπορούν να συμβούν τα παρακάτω:

- Εκτέλεση αυθαίρετου κώδικα στον εξυπηρέτη.
- Ανάγνωση των τιμών στην στοίβα.
- Πρόκληση σφαλμάτων κατάτμησης και διακοπή εκτέλεσης του λογισμικού.

Οι επιθέσεις αυτές συνδέονται και με άλλα είδη επίθεσης , όπως η υπερχείλιση ενδιάμεσης μνήμης και η υπερχείλιση ακεραίων. Και οι τρείς επιθέσεις βασίζονται στην ικανότητά τους να διαχειρίζονται τη μνήμη ή το διερμηνευτή του συστήματος με τέτοιο τρόπο που να συμβάλλουν στο στόχο του εισβολέα.

Ένα παράδειγμα που μπορεί να δοθεί για την επίθεση που στοχεύει στη συμβολοσειρά μορφοποίησης είναι αν υποθέσουμε ότι μία web εφαρμογή έχει την παράμετρο *emailAddress* η οποία δίνεται από τον χρήστη. Η εφαρμογή τυπώνει την τιμή της μεταβλητής χρησιμοποιώντας την μέθοδο printf: *printf(emailAddress);*

Εάν η τιμή που απέστειλε προς την παράμετρο emailAddress περιέχει χαρακτήρες μετατροπής, η printf θα αναλύσει τους χαρακτήρες αυτούς και για κάθε χαρακτήρα μετατροπής θα αναζητήσει την επόμενη παράμετρο στη λίστα για να την τυπώσει κατάλληλα. Αν δεν υπάρχουν τέτοιες παράμετροι, τα δεδομένα από τη στοίβα, θα χρησιμοποιούνται σύμφωνα με την αναμενόμενη σειρά που αναμένει η συνάρτηση printf.

Οι πιθανές χρήσεις των επιθέσεων που στοχεύουν στις συμβολοσειρές μορφοποίησης σε μια τέτοια περίπτωση μπορεί να είναι ως εξής:

- **Ανάγνωση των δεδομένων από τη στοίβα**: Αν η έξοδος της συνάρτησης printf εμφανίζεται πίσω στον επιτιθέμενο, τότε μπορεί να διαβάσει τις τιμές στη στοίβα στέλνοντας τον χαρακτήρα μετατροπής "%x”, μία ή περισσότερες φορές.
- **Ανάγνωση συμβολοσειράς χαρακτήρα από τη διαδικασία της μνήμης:** Αν η έξοδος της συνάρτησης printf εμφανιστεί πίσω στον επιτιθέμενο, τότε μπορεί να διαβάσει τις συμβολοσειρές χαρακτήρα σε αυθαίρετες θέσεις μνήμης με τη χρήση του χαρακτήρα μετατροπής "%s" καθώς και με άλλους χαρακτήρες μετατροπής με σκοπό να βρεθεί η σωστή θέση μνήμης.
- **Εγγραφή μιας ακεραίας τιμής σε θέσεις κατά τη διαδικασία μνήμη:** Με τη χρήση του χαρακτήρα μετατροπής "%n", ένας επιτιθέμενος μπορεί να γράψει μια ακέραια τιμή σε οποιαδήποτε θέση στη μνήμη (π.χ., αντικατάσταση σημαντικών ενδείξεων του προγράμματος που ελέγχουν προνόμια πρόσβασης, αντικατάσταση επιστρεφόμενων διευθύνσεων στη στοίβα, κ.λπ.).

4.4.11. «Λαθραία» αιτήματα HTTP (HTTP Request Smuggling)

Τα «λαθραία» αιτήματα HTTP είναι μία μέθοδος επίθεσης που βασίζεται σε αποστολή αιτημάτων τα οποία δεν είναι πλήρως συμμορφούμενα με τις προδιαγραφές του πρωτοκόλλου HTTP (π.χ. χρήση πολλαπλών επικεφαλίδων Content-Length). Η επίθεση λειτουργεί σε περιβάλλοντα όπου υπάρχουν πολλαπλές οντότητες που διερμηνεύουν το ίδιο αίτημα (π.χ. ένας αντιπρόσωπος HTTP και ένας εξυπηρέτης) και επιχειρεί να αξιοποιήσει διαφορές που υπάρχουν μεταξύ των οντοτήτων αυτών στον τρόπο χειρισμού των μη συμμορφούμενων προς τις προδιαγραφές αιτημάτων. Ένα παράδειγμα επίπτωσης της επίθεσης είναι να φορτωθεί στην κρυφή μνήμη του αντιπροσώπου το περιεχόμενο της σελίδας

http://www.target.site/~attacker/foo.html με αντιστοίχηση στο URL http://www.target.site/~victim/bar.html, κάτι που θα έχει ως αποτέλεσμα αν κάποιος χρήστης ζητήσει από τον αντιπρόσωπο τη σελίδα http://www.target.site/~victim/bar.html να λάβει το περιεχόμενο της σελίδας http://www.target.site/~attacker/foo.html.

4.4.12. Διάσπαση αιτήσεων HTTP (HTTP Request Splitting)

Η διάσπαση αιτήσεων HTTP είναι ένας τύπος επίθεσης που εξαναγκάζει την εφαρμογή πλοήγησης να αποστείλει ελεγχόμενα από τον εισβολέα αιτήματα HTTP, τα οποία προκαλούν ευπάθειες XSS και παραποιούν την κρυφή μνήμη της εφαρμογής πλοήγησης. Το βασικό χαρακτηριστικό της επίθεσης αφορά στην ικανότητα του επιτιθέμενου, από τη στιγμή που το θύμα (εφαρμογή πλοήγησης) έχει εξαναγκαστεί να φορτώσει την κακόβουλη σελίδα του εισβολέα, να αποκτήσει πρόσβαση στις λειτουργικότητες της εφαρμογής πλοήγησης ώστε να αποστείλει δύο αιτήματα HTTP αντί για ένα. Μέχρι σήμερα έχουν γίνει αντικείμενο εκμετάλλευσης δύο μηχανισμοί: το αντικείμενο XmlHttpRequest object και ο μηχανισμός HTTP digest authentication. Προκειμένου να λειτουργήσει αυτός ο τύπος επίθεσης, η εφαρμογή πλοήγησης πρέπει να χρησιμοποιεί έναν αντιπρόσωπο προώθησης HTTP, ή εναλλακτικά η επίθεση πρέπει να γίνει προς έναν υπολογιστή που βρίσκεται στο ίδιο δίκτυο με τον επιτιθέμενο.

4.4.13. «Λαθραίες» απαντήσεις HTTP (HTTP Response Smuggling)

Οι επιθέσεις που βασίζονται στις «λαθραίες» απαντήσεις HTTP επιχειρούν να αποστείλουν 2 απαντήσεις HTTP από τον εξυπηρέτη στον εξυπηρετούμενο μέσω ενός ενδιαμέσου πράκτορα HTTP που αναμένει (ή επιτρέπει) μία μόνο απάντηση από τον εξυπηρέτη. Μία χρήση αυτής της τεχνικής είναι να επαυξηθεί η βασική τεχνική διάσπασης απαντήσεων, με σκοπό να παρακαμφθούν μέτρα έναντι της διάσπασης των απαντήσεων. Στην περίπτωση αυτή, ο ενδιάμεσος είναι αυτός που υλοποιεί τα μέτρα έναντι της διάσπασης των απαντήσεων. Μία άλλη χρήση είναι να πλαστογραφηθούν οι απαντήσεις που παραλαμβάνονται από την εφαρμογή πλοήγησης. Στην περίπτωση αυτή, ένας κακόβουλος ιστοχώρος αποστέλλει στην εφαρμογή πλοήγησης μία σελίδα, την οποία η εφαρμογή πλοήγησης θα διερμηνεύσει ως προερχόμενη από άλλον τομέα του δικτύου. Αυτό μπορεί να επιτευχθεί όταν η εφαρμογή πλοήγησης χρησιμοποιεί έναν εξυπηρέτη αντιπροσώπευσης για να προσπελάσει και τους δύο ιστοχώρους.

4.4.14. Διάσπαση απαντήσεων HTTP (HTTP Response Splitting)

Σε αυτή τη μέθοδο επίθεσης υπάρχουν τουλάχιστον τρία μέρη τα οποία εμπλέκονται.

- **Ο εξυπηρέτης ιστού** (web server) στον οποίο υπάρχει κενό ασφαλείας και επιτρέπει τη διάσπαση απαντήσεων HTTP.
- **Ο στόχος**, μια οντότητα η οποία αλληλεπιδρά με τον εξυπηρέτη ιστού, ίσως για λογαριασμό του εισβολέα. Ο στόχος συνήθως είναι ένας ενδιάμεσος εξυπηρέτης (cache forward/reverse proxy) ή μία εφαρμογή πλοήγησης (πιθανότατα με κρυφή μνήμη).
- **Ο εισβολέας** ο οποίος ξεκινάει την επίθεση.

Κύριο χαρακτηριστικό αυτής της επίθεσης είναι η ικανότητα του εισβολέα να στέλνει ένα απλό αίτημα HTTP το οποίο ωθεί τον εξυπηρέτη ιστού να δημιουργήσει μια ροή εξόδου (output stream), η οποία στην συνέχεια ερμηνεύεται από το σύστημα-στόχο σαν δύο αποκρίσεις HTTP αντί για μια απόκριση που θα έπρεπε κανονικά. Η πρώτη απόκριση μπορεί να ελέγχεται μερικά από τον εισβολέα αλλά αυτό δεν είναι και τόσο σημαντικό. Το σημαντικό είναι πως ο εισβολέας ελέγχει απολύτως το πλαίσιο της δεύτερης απόκρισης από τη γραμμή κατάστασης HTTP μέχρι και το τελευταίο byte αυτής της απόκρισης. Όταν αυτό συμβεί ο εισβολέας πραγματοποιεί την επίθεση του στέλνοντας δύο αιτήματα μέσω του στόχου. Το πρώτο αίτημα προκαλεί τις δύο απαντήσεις από τον εξυπηρέτη ιστού και το δεύτερο αίτημα μπορεί να είναι τυπικά ένας «αθώος» πόρος για τον εξυπηρέτη ιστού. Παρόλα αυτά το δεύτερο αίτημα μπορεί να ταιριάζεται από τον στόχο με τη δεύτερη HTTP απόκριση, η οποία ελέγχεται από τον εισβολέα. Ο εισβολέας έτσι ξεγελάει τον στόχο και τον οδηγεί να πιστέψει ότι ένας συγκεκριμένος πόρος στον εξυπηρέτη ιστού (ο οποίος ζητήθηκε με το δεύτερο αίτημα) αντιστοιχεί σε κάποια δεδομένα τα οποία έχουν δημιουργηθεί από τον εισβολέα μέσα από τον εξυπηρέτη ιστού (το δεύτερο τμήμα της πρώτης απόκρισης).

4.4.15. Υπερχείλιση ακεραίων (Integer Overflows)

Η επίθεση τύπου υπερχείλιση ακεραίων (integer overflow), είναι μια κατάσταση η οποία συμβαίνει όταν το αποτέλεσμα από μια αριθμητική λειτουργία όπως ο πολλαπλασιασμός ή η πρόσθεση, υπερβαίνει τη μέγιστη τιμή που μπορεί να δοθεί και να αποθηκευτεί. Όταν συμβεί αυτό, η ερμηνευμένη τιμή που δίδεται, φαίνεται να έχει πάρει τιμή μεγαλύτερη από τη μέγιστη τιμή που μπορεί να πάρει με αποτέλεσμα να αρχίσει πάλι από την ελάχιστη τιμή. Για παράδειγμα ένας 8bit ακέραιος αριθμός στην αρχιτεκτονική των περισσότερων υπολογιστών έχει μια μέγιστη τιμή 127 και μια ελάχιστη -128. Εάν ο προγραμματιστής αποθηκεύσει την τιμή 127 σε μια τέτοια μεταβλητή και προσθέσει σε αυτή την τιμή 1, το αποτέλεσμα θα πρέπει να είναι 128. Όμως, αυτή η τιμή υπερβαίνει το μέγιστο για αυτό τον τύπο ακεραίου, με αποτέλεσμα να αποδίδεται στην ερμηνευμένη τιμή, η τιμή -128.

4.4.16. Έγχυση στον LDAP (LDAP Injection)

Η έγχυση στον LDAP είναι μια τεχνική επίθεσης η οποία χρησιμοποιείται για να εκμεταλλευτεί ιστοσελίδες οι οποίες κατασκευάζουν εντολές LDAP από δεδομένα εισόδου που παρέχει κάποιος χρήστης.

Το Lightweight Directory Access Protocol (LDAP) είναι ένα ανοιχτό πρωτόκολλο τόσο για εκτέλεση ερωτημάτων όσο και για την επεξεργασία και διαχείριση των υπηρεσιών καταλόγου X.500. Το LDAP εκτελείται πάνω από τα πρωτόκολλα του επιπέδου μεταφοράς όπως το TCP. Οι εφαρμογές ιστού μπορούν να χρησιμοποιούν τα δεδομένα εισόδου των χρηστών για να δημιουργήσουν προσαρμοσμένες εντολές LDAP για αιτήματα σε δυναμικές ιστοσελίδες.

Όταν μια εφαρμογή ιστού αποτυγχάνει στο να ελέγξει πλήρως τα δεδομένα εισόδου των χρηστών, είναι δυνατό για τον εισβολέα να τροποποιήσει όπως αυτός θέλει, την κατασκευή μιας εντολής LDAP. Όταν ένας εισβολέας πραγματοποιήσει μια τέτοια ενέργεια, η διαδικασία θα πραγματοποιηθεί με τα ίδια δικαιώματα που έχει το στοιχείο που εκτέλεσε την εντολή (π.χ. εξυπηρέτης βάσης δεδομένων,

εξυπηρέτης ιστού κ.λπ.). Αυτό μπορεί να προκαλέσει διάφορα και σοβαρά προβλήματα ασφάλειας, αφού τα δικαιώματα που έχουν χορηγηθεί στο ερώτημα, μπορούν να τροποποιήσουν ή να αφαιρέσουν οτιδήποτε μέσα από τη βάση δεδομένων LDAP.

4.4.17. Έγχυση στο σύστημα ηλεκτρονικής αλληλογραφίας (Mail Command Injection)

Η έγχυση στο σύστημα ηλεκτρονικής αλληλογραφίας είναι μια τεχνική επίθεσης η οποία χρησιμοποιείται για να εκμεταλλευτεί εξυπηρέτες ηλεκτρονικού ταχυδρομείου και εφαρμογές webmail, οι οποίες κατασκευάζουν εντολές IMAP/SMTP από δεδομένα εισόδου χρηστών τα οποία δεν είναι κατάλληλα ελεγμένα. Ανάλογα με τον τύπο της εφαρμογής, στην περίπτωση που ο εισβολέας επιτυγχάνει στην επίθεσή του, συναντάμε δύο τύπους έγχυσης. Την **έγχυση IMAP** και την **έγχυση SMTP**. Μια έγχυση IMAP/SMTP μπορεί να καταστήσει δυνατή την πρόσβαση σε κάποιον εξυπηρέτη ηλεκτρονικού ταχυδρομείου στον οποίο προηγουμένως δεν υπήρχε πρόσβαση. Σε μερικές περιπτώσεις, στους εσωτερικούς εξυπηρέτες που απαρτίζουν ένα δίκτυο δεν εφαρμόζονται τα ίδια επίπεδα ασφαλείας όπως συνήθως γίνεται σε αντίστοιχους εξυπηρέτες που βρίσκονται σε ένα μετωπικό επίπεδο (front-end). Σε μια τέτοια περίπτωση οι εισβολείς μπορούν εύκολα να αντιληφθούν πως ο εξυπηρέτης ηλεκτρονικού ταχυδρομείου είναι ευάλωτος σε μια επίθεση. Από την άλλη πλευρά, αυτή η τεχνική επιτρέπει να αποφευχθούν πιθανοί περιορισμοί οι οποίοι μπορεί να υπάρξουν σε επίπεδο εφαρμογής όπως CAPTCHA, όρια σε πλήθος ή ρυθμό αιτημάτων κ.λπ.).

4.4.18. Έγχυση μηδενικών bytes (Null Byte Injection)

Η έγχυση μηδενικών bytes έγχυση είναι μια τεχνική επίθεσης ενεργής εκμετάλλευσης, η οποία χρησιμοποιείται για να παρακάμψει τα λογικά φίλτρα ελέγχου που βρίσκονται στις υποδομές των δικτύων υπολογιστών, προσθέτοντας URL-κωδικοποιημένους Null Byte χαρακτήρες, (π.χ. %00, ή 0x00) στα δεδομένα εισόδου των χρηστών. Αυτή η επεξεργασία έγχυσης μπορεί να εναλλάξει την εσωτερική λογική της εφαρμογής και να επιτρέψει σε κακόβουλους χρήστες να αποκτήσουν πρόσβαση σε νευραλγικά αρχεία του συστήματος που επιτέθηκαν.

Οι περισσότερες εφαρμογές ιστού, έχουν αναπτυχθεί χρησιμοποιώντας γλώσσες υψηλού επιπέδου όπως η PHP, ASP, Perl και Java. Παρόλα αυτά όμως, απαιτούν σε κάποιο βαθμό, επεξεργασία κώδικα υψηλού επιπέδου σε επίπεδο συστήματος και αυτή η επεξεργασία, συνήθως επιτυγχάνεται με τη χρήση συναρτήσεων C/C++. Οι διαφορετικές φύσεις αυτών των ανεξάρτητων τεχνολογιών έχουν οδηγήσει σε μια κλάση επίθεσης η οποία λέγεται έγχυση μηδενικών byte (Null Byte Injection) ή δηλητηρίαση μηδενικών byte (Null Byte Poisoning). Στην C/C++ ένα μηδενικό byte αναπαριστά το τερματικό σημείο μιας συμβολοσειράς ή τον χαρακτήρα οριοθέτησης ο οποίος σταματάει την επεξεργασία της συμβολοσειράς άμεσα. Τα bytes τα οποία ακολουθούν τον χαρακτήρα οριοθέτησης αγνοούνται. Εάν η σειρά χάσει τον μηδενικό χαρακτήρα το μήκος της καθίσταται άγνωστο: η συμβολοσειρά εκτείνεται μέχρι ο έλεγχος της μνήμης συναντήσει το επόμενο μηδενικό byte. Αυτή η διευθέτηση μπορεί να προκαλέσει ασυνήθιστη συμπεριφορά και να παρουσιάσει ευπάθειες μέσα στο σύστημα. Κατά τον ίδιο ή

παρόμοιο τρόπο, διάφορες υψηλού επιπέδου γλώσσες διαχειρίζονται τα null byte σαν να κρατούν θέση για το μήκος της συμβολοσειράς, καθώς αυτό δεν έχει συγκεκριμένο νόημα στο περιεχόμενό τους. Εξαιτίας αυτής διαφοράς στην ερμηνεία τους, τα μηδενικά bytes μπορούν εύκολα να εγχυθούν και να χειραγωγήσουν τη συμπεριφορά της εφαρμογής.

Τα URL περιορίζονται στο να περιέχουν ASCII χαρακτήρες οι οποίοι κυμαίνονται από το 0x20 στο 0X7E (δεκαεξαδικό) ή 32 έως 126 (δεκαδικό). Παρόλα αυτά η προαναφερθείσα περιοχή τιμών περιέχει διάφορους χαρακτήρες οι οποίοι δεν επιτρέπονται, επειδή έχουν ειδική σημασία στα πλαίσια του HTTP πρωτοκόλλου. Για αυτό το λόγο το σχήμα κωδικοποίησης του URL περιλαμβάνει ειδικούς χαρακτήρες οι οποίοι χρησιμοποιούν εκτεταμένη αναπαράσταση χαρακτήρων ASCII. Σχετικά με τα null bytes, αυτά κωδικοποιούνται σαν %00 στο δεκαεξαδικό σύστημα. Ο σκοπός της επίθεσης μηδενικού byte ξεκινάει, εκεί που οι εφαρμογές ιστού αλληλοεπιδρούν με ενεργές ρουτίνες της γλώσσας “C” και εξωτερικά APIs (Application Programming Interface) του λειτουργικού συστήματος. Έτσι επιτρέπεται στον εισβολέα να διαχειριστεί υπολογιστικούς πόρους διαβάζοντας ή γράφοντας αρχεία τα οποία βασίζονται στα δικαιώματα χρήστη της εφαρμογής.

4.4.19. Εκτέλεση εντολών λειτουργικού συστήματος (OS Commanding)

Η OS Commanding είναι μια τεχνική επίθεσης η οποία χρησιμοποιείται για μη εξουσιοδοτημένη εκτέλεση εντολών του λειτουργικού συστήματος. Είναι το αποτέλεσμα μιας μίξης έμπιστου κώδικα και μη έμπιστων δεδομένων. Αυτή η επίθεση μπορεί να συμβεί όταν μια εφαρμογή δέχεται μη έμπιστα δεδομένα με σκοπό να δημιουργήσει εντολές προς το λειτουργικό σύστημα με έναν μη ασφαλή τρόπο, ο οποίος προϋποθέτει ανεπαρκή έλεγχο δεδομένων και ακατάλληλες κλήσεις προς εξωτερικά προγράμματα. Έτσι οι εκτελέσιμες εντολές του εισβολέα εκτελούνται με τα ίδια δικαιώματα που εκτελούνται και οι εντολές όπως για παράδειγμα του εξυπηρέτη ιστού, ή του εξυπηρέτη βάσης δεδομένων. Καθώς οι εντολές εκτελούνται με αυξημένα δικαιώματα ο εισβολέας μπορεί να τις χρησιμοποιήσει για να αποκτήσει πρόσβαση ή να καταστρέψει τμήματα τα οποία υπό άλλες συνθήκες δεν θα μπορούσε.

4.4.20. Διάσχιση διαδρομής (Path Traversal)

Με την τεχνική της επίθεσης τύπου διάσχισης διαδρομής, επιτρέπει την πρόσβαση του εισβολέα σε αρχεία, καταλόγους και εντολές, τα οποία κανονικά βρίσκονται έξω από τον κατάλογο ρίζας (Root Directory) του ιστοχώρου. Ο επιτιθέμενος μπορεί να διαχειριστεί ένα URL, με τέτοιο τρόπο, έτσι ώστε ο ιστοχώρος να εκτελέσει ή να αποκαλύψει περιεχόμενα αυθαίρετων αρχείων οπουδήποτε μέσα στον εξυπηρέτη ιστού. Οποιαδήποτε συσκευή που είναι ενεργοποιημένη με διασύνδεση τύπου HTTP, είναι εκτεθειμένη σε τρωτότητες αυτού του είδους.

Οι περισσότεροι ιστοχώροι περιορίζουν την πρόσβαση των χρηστών σε συγκεκριμένο τμήμα του συστήματος αρχείων, το οποίο λέγεται περιοχή ιστοχώρου και ουσιαστικά περιλαμβάνει όλα τα αντικείμενα κάτω από έναν κατάλογο που καλείται «ρίζα εγγράφων ιστού» (web document root). Αυτοί οι κατάλογοι περιέχουν αρχεία για την εκτέλεση των λειτουργιών της εφαρμογής. Οι επιτιθέμενοι

που κάνουν χρήση της διάσχισης διαδρομής χρησιμοποιούν ειδικές ακολουθίες χαρακτήρων, όπως η «../», έτσι ώστε να μπορούν να επιτεθούν στο σύστημα των αρχείων του συστήματος.

4.4.21. Προβλέψιμη τοποθεσία πόρων (Predictable Resource Location)

Είναι μια τεχνική επίθεσης που χρησιμοποιείται για να αποκαλύψει το κρυμμένο περιεχόμενο μιας ιστοσελίδας, καθώς και την λειτουργικότητας της. Κάνοντας μελετημένες προβλέψεις και σε συνδυασμό με την μέθοδο επίθεσης ωμής βίας (Brute Force), ο εισβολέας μπορεί να προβλέψει ονόματα αρχείων και καταλόγων τα οποία δεν προορίζονται για κοινή θέα. Αυτό είναι το εύκολο κομμάτι της επίθεσης, διότι τα αρχεία και οι κατάλογοι, συχνά, έχουν κοινά ονόματα τα οποία βρίσκονται σε συγκεκριμένες πάντα θέσεις. Έτσι, μπορεί να εκτεθούν, προσωρινά αρχεία, αρχεία ασφαλείας, αρχεία καταγραφής, αρχεία διαμόρφωσης, ή αρχεία δειγμάτων. Αυτά τα αρχεία, είναι πιθανό να αποκαλύψουν ευαίσθητες πληροφορίες σχετικά με την ιστοσελίδα, τις εφαρμογές της, πληροφορίες για τη βάση δεδομένων της, κωδικούς, ονόματα συσκευών, διαδρομές αρχείων προς άλλες ευαίσθητες περιοχές κ.ά.

Όμως, με αυτόν τον τρόπο, δεν υποβοηθάται μόνο η αναγνώριση της δομής της ιστοσελίδας, η οποία μπορεί να οδηγήσει σε πρόσθετες ευπάθειες, αλλά μπορεί να αποκαλύψει στον εισβολέα πληροφορίες ευπάθειας που αφορούν το περιβάλλον ή και τους χρήστες της.

4.4.22. Συμπερίληψη απομακρυσμένου αρχείου (Remote File Inclusion - RFI)

Είναι μια τεχνική επίθεσης που χρησιμοποιείται για να εκμεταλλευτεί μηχανισμούς δυναμικής συμπερίληψης αρχείων σε εφαρμογών ιστού. Όταν μια εφαρμογή ιστού δέχεται δεδομένα τα οποία εισάγει ο χρήστης όπως URL, τιμές παραμέτρων κ.ά. και τα μεταβιβάζει σε εντολές συμπερίληψης αρχείων, τότε η εφαρμογή μπορεί να ξεγελαστεί και να συμπεριλάβει αρχεία που εμπεριέχουν κακόβουλο κώδικα.

Σχεδόν όλες οι εφαρμογές ιστού υποστηρίζουν συμπερίληψη αρχείου (file inclusion). Χρησιμοποιείται κυρίως για να συμπεριλάβει κοινό κώδικα μέσα σε ξεχωριστά αρχεία. Όταν μια εφαρμογή ιστού αναφέρει ότι περιλαμβάνει ένα τέτοιο αρχείο συμπερίληψης, τότε ο κώδικας σε αυτό, μπορεί να εκτελεστεί έμμεσα ή άμεσα καλώντας συγκεκριμένες διαδικασίες. Μια τέτοια επίθεση μπορεί να έχει σχετικά ανώδυνες επιπτώσεις όπως η γνωστοποίηση των περιεχομένων του αρχείου, αλλά μπορεί να οδηγήσει και σε εκτέλεση κώδικα στον εξυπηρέτη ή σε εκτέλεση κώδικα στον πελάτη, π.χ. με τη μορφή Javascript. Επίσης, μπορεί να οδηγήσει και σε άλλες επιθέσεις όπως cross-site scripting (xss), σε άρνηση υπηρεσίας (DoS) ακόμα και σε κλοπή και εκμετάλλευση δεδομένων.

4.4.23. Παράκαμψη Δρομολόγησης (Routing Detour)

Το πρωτόκολλο WS-Routing είναι ένα πρωτόκολλο που προορίζεται για ανταλλαγή μηνυμάτων SOAP (Simple Object Access Protocol) από ένα αρχικό αποστολέα μηνυμάτων σε έναν τελικό δέκτη, συνήθως μέσα από ένα σύνολο ενδιάμεσων σταθμών. Το πρωτόκολλο αυτό υλοποιείται σαν μια επέκταση του

SOAP και είναι ενσωματωμένο στην επικεφαλίδα SOAP. Το WS-Routing καθορίζει ένα πρότυπο τμήμα επικεφαλίδας SOAP, για να εκφράσει τις πληροφορίες της δρομολόγησης. Ο ρόλος του είναι να καθορίσει την ακριβή αλληλουχία ενδιαμέσων, διαμέσου των οποίων πρέπει να διέλθει το μήνυμα.

Η παράκαμψη δρομολόγησης (Routing Detour) είναι μια επίθεση του τύπου «Man in the Middle» με την οποία οι ενδιάμεσοι σταθμοί μπορεί να τελούν υπό ομηρία με σκοπό να δρομολογήσουν ευαίσθητα μηνύματα σε μια εξωτερική θέση. Οι πληροφορίες δρομολόγησης μπορεί να τροποποιηθούν κατά τη διάρκεια μεταφοράς της πληροφορίας και η διαδρομή της δρομολόγησης μπορεί να αφαιρεθεί από την κεφαλίδα και το μήνυμα, έτσι ώστε η πληροφορία που θα λάβει η εφαρμογή να είναι τέτοια που να μην υπάρχει η υποψία πως έχει γίνει παράκαμψη δρομολόγησης. Η κεφαλίδα και η εισαγωγή αντικειμένων σε αυτή είναι συνήθως λιγότερο προστατευμένα από το μήνυμα. Αυτό συμβαίνει λόγω του γεγονότος ότι η κεφαλίδα χρησιμοποιείται για να συμπεριλάβει όλα τα μεταδεδομένα της συναλλαγής όπως η γνησιότητα, η δρομολόγηση, η μορφοποίηση κ.λπ. Επίσης πολλές διαδικασίες μπορεί να σχετίζονται με την πρόσθεση/επεξεργασία της κεφαλίδας ενός αρχείου XML. Σε πολλές υλοποιήσεις η πληροφορία της δρομολόγησης μπορεί να λαμβάνεται από μια εξωτερική υπηρεσία ιστού, η οποία παρέχει την δεδομένη δρομολόγηση για τη συναλλαγή.

4.4.24. Εκμετάλλευση πινάκων σε μηνύματα SOAP (SOAP Array Abuse)

Οι πίνακες XML του SOAP είναι συνηθισμένος στόχος προκειμένου να επιτευχθεί κακόβουλη κατάχρηση. Οι πίνακες SOAP έχουν μια ή περισσότερες διαστάσεις (Rank) των οποίων τα μέλη βρίσκονται σε συγκεκριμένη θέση. Μια τιμή του πίνακα περιγράφεται ως μια σειρά από στοιχεία τα οποία απεικονίζονται στον πίνακα, με τα μέλη να εμφανίζονται σε αύξουσα σειρά. Όταν πρόκειται για πολυδιάστατους πίνακες, η τελευταία (προς τα δεξιά) διάσταση είναι αυτή που μεταβάλλεται ταχύτερα. Μια υπηρεσία ιστού, η οποία περιμένει ένα τέτοιο πίνακα, μπορεί να είναι ο στόχος μια XML DoS επίθεσης ωθώντας τον εξυπηρέτη SOAP να κατασκευάσει ένα τεράστιο πίνακα στη μνήμη της συσκευής, έτσι ώστε να προκαλέσει μια κατάσταση DoS στη συσκευή εξαιτίας της προ-δέσμευσης της μνήμης.

4.4.25. Έγχυση σε αρχεία συμπερίληψης εξυπηρέτη (SSI Injection)

Η έγχυση σε αρχεία συμπερίληψης εξυπηρέτη είναι μια τεχνική εκμετάλλευσης στην πλευρά του εξυπηρέτη, η οποία επιτρέπει στον εισβολέα να στείλει κώδικα μέσα σε μια εφαρμογή ιστού, ο οποίος αργότερα μπορεί να εκτελεστεί τοπικά από τον εξυπηρέτη ιστού. Η επίθεση αυτή εκμεταλλεύεται την αδυναμία μιας εφαρμογής ιστού να ελέγξει πλήρως τα δεδομένα εισόδου που παρέχει ο χρήστης, πριν τα εισαγάγει σε ένα αρχείο HTML που διερμηνεύεται στην πλευρά του εξυπηρέτη.

Ο εξυπηρέτης ιστού, πριν αποστείλει μια HTML σελίδα, μπορεί να αναλύσει και να εκτελέσει δηλώσεις συμπερίληψης αρχείων. Σε μερικές περιπτώσεις δε, όπως σε πίνακες ανακοινώσεων, σε βιβλία επισκεπτών ή σε συστήματα διαχείρισης περιεχομένου, μια εφαρμογή ιστού μπορεί να εισαγάγει τα δεδομένα που παρέχουν οι χρήστες κατευθείαν στον κώδικα της ιστοσελίδας.

Εάν ένας εισβολέας υποβάλει μια δήλωση συμπερίληψης αρχείου, τότε μπορεί να έχει την δυνατότητα να εκτελέσει αυθαίρετες εντολές του λειτουργικού συστήματος, ή να εμφανίσει το περιεχόμενο ενός περιορισμένου αρχείου, την επόμενη φορά που θα ανανεωθεί η ιστοσελίδα.

4.4.26. Κακόβουλος ορισμός αναγνωριστικών συνόδου (Session Fixation)

Ο κακόβουλος ορισμός αναγνωριστικών συνόδου είναι μια τεχνική επίθεσης η οποία θέτει την ταυτότητα συνεδρίας ενός χρήστη (session ID) σε μια συγκεκριμένη τιμή. Ανάλογα με τη λειτουργικότητα της ιστοσελίδας στόχου, ένα πλήθος από τεχνικές μπορεί να χρησιμοποιηθούν για να θέσουν (fix) την τιμή της ταυτότητας συνεδρίας. Αυτές οι τεχνικές περιλαμβάνουν Cross Site Scripting εκμεταλλευόμενες την εκτέλεση προηγούμενων αιτημάτων HTTP στην ιστοσελίδα. Όταν η ταυτότητα συνεδρίας του χρήστη έχει τεθεί, ο εισβολέας περιμένει τον χρήστη για να συνδεθεί. Όταν πραγματοποιηθεί αυτό, ο εισβολέας χρησιμοποιεί την προκαθορισμένη τιμή της ταυτότητας συνεδρίας για να αναλάβει απευθείας την ταυτότητα του χρήστη.

Γενικά, υπάρχουν δύο τύποι συστημάτων διαχείρισης συνεδρίας που αφορούν τις τιμές ταυτοποίησης (ID). Ο πρώτος τύπος είναι το **ανεκτικό σύστημα** το οποίο επιτρέπει στην εφαρμογή πλοήγησης ιστού να καθορίσει οποιαδήποτε ταυτότητα (ID). Ο δεύτερος τύπος είναι το **αυστηρό σύστημα** κατά το οποίο επιτρέπει μόνο τις τιμές οι οποίες δημιουργούνται στην πλευρά του εξυπηρέτη ιστού. Στα ανεκτικά συστήματα, οι αυθαίρετες τιμές ταυτότητας συνεδρίας διατηρούνται χωρίς καμία επαφή με την ιστοσελίδα. Στα αυστηρά συστήματα, απαιτείται από τον εισβολέα να διατηρεί μια συνεδρία-παγίδα (trap-session), διατηρώντας περιοδική επαφή με την ιστοσελίδα, αποτρέποντας με αυτό τον τρόπο την εκπνοή των χρονομετρητών λήξης της συνεδρίας (timeouts).

Χωρίς ενεργή προστασία ενάντια στον κακόβουλο ορισμό αναγνωριστικών συνόδου , η εισβολή μπορεί να τοποθετηθεί έναντι οποιουδήποτε δικτυακού τόπου που χρησιμοποιεί συνεδρίες για να προσδιορίσει τους πιστοποιημένους χρήστες. Οι τοποθεσίες ιστού που χρησιμοποιούν την ταυτότητα συνεδρίας βασίζονται συνήθως σε cookies, μέσω διευθύνσεων URL καθώς επίσης και μέσω κρυφών πεδίων φόρμας. Δυστυχώς, οι συνεδρίες που βασίζονται σε cookies είναι και οι πιο εύκολοι στόχοι για να δεχθούν επίθεση. Οι περισσότερες από τις μεθόδους επίθεσης αποσκοπούν στην ρύθμιση (fixation) των cookies.

Σε αντίθεση με την κλοπή της ταυτότητας συνεδρίας ενός χρήστη αφού αυτός έχει συνδεθεί σε μια ιστοσελίδα, ο κακόβουλος ορισμός αναγνωριστικών συνόδου παρέχει περισσότερες πιθανότητες πετυχημένης επίθεσης. Το ενεργό μέρος της επίθεσης λαμβάνει χώρα πριν ο χρήστης συνδεθεί (κατά τη διαδικασία του login).

4.4.27. Έγχυση σε εντολές SQL (SQL Injection)

Η έγχυση σε εντολές SQL είναι μια τεχνική επίθεσης η οποία χρησιμοποιείται για να εκμεταλλευτεί εφαρμογές οι οποίες κατασκευάζουν εντολές SQL, από δεδομένα εισόδου τα οποία παρέχουν οι χρήστες.

Η SQL (Structured Query Language) είναι μια ειδική γλώσσα προγραμματισμού με σκοπό την αποστολή ερωτημάτων σε βάσεις δεδομένων. Η SQL είναι πρότυπο τόσο από τον οργανισμό ANSI όσο και ISO, ωστόσο πολλά συστήματα βάσεων

δεδομένων υλοποιούν επεκτάσεις προς το βασικό πρότυπο. Οι εφαρμογές συχνά χρησιμοποιούν τα δεδομένα εισόδου των χρηστών για να δημιουργήσουν προσαρμοσμένες εντολές SQL. Εάν μια εφαρμογή αποτύχει στο να δημιουργήσει σωστά μια κατάλληλη εντολή SQL, είναι δυνατό για έναν εισβολέα να τροποποιήσει την εντολή και να εκτελέσει αυθαίρετες και πιθανά εχθρικές εντολές. Όταν ο εισβολέας είναι σε θέση να τροποποιήσει μια εντολή SQL, η διαδικασία θα εκτελεστεί με τα ίδια δικαιώματα που έχει και το στοιχείο που εκτελεί την εντολή (π.χ. ο εξυπηρέτης Βάσης δεδομένων, ο εξυπηρέτης ιστού κ.λπ.). Αυτή η δυνατότητα επιτρέπει στους εισβολείς να αποκτήσουν τον πλήρη έλεγχο της βάσης δεδομένων περιλαμβανομένης και της δυνατότητας να εκτελέσουν εντολές στο σύστημα.

4.4.28. Κατάχρηση ανακατεύθυνσης URL (URL Redirector Abuse)

Η κατάχρηση ανακατεύθυνσης URL, είναι ένα είδος επίθεσης που αντιπροσωπεύει κοινές λειτουργίες, οι οποίες χρησιμοποιούνται από τις εφαρμογές ιστού για να προωθηθεί ένα εισερχόμενο αίτημα σε έναν εναλλακτικό πόρο. Αυτό μπορεί να γίνει για διάφορους λόγους, αλλά συνήθως γίνεται για να επιτρέψει σε πόρους να μετακινηθούν μέσα στη δομή του καταλόγου και να αποφύγουν μια ενδεχόμενη δυσλειτουργία για χρήστες που ζητούν τον πόρο με την παλιά του διεύθυνση. Μπορεί επίσης να χρησιμοποιηθεί για να εφαρμόσει εξισορρόπηση φορτίου ή για την καταγραφή εξερχομένων συνδέσμων. Η τελευταία χρηστικότητα αξιοποιείται συχνά και σε επιθέσεις phishing. Η ανακατεύθυνση URL δεν σημαίνει απαραίτητα μια ευπάθεια ασφαλείας, αλλά μπορεί να χρησιμοποιηθεί από τους εισβολείς κάνοντας τα θύματα να πιστεύουν πως πλοηγούνται σε μια ιστοσελίδα διαφορετική από αυτή που επιθυμούν στην πραγματικότητα.

4.4.29. Έγχυση σε XPath (XPath Injection)

Είναι μια τεχνική επίθεσης που χρησιμοποιείται για να εκμεταλλευτεί αδυναμίες εφαρμογών οι οποίες συντάσσουν ερωτήματα XPath (XML Path Language) από δεδομένα που εισάγει ο χρήστης για να κάνουν ερωτήματα ή να πλοηγηθούν σε έγγραφα XML. Μπορεί να χρησιμοποιηθεί απευθείας από μια εφαρμογή για να κάνει ερώτημα σε ένα XML έγγραφο, ή ως μέρος μιας συνολικότερης διαδικασίας όπως η εφαρμογή μιας μετατροπής XSLT σε ένα έγγραφο XML ή ακόμα και εφαρμογή ενός ερωτήματος XQuery σε ένα έγγραφο XML. Η σύνταξη της XPath έχει κάποιες ομοιότητες με τα ερωτήματα της SQL. Για παράδειγμα μπορούμε να θεωρήσουμε ένα έγγραφο XML το οποίο περιέχει στοιχεία με το όνομα «user», κάθε ένα από τα οποία, περιέχει τρία επιμέρους υπο-στοιχεία «name», «password» και «account». Το ακόλουθο ερώτημα XPath ζητά να επιστραφεί το υπο-στοιχείο «account» του χρήστη, του οποίου το υπο-στοιχείο “name” έχει την τιμή “jsmith” και το υπο-στοιχείο “password” έχει την τιμή “demo1234” (ή μία κενή συμβολοσειρά, αν δεν υπάρχει αυτός ο χρήστης):

```
String(//user[name/text()='jsmith' and
Password/text()='Demo1234']/account/text())
```

Εάν μια εφαρμογή δημιουργεί τα ερωτήματα XPath κατά την εκτέλεση και ενσωματώνει στο κείμενο των ερωτημάτων μη ασφαλή δεδομένα των χρηστών, είναι δυνατό για έναν εισβολέα που πραγματοποιεί επίθεση στην εφαρμογή να

εγχύσει δεδομένα μέσα στο ερώτημα, τέτοια που το νέο-σχεδιασμένο ερώτημα να αναλύεται με ένα τρόπο διαφορετικό από αυτόν που ήθελε ο προγραμματιστής.

4.4.30. Καταιγισμός γνωρισμάτων XML (XML Attribute Blowup)

Η επίθεση τύπου Καταιγισμού γνωρισμάτων στοχεύει στην άρνησης υπηρεσίας (DoS), και εκδηλώνεται ενάντια σε συντακτικούς αναλυτές XML (XML parsers[13]). Ο εισβολέας διοχετεύει ένα κατάλληλα διαμορφωμένο έγγραφο XML, το οποίο οδηγεί τον αναλυτή XML να το αναλύσει με τρόπο αναποτελεσματικό, οδηγώντας σε υψηλά επίπεδα φόρτου για τη CPU. Η ουσία της επίθεσης είναι να συμπεριλάβει πολυάριθμα στοιχεία «ιδιότητας» (attribute) στον ίδιο κόμβο του εγγράφου XML. Οι ευάλωτοι XML αναλυτές διαχειρίζονται τις ιδιότητες με μη αποτελεσματικό τρόπο.

4.4.31. Εξωτερικές οντότητες XML (XML External Entities)

Η τεχνική XML External Entities, χρησιμοποιεί ένα χαρακτηριστικό της XML, το οποίο επιτρέπει τη δυναμική δημιουργία εγγράφων, τη στιγμή της επεξεργασίας. Ένα μήνυμα XML, μπορεί να ορίζει τα δεδομένα άμεσα ή να υποδεικνύει ένα εξωτερικό URI[14] (Uniform Resource Identifier) όπου θα βρεθούν τα δεδομένα: οι αναφορές σε URI είναι αυτές μέσω των οποίων ορίζονται οι εξωτερικές οντότητες. Σε αυτή την τεχνική επίθεσης, οι εξωτερικές οντότητες μπορεί να αντικαταστήσουν την κανονική τιμή της οντότητας με κακόβουλα δεδομένα, να εναλλάξουν παραπομπές, ακόμα και θέσουν σε κίνδυνο την ασφάλεια των δεδομένων στα οποία έχει πρόσβαση η εφαρμογή XML ή και ο εξυπηρέτης.

4.4.32. Ανάπτυξη οντοτήτων XML (XML Entity Expansion)

Η επίθεση επέκτασης οντότητας (XML Entity Expansion) εκμεταλλεύεται μια δυνατότητα στα πρότυπα DTDs[15] της XML, η οποία επιτρέπει τη δημιουργία προσαρμοσμένων μακροεντολών που καλούνται *οντότητες* και οι οποίες μπορεί να χρησιμοποιηθούν οπουδήποτε στο έγγραφο XML. Ορίζοντας αναδρομικά ένα σύνολο προσαρμοσμένων οντοτήτων στο έγγραφο, ο εισβολέας μπορεί να

[13] XML Parser ή ένας συντακτικός αναλυτής XML είναι ένα λογισμικό το οποίο διαβάζει αρχεία XML, ελέγχει την ορθή μορφοποίηση τους και την εγκυρότητα τους και τα προωθεί για περαιτέρω επεξεργασία. Ο συντακτικός αναλυτής XML είναι η πιο απλή μορφή ενός επεξεργαστή XML και είναι απαραίτητο τμήμα κάθε εφαρμογής XML.

[14] Μια από τις δυσκολίες που ανακύπτουν κατά την εργασία με XML έγγραφα που ανακτώνται από πολλές πηγές, είναι ότι η έννοια των ετικετών της κάθε εφαρμογής, μπορεί να γίνει πολλές φορές ασαφής. Το πρόβλημα αυτό το επιλύουν οι χώροι ονοματοδοσίας XML, όπου επιτρέπουν στους χρήστες να εξαλείψουν την ασάφεια της έννοιας των XML στοιχείων και ιδιοτήτων. Οι χώροι ονοματοδοσίας XML επιτρέπουν στους χρήστες να προσδιορίσουν ονόματα στοιχείων και ιδιοτήτων, συσχετίζοντας τα με χώρους ονοματοδοσίας που συνδέονται με αναφορές URI.

[15] Τεχνολογία μοντελοποίησης των XML εγγράφων. Ένα DTD παρέχει στις εφαρμογές τη χρήσιμη πληροφορία για τα ποια ονόματα και δομές μπορούν να χρησιμοποιηθούν σε ένα συγκεκριμένο τύπο εγγράφου. Η χρήση ενός DTD κατά τη σύνταξη των εγγράφων σημαίνει ότι μπορεί να βεβαιωθεί ότι όλα τα έγγραφα που ανήκουν σε αυτό τον συγκεκριμένο τύπο θα δομηθούν και θα ονομαστούν σύμφωνα με ένα συνεπή και ομοιόμορφο τρόπο. Τα DTDs είναι απλοϊκά και περιορισμένα. Κάθε DTD αποτελείται από ένα σύνολο από κανόνες ή δηλώσεις. Κάθε δήλωση προσθέτει ένα νέο στοιχείο, σύνολο από ιδιότητες, οντότητες, ή σημειώσεις που πρόκειται να χρησιμοποιηθούν σε ένα έγγραφο XML.

υπερφορτώσει τον συντακτικό αναλυτή (parser) του εγγράφου, ο οποίος προσπαθεί να επιλύσει πλήρως τις οντότητες, αναγκάζοντάς τον να επεξεργάζεται σχεδόν επ' άπειρον σε αυτούς τους αναδρομικούς ορισμούς.

Ο επιτιθέμενος χρησιμοποιεί ένα κακόβουλο μήνυμα XML για να αναγκάσει τις οντότητες σε αναδρομική επέκταση (ή σε άλλη επαναλαμβανόμενη διαδικασία), η οποία εξαντλεί τους διαθέσιμους πόρους του εξυπηρέτη.

4.4.33. Έκχυση XML (XML Injection)

Η έγχυση XML (XML Injection) είναι μια τεχνική επίθεσης που χρησιμοποιείται από τους επιτιθέμενους για να χειραγωγήσουν ή να θέσουν σε κίνδυνο τη λογική μιας εφαρμογής ή υπηρεσίας XML. Η έγχυση απρόσμενου περιεχομένου XML ή ακόμα και δομών μέσα σε ένα έγγραφο XML μπορεί να εναλλάξει τη λογική της εφαρμογής. Επιπλέον η έγχυση XML μπορεί να προκαλέσει την εισαγωγή κακόβουλου περιεχομένου μέσα στο τελικό έγγραφο.

4.4.34. Έγχυση XQUERY (XQUERY Injection)

Η έγχυση XQuery είναι μια παραλλαγή της κλασικής επίθεσης έγχυσης SQL, ενάντια στη γλώσσα XML XQUERY. Η επίθεση χρησιμοποιεί δεδομένα που δεν έχουν ελεγχθεί πλήρως και τα οποία ενσωματώνονται στις εντολές XQuery. Με τον τρόπο αυτό ο επιτιθέμενος μπορεί να καταφέρει να εκτελέσει εντολές, οι οποίες μπορούν να έχουν αντίκτυπο σε όλα τα δεδομένα και υποσυστήματα στα οποία έχει το υποσύστημα XQuery. Η επίθεση μπορεί να χρησιμοποιηθεί για να απαριθμήσει στοιχεία στο περιβάλλον του θύματος, για να εγχύσει εντολές που θα εκτελεστούν στον εξυπηρέτη ή να εκτελέσει ερωτήματα σε απομακρυσμένα αρχεία και πηγές δεδομένων. Όπως και στην επίθεση έγχυσης SQL, ο εισβολέας διεισδύει μέσω του σημείου εισόδου της εφαρμογής για να αποκτήσει την απαιτούμενη πρόσβαση.

4.5. Ευπάθειες, (Weaknesses)

4.5.1. Κακή διαμόρφωση εφαρμογής (Application Misconfiguration)

Οι επιθέσεις τύπου κακής διαμόρφωσης εφαρμογής εκμεταλλεύονται τα λάθη που ενδεχομένως έχουν κάνει οι προγραμματιστές κατά τον προγραμματισμό και την παραμετροποίηση ή την συντήρηση των εφαρμογών ιστού. Πολλές εφαρμογές, διαθέτουν ενεργοποιημένες από προεπιλογή περιττές λειτουργίες με επισφαλή χαρακτηριστικά, όπως αποσφαλμάτωση (debug), χαρακτηριστικά διασφάλισης ποιότητας (quality assurance), κ.λπ. Οι λειτουργίες αυτές, για έναν επιτιθέμενο, μπορεί να είναι ο τρόπος για να παρακάμψει τις μεθόδους ελέγχου ταυτότητας και να αποκτήσει πρόσβαση σε ευαίσθητες πληροφορίες, πολλές φορές με αυξημένα προνόμια.

Ομοίως, εγκαταστάσεις στις οποίες δεν αλλάχτηκαν οι προεπιλεγμένες ρυθμίσεις, μπορεί να περιλαμβάνουν γνωστά ονόματα χρηστών και κωδικούς πρόσβασης, λογαριασμούς ενσωματωμένους στον κώδικα, εναλλακτικές διόδους πρόσβασης (backdoors), ειδικούς μηχανισμούς πρόσβασης και εσφαλμένα δικαιώματα αρχείων και καταλόγων τα οποία είναι προσβάσιμα μέσω των εξυπηρετών ιστού. Διαμορφώσεις εφαρμογών που δεν έχουν ασφαλιστεί κατάλληλα μπορεί να αποκαλύψουν τις συμβολοσειρές σύνδεσης (connection

strings) προς τις βάσεις δεδομένων, όπου τυπικά τα συνθηματικά δίνονται με μη κρυπτογραφημένο κείμενο. Επίσης, οι προεπιλεγμένες ρυθμίσεις των αρχείων διαμορφώσεων ενδέχεται να μην έχουν τεθεί με γνώμονα την ασφάλεια. Όλες αυτές οι επισφάλειες στην παραμετροποίηση κάποιας εφαρμογής μπορούν να επιτρέψουν σε έναν κακόβουλο χρήστη σε μη εξουσιοδοτημένη πρόσβαση στο σύστημα και στην υποκλοπή ευαίσθητων πληροφοριών.

4.5.2. Εμφάνιση λίστας περιεχομένων καταλόγων (Directory Indexing)

Η αυτόματη λίστα περιεχομένων καταλόγου είναι μία λειτουργία του εξυπηρέτη ιστού, η οποία απαριθμεί όλα τα αρχεία μέσα σε ένα κατάλογο εφ' όσον το εξ ορισμού αρχείο (index.html/home.html/default.htm/default.asp/default.aspx/index.php) δεν βρεθεί. Όταν ένας χρήστης αναζητά την κεντρική σελίδα μιας ιστοσελίδας, κανονικά πληκτρολογεί ένα URL, όπως HTTP://www.example.com/directory1/, χρησιμοποιώντας το όνομα του τομέα (domain) και χωρίς να παραθέσει κάποιο συγκεκριμένο αρχείο. Ο εξυπηρέτης ιστού επεξεργάζεται το αίτημα αυτό και αναζητά στον κατάλογο ρίζας (root directory) το εξ ορισμού αρχείο, και στέλνει αυτή την σελίδα πίσω στον πελάτη. Εάν αυτή η σελίδα δεν υφίσταται, ο εξυπηρέτης ιστού θα διαμορφώσει δυναμικά μια λίστα αρχείων που περιέχονται στον κατάλογο και θα στείλει τα εξαγόμενα αποτελέσματα στον πελάτη. Ουσιαστικά, αυτό ισοδυναμεί με την εντολή "ls" στο Unix ή με την εντολή "dir" στα Windows μέσα σε αυτόν τον κατάλογο και την παρουσίαση των αποτελεσμάτων σε μορφή HTML. Σχετικά με την προοπτική της ασφαλείας, είναι σημαντικό να συνειδητοποιήσουμε ότι μπορεί να είναι δυνατόν να αποσταλούν λίστες περιεχομένων καταλόγων χωρίς να είναι αυτή η πρόθεση του ιδιοκτήτη του ιστοχώρου, λόγω ευπαθειών του λογισμικού τις οποίες εκμεταλλεύονται οι επιτιθέμενοι υποβάλλοντας συγκεκριμένες αιτήσεις.

Όταν ένας εξυπηρέτης ιστού αποκαλύψει τα περιεχόμενα ενός καταλόγου, τότε τα εξαγόμενα δεδομένα είναι πιθανό να περιέχουν πληροφορίες που δεν προορίζονται για δημόσια προβολή. Θα πρέπει να σημειωθεί ότι δεν θα πρέπει οι διαχειριστές να βασίζονται στο μοντέλο «ασφάλεια μέσω συσκότισης», δηλ. να δίνουν μη αναμενόμενα ονόματα σε πόρους υποθέτοντας ότι με τον τρόπο αυτό κανείς δεν θα τους εντοπίσει: πλέον, υπάρχουν σαρωτές ευπαθειών (vulnerability scanners) όπως είναι ο Wikto, ο οποίος μπορεί να ελέγξει για ύπαρξη πόρων, διαμορφώνοντας μάλιστα τη λίστα των URI που θα ελέγξει δυναμικά, καθώς με βάση τα αποτελέσματα των αρχικών ελέγχων μπορεί να προστεθούν ή να αφαιρεθούν περαιτέρω έλεγχοι που θα γίνουν. Η ευπάθεια αυτή, αν και δυνητικά αβλαβής, θα μπορούσε να επιτρέψει την διαρροή πληροφοριών σε έναν εισβολέα, που θα τον εφοδιάσει με τις απαραίτητες πληροφορίες για να δρομολογήσει περαιτέρω επιθέσεις εναντίον του συστήματος.

4.5.3. Ακατάλληλα δικαιώματα στο σύστημα αρχείων (Improper Filesystems Permission)

Η ευπάθεια του τύπου «ακατάλληλα δικαιώματα αρχείων» είναι μια απειλή που αφορά την εμπιστευτικότητα, την ακεραιότητα και τη διαθεσιμότητα μιας εφαρμογής ιστού. Το πρόβλημα προκύπτει όταν λανθασμένα δικαιώματα αρχείων,

έχουν εφαρμοστεί σε αρχεία, φακέλους, και συμβολικούς συνδέσμους. Όταν σε ένα σύστημα δεν έχουν εφαρμοστεί τα σωστά δικαιώματα, ένας εισβολέας μπορεί να είναι σε θέση να αποκτήσει πρόσβαση σε προστατευόμενα αρχεία ή σε καταλόγους και να τροποποιήσει ή να διαγράψει το περιεχόμενό τους. Για παράδειγμα, αν ένας ανώνυμος λογαριασμός χρήστη έχει δικαιώματα εγγραφής σε ένα αρχείο, τότε ο εισβολέας μπορεί να τροποποιήσει το περιεχόμενο του αρχείου και να επηρεάσει την εφαρμογή με ανεπιθύμητο τρόπο. Επίσης, ένας εισβολέας μπορεί να εκμεταλλευτεί ακατάλληλους συμβολικούς συνδέσμους (symlinks[16]) με σκοπό να κλιμακώσει τα προνόμιά του ή να αποκτήσει μη εξουσιοδοτημένη πρόσβαση σε αρχεία. Για παράδειγμα, ένας συμβολικός σύνδεσμος που παραπέμπει σε έναν κατάλογο έξω από τον κατάλογο ρίζας (root catalog) ενός ιστοχώρου μπορεί δυνητικά να δώσει στον επιτιθέμενο πρόσβαση σε οποιοδήποτε αρχείο.

Παρακάτω παραθέτω μερικά από τα δικαιώματα που σχετίζονται με αρχεία:

- Read
- Write
- Modify
- Execute
- List Folder Contents
- Traverse Folder
- List Folder
- Read Attributes
- Read Extended Attributes
- Create Files/Write Data
- Create Folders/Append Data
- Write Attributes
- Write Extended Attributes
- Delete Subfolders and Files
- Delete Read Permissions
- Change Permissions
- Take Ownership and Synchronize

Κάθε αρχείο, κατάλογος ή συμβολικός σύνδεσμος για το λειτουργικό σύστημα και τον εξυπηρέτη ιστού, έχει ένα σύνολο από δικαιώματα που συνδέονται με αυτό.

Οι εξυπηρέτες ιστού χρησιμοποιούν ένα λογαριασμό στο λειτουργικό σύστημα που συστήματος που φιλοξενούνται, για να έχουν πρόσβαση στους πόρους που προσφέρει το σύστημα καθώς και το σύστημα αρχείων του. Ο λογαριασμός του λειτουργικού συστήματος διαθέτει ένα σύνολο των δικαιωμάτων που απαιτούνται, για την πρόσβαση του στον πηγαίο κώδικα εντολών και scripts και την εκτέλεσή τους στον εξυπηρέτη. Όταν η εφαρμογή πλοήγησης του χρήστη ζητάει ένα αρχείο, τότε ο εξυπηρέτης ιστού αποφασίζει πώς να εξυπηρετήσει το αρχείο αυτό, με βάση

[16] Από en.wikipedia.org (HTTP://en.wikipedia.org/wiki/Symbolic_link)

τον τύπο του και τις προκαθορισμένες ρυθμίσεις ασφαλείας. Στην περίπτωση που ο χρήστης ζητάει ένα αρχείο HTML, ο εξυπηρέτης ιστού προσπαθεί να φορτώσει το αρχείο από το σύστημα αρχείων (file system), χρησιμοποιώντας τον λογαριασμό που διαθέτει στο σύστημα αυτό. Ανάλογα με τα δικαιώματα που έχουν εκχωρηθεί στο αρχείο, ο εξυπηρέτης ιστού είτε θα καταφέρει να διαβάσει το αρχείο και να επιστρέψει τα περιεχόμενά του ως απάντηση στο αίτημα ή θα επιστρέψει ένα μήνυμα σφάλματος επισημαίνοντας την άρνηση πρόσβασης (403). Αν ο χρήστης ζητήσει ένα script (π.χ. default.jsp), τότε ο εξυπηρέτης ιστού θα καθορίσει τον τρόπο που θα επεξεργαστεί το αίτημα αυτό και θα του επιτρέψει να χειριστεί την αίτηση. Αν το script είναι χαρακτηρισμένο ως μόνο για ανάγνωση (read only), και δεν διαθέτει δικαιώματα εκτέλεσης, ο εξυπηρέτης ιστού μπορεί να στείλει άμεσα το αρχείο στον χρήστη αντί της εκτέλεσης του κώδικα μέσα στο αρχείο jsp[17].

4.5.4. Ακατάλληλος χειρισμός εισόδου (Improper Input Handling)

Μια από τις πιο κοινές ευπάθειες που έχουν εντοπιστεί σε δικτυακές εφαρμογές είναι ο «Ακατάλληλος χειρισμός εισόδου» (Improper Input Handling). Η κακή διαχείριση εισόδου του χρήστη, είναι η κύρια αιτία για μια σειρά από κρίσιμης σημασίας τρωτά σημεία που έχουν τα συστήματα και οι εφαρμογές.

Γενικά, ο όρος *χειρισμός εισόδου* χρησιμοποιείται για να περιγράψει λειτουργίες όπως η επικύρωση, φιλτράρισμα, κωδικοποίηση και αποκωδικοποίηση των δεδομένων εισόδου. Οι εφαρμογές λαμβάνουν στοιχεία εισόδου από διάφορες πηγές. Μερικές πηγές που μπορούμε να αναφέρουμε είναι οι χρήστες, οι πράκτορες λογισμικού (εφαρμογές πλοήγησης), καθώς και δικτυακές περιφερειακές συσκευές. Στην περίπτωση των εφαρμογών ιστού, η είσοδος μπορεί να επιτευχθεί σε διάφορες μορφές (ζεύγη τιμών με όνομα, JSON[18], SOAP, κ.λπ.) και η λήψη τους μπορεί να γίνει μέσω URL που περιέχουν συμβολοσειρές ερωτημάτων (query strings), δεδομένων POST, επικεφαλίδες HTTP, cookies, κ.τ.λ. Σε εφαρμογές οι οποίες δεν είναι δικτυακές, οι αιτήσεις εισόδου μπορούν να ληφθούν μέσω μεταβλητών της εφαρμογής, όπως μεταβλητές περιβάλλοντος, μητρώο, αρχεία ρυθμίσεων, κ.λπ. Ανεξάρτητα από τη μορφή των δεδομένων, την πηγή ή την τοποθεσία της εισόδου, όλα τα δεδομένα εισόδου πρέπει να θεωρούνται μη αξιόπιστα και ενδεχομένως κακόβουλα. Οι εφαρμογές που επεξεργάζονται μη αξιόπιστα δεδομένα εισόδου, μπορούν να καταστούν ευάλωτες σε διάφορες επιθέσεις, όπως υπερχείλιση, έγχυση SQL, εντολών λειτουργικού συστήματος, άρνηση υπηρεσία κ.λπ. Οι εφαρμογές που επεξεργάζονται μη αξιόπιστα δεδομένα εισόδου πρέπει να προβαίνουν στον πλήρη έλεγχο των δεδομένων εισόδου, συμπεριλαμβάνοντας ελέγχους για τον τύπο, το εύρος τιμών, το μήκος και το συντακτικό. Οι έλεγχοι πρέπει να γίνονται στην πλευρά του εξυπηρέτη και όχι του εξυπηρετούμενου (π.χ. στην εφαρμογή πλοήγησης) καθώς ο τελευταίος θεωρείται αναξιόπιστος, ενώ πρέπει να εκτελούνται και οι διαδικασίες κανονικοποίησης (π.χ. επεξεργασία όλων των συμβολισμών της μορφής %xx, & κ.λπ. πριν τη

[17] **Java Server Pages** HTTP://en.wikipedia.org/wiki/JavaServer_Pages

[18] **JavaScript Object Notation**, HTTP://en.wikipedia.org/wiki/Json

διενέργεια των ελέγχων), χρήση χαρακτήρων διαφυγής και ταίριασμα έναντι λευκής ή μαύρης λίστας.

4.5.5. Ακατάλληλος χειρισμός εξόδου (Improper Output Handling)

Με την έννοια διαχείριση εξόδου αναφερόμαστε στον τρόπο κατά τον οποίο, μια εφαρμογή δημιουργεί δεδομένα προς έξοδο. Εάν η εφαρμογή δεν διαθέτει τον τρόπο για να διαχειριστεί κατάλληλα την έξοδο, τα δεδομένα εξόδου μπορούν να καταναλωθούν με τρόπο που να οδηγεί σε ευπάθειες και σε ενέργειες που δεν είναι προτιθέμενες από τον κατασκευαστή της εφαρμογής.

Κάθε σημείο από το οποίο τα δεδομένα εξέρχονται από την εφαρμογή, μπορεί να υποπέσει σε ακατάλληλη διαχείριση εξόδου. Τα όρια της εφαρμογής υφίσταται όταν τα δεδομένα εγκαταλείπουν την εφαρμογή και μπαίνουν ως είσοδος σε κάποια άλλη εφαρμογή. Το πέρασμα των δεδομένων σε άλλες εφαρμογές μπορεί να πραγματοποιηθεί μέσω των υπηρεσιών ιστού, της γραμμής εντολών, των υποδοχών των συσκευών, των μεταβλητών του περιβάλλοντος κ.ο.κ. Περιλαμβάνει επίσης το πέρασμα των δεδομένων μεταξύ επιπέδων που υπάρχουν στην αρχιτεκτονική της εφαρμογής, όπως η βάση δεδομένων, ο εξυπηρέτης καταλόγου, ο διερμηνευτής JavaScript , η εφαρμογή πλοήγησης ή το λειτουργικό σύστημα.

Η ακατάλληλη διαχείριση εξόδου μπορεί να έχει διάφορες μορφές μέσα σε μια εφαρμογή. Οι μορφές αυτές μπορούν να ταξινομηθούν σε: **σφάλματα πρωτόκολλου** και **σφάλματα σχετιζόμενα με τους καταναλωτές δεδομένων**. Τα σφάλματα πρωτοκόλλου, περιλαμβάνουν ελλειμματική ή ακατάλληλη κωδικοποίηση εξόδου ή έξοδο των μη έγκυρων δεδομένων. Τα σφάλματα της εφαρμογής, περιλαμβάνουν σφάλματα λογικής, όπως την εξαγωγή λανθασμένων δεδομένων ή τη μεταφορά μη φιλτραρισμένου κακόβουλου περιεχόμενου. Εάν η εφαρμογή δεν είναι σε θέση να διακρίνει σωστά το ορθό περιεχόμενο από το προβληματικό ή δεν λαμβάνει μέτρα για την αντιμετώπιση γνωστών ευπαθειών, τότε αυτό μπορεί να οδηγήσει σε κατάχρηση από την πλευρά του καταναλωτή δεδομένων, η οποία προκαλείται από τον εσφαλμένο χειρισμό εξόδου.

Μια εφαρμογή η οποία δεν παρέχει δεδομένα στο σωστό πλαίσιο, μπορεί να επιτρέψει σε έναν εισβολέα να κάνει κατάχρηση των δεδομένων των χρηστών. Αυτό μπορεί να οδηγήσει σε συγκεκριμένες απειλές οι οποίες αναφέρονται λεπτομερώς στην κατάταξη απειλών του WASC, όπως: Content Spoofing, Cross-Site Scripting, HTTP Response Splitting, HTTP Response Smuggling, LDAP Injection, OS Commanding, Routing Detour, Soap Array Abuse, URL Redirector, XML Injection, XQuery Injection, XPath Injection, Mail Command Injection, Null Injection και SQL Injection.

Η διαχείριση εξόδου με τον σωστό τρόπο εμποδίζει την απροσδόκητη ή την ακούσια ερμηνεία των δεδομένων από τον χρήστη. Για την επίτευξη αυτού του στόχου, οι προγραμματιστές πρέπει να κατανοήσουν το μοντέλο δεδομένων της εφαρμογής, τον τρόπο με τον οποίο τα δεδομένα θα πρέπει να παραδίδονται σε άλλα τμήματα της εφαρμογής, και πώς τελικά θα παρουσιαστούν στον χρήστη. Μερικές τεχνικές για την εξασφάλιση του ορθού χειρισμού των δεδομένων εξόδου είναι το φιλτράρισμα και η εξυγίανση των δεδομένων. Οι προγραμματιστές, για να εξασφαλίσουν «άμυνα σε βάθος», θα πρέπει να υποθέτουν ότι όλα τα δεδομένα μέσα στην εφαρμογή δεν είναι αξιόπιστα, και με αυτή την υπόθεση θα πρέπει να

επιλέγουν τους τρόπους με τους οποίους θα χειριστούν τα πιθανά σενάρια εξόδου των δεδομένων.

Ενώ η σωστή διαχείριση της εξόδου μπορεί να λάβει πολλές διαφορετικές μορφές, η εφαρμογή δεν μπορεί να είναι ασφαλής αν δεν προστατεύει τον χρήστη, από ανεπιθύμητες ερμηνείες. Αυτή η βασική απαίτηση είναι απαραίτητη για μια εφαρμογή, προκειμένου να χειριστεί με ασφάλεια τα δεδομένα εξόδου.

4.5.6. Διαρροή πληροφοριών (Information Leakage)

Η διαρροή πληροφοριών είναι μια ευπάθεια της εφαρμογής όταν αυτή αποκαλύπτει ευαίσθητα δεδομένα, όπως τεχνικές λεπτομέρειες της εφαρμογής ιστού, το περιβάλλον εγκατάστασης και τα δεδομένα των χρηστών του συστήματος. Τα ευαίσθητα δεδομένα μπορούν να χρησιμοποιηθούν από τον επιτιθέμενο για να εκμεταλλευτεί την εφαρμογή, το δίκτυο που τη φιλοξενεί, ή τους χρήστες της. Ως εκ τούτου, η διαρροή των ευαίσθητων δεδομένων πρέπει να περιορίζεται ή να προλαμβάνεται όπου αυτό είναι δυνατόν. Η διαρροή πληροφοριών, στην πιο κοινή μορφή της, είναι το αποτέλεσμα ενός ή περισσοτέρων από τις παρακάτω προϋποθέσεις: αποτυχία στην διαγραφή των σχολίων εκείνων που περιέχουν ευαίσθητες πληροφορίες, ακατάλληλη διαμόρφωση της εφαρμογής ή του εξυπηρέτη, ή οι διαφορές στις απαντήσεις της ιστοσελίδας για έγκυρα έναντι μη έγκυρων δεδομένων.

Η αποτυχία στην διαγραφή των σχολίων πριν από έκθεση του κώδικα σε περιβάλλον παραγωγής μπορεί να οδηγήσει σε διαρροή ευαίσθητων δεδομένων, από στοιχεία τα οποία μπορούν να εξαχθούν ως συμπεράσματα, όπως είναι η δομή καταλόγου του εξυπηρέτη, η δομή της SQL από την κατασκευή των ερωτημάτων, καθώς και πληροφορίες για την δομή του δικτύου. Συχνά, οι προγραμματιστές αφήνουν σχόλια μέσα στον κώδικα HTML ή τα scripts που δημιουργούν, προκειμένου να διευκολυνθούν στη διαδικασία εντοπισμού των σφαλμάτων (debugging) ή στην ενσωμάτωση πρόσθετου ή και διορθωτικού κώδικα κατά τη φάση της προ-παραγωγής. Τα σχόλια των προγραμματιστών παρόλο που δεν επιφέρουν καμία ζημιά στην εφαρμογή, θα πρέπει να αφαιρούνται πριν η εφαρμογή τεθεί σε παραγωγική λειτουργία.

Οι αριθμοί έκδοσης λογισμικού και τα μηνύματα λάθους με περιττές λεπτομέρειες είναι παραδείγματα κακής διαμόρφωσης εξυπηρέτη. Η πληροφορία αυτή είναι χρήσιμη για τον εισβολέα, παρέχοντας του, λεπτομερή στοιχεία ως προς το περιβάλλον ανάπτυξης, την γλώσσα προγραμματισμού, ή τις προ-ενσωματωμένες λειτουργίες που χρησιμοποιούνται από την εφαρμογή ιστού. Οι περισσότερες εργοστασιακές ρυθμίσεις των εξυπηρετητών δίνουν τον αριθμό έκδοσης του λογισμικού και περιττά μηνύματα λάθους για τον εντοπισμό σφαλμάτων (debugging) και την αντιμετώπιση δυσλειτουργιών (troubleshooting). Οι αλλαγές που πρέπει να γίνουν στην διαμόρφωση, είναι μεταξύ άλλων η απενεργοποίηση αυτών των δυνατοτήτων, καθώς και η μη εμφάνιση των εν λόγω πληροφοριών.

Ιστοσελίδες που παρέχουν διαφορετικές απαντήσεις με βάση την εγκυρότητα των δεδομένων, μπορεί επίσης να οδηγήσουν σε διαρροή πληροφοριών, ειδικά όταν δεδομένα που θεωρούνται εμπιστευτικά, φανερώνονται, ως αποτέλεσμα του σχεδιασμού της εφαρμογής. Ορισμένα παραδείγματα ευαίσθητων δεδομένων που

υπόκεινται σε διαρροή είναι: αριθμοί λογαριασμών, αναγνωριστικά χρήστη, αριθμός άδειας οδήγησης, αριθμός διαβατηρίου, αριθμός κοινωνικής ασφάλισης καθώς και κωδικοί πρόσβασης, συνεδρίες, διευθύνσεις κ.λπ. Οι αριθμοί πιστωτικών καρτών και άλλες ευαίσθητες πληροφορίες είναι παραδείγματα δεδομένων των χρηστών που πρέπει να έχουν επιπλέον προστασία από την έκθεση ή από διαρροές, ακόμη και στις περιπτώσεις που υπάρχει ήδη κατάλληλη κρυπτογράφηση και έλεγχος πρόσβασης.

4.5.7. Μη ασφαλής ευρετηριασμός (Insecure Indexing)

Ο μη ασφαλής ευρετηριασμός είναι μια απειλή που αφορά την εμπιστευτικότητα των δεδομένων των ιστοσελίδων. Ευρετηριάζοντας το περιεχόμενο μιας ιστοσελίδας μέσω μιας διαδικασίας η οποία έχει πρόσβαση σε αρχεία που δεν θα έπρεπε να είναι προσβάσιμα δημόσια, υπάρχει η πιθανότητα να υπάρξει διαρροή των πληροφοριών που σχετίζονται με την ύπαρξη των αρχείων αυτών, καθώς και για το περιεχόμενό τους. Κατά τη διαδικασία της ευρετηρίασης, οι πληροφορίες που συλλέγονται και αποθηκεύονται, μπορούν αργότερα να ανακτηθούν από έναν επιτιθέμενο, συνήθως μέσα από μια σειρά ερωτημάτων στην μηχανή αναζήτησης. Ο επιτιθέμενος όμως δεν μπορεί να παρακάμψει εύκολα το μοντέλο ασφαλείας της μηχανής αναζήτησης. Για αυτό τον λόγο η επίθεση του είδους αυτού, είναι λεπτή και πολύ δύσκολη στην ανίχνευση της και την αποτροπή της.

Όσο οι ιστοσελίδες γίνονται μεγαλύτερες και πολυπλοκότερες, το πρόβλημα των χρηστών για τον τρόπο με τον οποίο θα βρουν τις πληροφορίες που χρειάζονται, έχει όλο και πιο μεγάλη σημασία για τον ιδιοκτήτη της ιστοσελίδας. Εδώ έρχονται οι μηχανές αναζήτησης να λύσουν πρακτικά το πρόβλημα. Μια μηχανή αναζήτησης πρώτα «μαθαίνει» την ιστοσελίδα, εξετάζοντας τις σελίδες της και συνδέει λέξεις-κλειδιά ενημερώνοντας την εσωτερική βάση δεδομένων. Αυτή η διαδικασία ονομάζεται ευρετηριασμός (indexing). Στη συνέχεια, όταν ένας χρήστης εκτελεί ένα ερώτημα στη μηχανή αναζήτησης, τότε αυτή συμβουλεύεται τη βάση δεδομένων για να δώσει με την σειρά της ως απάντηση τον κατάλογο των σχετικών σελίδων. Η διαδικασία δημιουργίας ευρετηρίου εκτελείται συνεχώς ούτως ώστε να εξασφαλιστεί ότι η βάση δεδομένων της μηχανής αναζήτησης είναι ενημερωμένη. Υπάρχουν δύο είδη ευρετηρίασης: η εξ αποστάσεως (web/HTTP based) και η τοπική που πραγματοποιείται με βάση το αρχείο. Στην πρώτη κατηγορία, η μηχανή αναζήτησης διασχίζει την εφαρμογή από σελίδα σε σελίδα, συνήθως ξεκινώντας από την αρχική σελίδα του ιστοχώρου και συνεχίζοντας αναδρομικά ακολουθώντας συνδέσμους. Σε αυτή τη διαδικασία χρησιμοποιούνται απομακρυσμένες (3rd party) μηχανές αναζήτησης όπως το Google και το Yahoo. Στην δεύτερη περίπτωση, η μηχανή αναζήτησης θα πρέπει να έχει άμεση πρόσβαση στο εξυπηρέτη ιστού και στο σύστημα αρχείων, ευρετηριάζοντας την ιστοσελίδα με τη μετάβαση σε όλα τα αρχεία, μέχρι και στη ρίζα του συστήματος. Σε ορισμένες περιπτώσεις, αυτή η μέθοδος ευρετηρίασης μπορεί να ανοίξει τον χώρο για τις επιθέσεις αυτού του είδους.

4.5.8. Ανεπαρκή μέτρα έναντι αυτοματοποιημένης εκτέλεσης (Insufficient Anti-automation)

Η ευπάθεια αυτού του τύπου συμβαίνει όταν μια εφαρμογή ιστού επιτρέπει σε έναν εισβολέα να αυτοματοποιήσει μια διαδικασία, που αρχικά είχε σχεδιαστεί για να πραγματοποιηθεί με χειροκίνητο τρόπο, όπως για παράδειγμα από έναν άνθρωπο.

Οι λειτουργίες μιας εφαρμογής ιστού που τυγχάνουν συχνά στόχοι επιθέσεων αυτοματισμού, μπορεί να περιλαμβάνουν:

- Εφαρμογές με **φόρμες εισόδου**: Ο εισβολέας μπορεί να αυτοματοποιήσει με αιτήματα ωμής βίας “brute force” την αίτηση εισόδου στην εφαρμογή, σε μια προσπάθεια του, να μαντέψει τα διαπιστευτήρια του χρήστη.
- Εφαρμογές με φόρμες εγγραφής: Ο εισβολέας μπορεί να δημιουργήσει αυτόματα χιλιάδες νέους λογαριασμούς.
- Εφαρμογές με **φόρμες e-mail**: Ο επιτιθέμενος μπορεί να εκμεταλλευτεί τις φόρμες αιτήσεων ηλεκτρονικού ταχυδρομείου και να τις χρησιμοποιήσει για προώθηση ενοχλητικών μηνυμάτων (“spam relay”) ή για πλημμυρίδα (“flooding”) στο γραμματοκιβώτιο συγκεκριμένου χρήστη.
- **Συντήρηση λογαριασμού**: Ο επιτιθέμενος μπορεί να κάνει χρήση μαζικών επιθέσεων άρνησης της υπηρεσίας (DoS) στην εφαρμογή και να δημιουργήσει πλημμυρίδα με πολυάριθμα αιτήματα, με στόχο την απενεργοποίηση ή την διαγραφή των λογαριασμών των χρηστών.
- Εφαρμογές με **φόρμες στοιχείων λογαριασμού**: Ο επιτιθέμενος μπορεί να κάνει μαζικές προσπάθειες για να συλλέξει προσωπικές πληροφορίες των χρηστών από μια εφαρμογή ιστού
- Εφαρμογές με **φόρμες σχολίων ή υποβολής περιεχομένου**: Αυτές οι φόρμες μπορούν να χρησιμοποιηθούν σε spamming blogs, σε forums και σε πίνακες ανακοινώσεων στο διαδίκτυο με την αυτόματη υποβολή περιεχομένου, όπως spam ή και κακόβουλο λογισμικό (malware).
- Εφαρμογές με **φόρμες που σχετίζονται με ερωτήματα SQL βάσεων δεδομένων**: Αυτές οι φόρμες μπορούν να αξιοποιηθούν από τον επιτιθέμενο προκειμένου να εκτελέσει μια επίθεση άρνησης υπηρεσίας (DoS) στην εφαρμογή. Αυτό πραγματοποιείται με την αποστολή πολλών και απαιτητικών από άποψης επεξεργασίας ερωτημάτων SQL σε σύντομο χρονικό διάστημα, κάτι που έχει ως αποτέλεσμα την άρνηση των υπηρεσιών της εφαρμογής από χρήστες που έχουν πραγματική ανάγκη την υπηρεσία.
- Εφαρμογές **eShopping και ηλεκτρονικού εμπορίου**: Οι εν λόγω εφαρμογές μπορούν να χρησιμοποιηθούν από κακόβουλους χρήστες (scalpers) προκειμένου να αγοράσουν τα είδη που επιθυμούν σε μεγάλες ποσότητες, όπως για παράδειγμα εισιτήρια για αθλητικές εκδηλώσεις. Από τη στιγμή που τεχνητά δημιουργούν πρόβλημα επάρκειας, στη συνέχεια τα πωλούν σε υψηλότερες τιμές.
- Εφαρμογές για **online ψηφοφορίες/δημοσκοπήσεις**: Αυτές οι εφαρμογές μπορεί να χρησιμοποιηθούν κακόβουλα για να υπονομευθεί η διαδικασία υπέρ μιας συγκεκριμένης επιλογής.

- Εφαρμογές για **αποστολή μηνυμάτων (SMS)**: Ο επιτιθέμενος μπορεί να εκμεταλλευτεί το σύστημα αποστολής μηνυμάτων sms, προκειμένου να στείλει μαζικά μηνύματα (spam) σε χρήστες κινητών τηλεφώνων.

4.5.9. Ανεπαρκής αυθεντικοποίηση (Insufficient Authentication)

Η ευπάθεια της ανεπαρκούς αυθεντικοποίησης συμβαίνει, όταν μια ιστοσελίδα επιτρέπει σε έναν εισβολέα να αποκτήσει πρόσβαση σε ευαίσθητο περιεχόμενο ή στη λειτουργικότητα της ιστοσελίδας, χωρίς προηγουμένως να έχει προβεί σε σωστό και πλήρη έλεγχο της ταυτότητάς του. Τα εργαλεία διαχείρισης που είναι βασισμένα στο web (Web-based), είναι ένα καλό παράδειγμα των δικτυακών τόπων που παρέχουν πρόσβαση σε ευαίσθητες λειτουργίες. Ανάλογα με τον διαδικτυακό πόρο, αυτές οι εφαρμογές, δεν πρέπει να είναι άμεσα προσπελάσιμες χωρίς να απαιτείται από τον χρήστη ο έλεγχος της ταυτότητά του.

Ορισμένοι πόροι προστατεύονται με την απόκρυψη της θέσης των και δεν συνδέουν την θέση αυτή μέσω συνδέσμου με την κύρια ιστοσελίδα ή με άλλα δημόσια μέρη. Αυτή η προσέγγιση ασφαλείας ονομάζεται «ασφάλεια μέσω της συσκότισης». Είναι σημαντικό να γίνει κατανοητό, ότι ένας πόρος ακόμα και αν είναι άγνωστος σε έναν εισβολέα, αυτός παραμένει προσβάσιμος εάν δεχτεί απ' ευθείας κλήση από μια συγκεκριμένη διεύθυνση URL. Η συγκεκριμένη διεύθυνση URL θα μπορούσε να ανακαλυφθεί μέσα από μια επίθεση ωμής βίας σε ένα αρχείο το οποίο διαθέτει κοινότυπο όνομα και κοινότυπο φάκελο τοποθεσίας, όπως για παράδειγμα ο φάκελος "/admin" καθώς και οι φάκελοι αποθήκευσης μηνυμάτων λάθους, αρχείων καταγραφής, αρχείων βοηθείας κ.λπ. Οι πόροι αυτοί, θα πρέπει να προστατεύονται επαρκώς.

4.5.10. Ανεπαρκής εξουσιοδότηση (Insufficient Autorization)

Η ευπάθεια της ανεπαρκούς εξουσιοδότησης συμβαίνει, όταν μια εφαρμογή δεν διενεργεί επαρκείς ελέγχους εξουσιοδότησης για να εξασφαλίσει ότι ο χρήστης εκτελεί μια λειτουργία ή έχει πρόσβαση στα δεδομένα της εφαρμογής, κατά τρόπο σύμφωνο με την πολιτική ασφαλείας αυτής. Οι διαδικασίες εξουσιοδότησης θα πρέπει να εξασφαλίζουν ότι ο χρήστης, η υπηρεσία ή η εφαρμογή επιτρέπεται να χρησιμοποιεί ό,τι ορίζεται από την πολιτική ασφαλείας. Όταν ένας χρήστης πιστοποιείται σε μια ιστοσελίδα, αυτό δεν σημαίνει απαραίτητα ότι ο χρήστης θα πρέπει να έχει πλήρη πρόσβαση σε όλο το περιεχόμενο και τη λειτουργικότητα της σελίδας αυτής.

Πολλές εφαρμογές παρέχουν διαφορετικές λειτουργίες σε πολλούς τύπους χρηστών. Μια ειδησεογραφική ιστοσελίδα πρέπει να επιτρέπει στους χρήστες να διαβάσουν τις ειδήσεις, αλλά όχι και να δημοσιεύουν αυτές. Ένα λογιστικό σύστημα θα πρέπει να έχει διαφορετικά δικαιώματα πρόσβασης για χρήστες που επιθυμούν να πληρώσουν έναν λογαριασμό, από αυτούς που επιθυμούν να εισπράξουν.

4.5.11. Ανεπαρκής διαδικασία ανάκτησης συνθηματικών (Insufficient Password Recovery)

Η ευπάθεια ανεπαρκούς διαδικασίας ανάκτησης συνθηματικών (Insufficient Password Recovery) συμβαίνει όταν η ιστοσελίδα επιτρέπει σε έναν εισβολέα να αποκτήσει παράνομα, να αλλάξει ή να ανακτήσει τον κωδικό πρόσβασης ενός

τρίτου χρήστη. Οι συμβατικές μέθοδοι ελέγχου ταυτότητας των ιστοσελίδων απαιτούν από τους χρήστες να πληκτρολογούν και να θυμούνται τον κωδικό τους ή μια φράση πρόσβασης. Ο κωδικός αυτός απαιτείται να είναι μυστικός και απομνημονευμένος με ακρίβεια από αυτούς. Όμως η ικανότητα αυτής της απομνημόνευσης εξασθενίζει με την πάροδο του χρόνου. Βέβαια, το θέμα περιπλέκεται ακόμη περισσότερο όταν ο μέσος χρήστης επισκέπτεται περισσότερες της μίας ιστοσελίδες που απαιτούν από τον χρήστη την ίδια λειτουργία, δηλαδή την εισαγωγή του κωδικού πρόσβασης. Αυτό οδήγησε τους προγραμματιστές online εφαρμογών στην λειτουργία αυτόματης ανάκτησης του κωδικού των χρηστών μέσω ειδικής εφαρμογής στις ιστοσελίδες αυτές. Έρευνα έδειξε πως αυτή η λειτουργία είναι από τις πιο σημαντικές στην εξυπηρέτηση των online χρηστών. (RSA Έρευνα: HTTP://news.bbc.co.uk/1/hi/technology/3639679.stm).

Οι αυτοματοποιημένες διαδικασίες ανάκτησης κωδικού πρόσβασης απαιτούν από τον χρήστη να απαντήσει σε μια «μυστική ερώτηση» η οποία τέθηκε σε αυτόν κατά τη διαδικασία εγγραφής του. Η ερώτηση, επιλέχθηκε από τον χρήστη μέσα από μια λίστα έτοιμων ερωτήσεων ή από πληκτρολογήθηκε άμεσα από τον ίδιο. Ένας άλλος μηχανισμός που χρησιμοποιείται ευρέως, είναι η τεχνική κατά την οποία ο χρήστης παράσχει ένα "hint" κατά την εγγραφή του, που πιθανά να τον βοηθήσει μελλοντικά σε περίπτωση απώλειας του κωδικού του. Άλλοι μηχανισμοί ανάκτησης απαιτούν από το χρήστη να παράσχει πολλά στοιχεία των προσωπικών του δεδομένων, όπως ο αριθμός κοινωνικής ασφάλισης, διεύθυνση κατοικίας, κ.λπ. Στη συνέχεια, εφόσον ο χρήστης αποδείξει ποιος είναι, το σύστημα ανάκτησης θα αποστείλει στο e-mail του έναν νέο κωδικό πρόσβασης.

Θεωρούμε ότι ένας δικτυακός τόπος έχει μηχανισμό ανεπαρκούς ανάκτησης κωδικού πρόσβασης, όταν ο εισβολέας είναι σε θέση να τον εξουδετερώσει. Αυτό συμβαίνει όταν οι πληροφορίες που απαιτούνται για την επικύρωση ενός χρήστη μπορεί να προβλεφθούν ή να παρακαμφθούν. Τα συστήματα ανάκτησης κωδικού πρόσβασης μπορούν να τεθούν σε κίνδυνο με επιθέσεις ωμής βίας (brute Force).

4.5.12. Ανεπαρκής επικύρωση διαδικασιών (Insufficient Process Validation)

Η ευπάθεια της ανεπαρκούς επικύρωσης διαδικασιών συμβαίνει όταν μια εφαρμογή ιστού αποτυγχάνει να αποτρέψει έναν εισβολέα από το να παρακάμψει την προβλεπόμενη λογική ροή ή την λογική της εφαρμογής. Υπάρχουν δύο κύριοι τύποι των διαδικασιών που απαιτούν επικύρωση: Ο έλεγχος της ροής (flow control) και ο τύπος της επιχειρηματικής λογικής (business logic).

Ο **έλεγχος ροής** αναφέρεται σε μια διαδικασία που πραγματοποιείται βήμα προς βήμα και σε κάθε βήμα απαιτείται από τον χρήστη η εκτέλεση μίας συγκεκριμένης ενέργειας. Όταν ένας εισβολέας εκτελεί ένα βήμα λανθασμένα ή εκτός σειράς, ο έλεγχος πρόσβασης μπορεί να παρακαμφθεί και να εμφανιστεί ένα σφάλμα στην εφαρμογή. Μερικά παραδείγματα διαδικασίας πολλαπλών βημάτων, αποτελούν οι διαδικασίες ανάκτησης κωδικού πρόσβασης, η ολοκλήρωση μιας ηλεκτρονικής αγοράς, και η διαδικασία δημιουργίας λογαριασμού.

Η **Επιχειρηματική λογική**, αναφέρεται στο πλαίσιο εκείνο το οποίο μια διαδικασία θα εκτελεστεί, όπως διέπεται από τις επιχειρηματικές απαιτήσεις. Η αξιοποίηση των αδυναμιών της επιχειρηματικής λογικής απαιτεί γνώση της

επιχείρησης. Στην περίπτωση που ο επιτιθέμενος δεν χρειάζεται γνώση της επιχείρησης για να εκμεταλλευτεί την εφαρμογή, τότε κατά πάσα πιθανότητα αυτή να πάσχει από έλλειψη επιχειρηματικής λογικής. Για αυτό τον λόγο, πρέπει να λαμβάνονται τυπικά μέτρα ασφαλείας, όπως έλεγχοι και επιθεωρήσεις του κώδικα. Μια προσέγγιση δοκιμών αυτού του είδους προσφέρεται και στον οδηγό testing του OWASP.

4.5.13. Ανεπαρκής αυτόματη λήξη συνόδων (Insufficient Session Expiration)

Η ευπάθεια αυτή συμβαίνει όταν μια εφαρμογή ιστού επιτρέπει σε έναν εισβολέα να επαναχρησιμοποιήσει παλιά διαπιστευτήρια ή αναγνωριστικά συνόδου για την αυθεντικοποίηση του. Η ευπάθεια της ανεπαρκούς λήξης της συνόδου αυξάνει την έκθεση της εφαρμογής ιστού σε επιθέσεις που κλέβουν ή επαναχρησιμοποιούν τα αναγνωριστικά εισόδου των χρηστών.

Επειδή το HTTP είναι ένα πρωτόκολλο άνευ μνήμης (stateless)[19], οι τοποθεσίες ιστού χρησιμοποιούν τα cookies για να αποθηκεύουν αναγνωριστικά συνόδου, που προσδιορίζουν μονοσήμαντα τους χρήστες και τα αιτήματά τους. Κατά συνέπεια, κάθε αναγνωριστικό συνόδου πρέπει να είναι εμπιστευτικό και πρέπει να διατηρείται, προκειμένου να αποφευχθεί η πρόσβαση του ίδιου λογαριασμού από πολλούς χρήστες. Μια κλεμμένη ταυτότητα συνόδου μπορεί να χρησιμοποιηθεί από κακόβουλους χρήστες για την πρόσβαση τους σε λογαριασμούς άλλων χρηστών και την εκτέλεση παράνομων συναλλαγών.

Η λειτουργία λήξης της συνόδου έχει δύο τύπους:

- **Το χρονικό όριο αδράνειας (inactivity)**. Με τον όρο αυτό ορίζουμε ένα χρονικό όριο, ως μέγιστο επιτρεπτό χρόνο αδράνειας για κάθε σύνοδο πριν αυτή ακυρωθεί.
- **Το απόλυτο χρονικό όριο (absolute).** Ως απόλυτο χρονικό όριο ορίζουμε τον συνολικό χρόνο για τον οποίο επιτρέπεται μία σύνοδος να είναι έγκυρη, χωρίς να χρειαστεί η εκ νέου επαλήθευση της ταυτότητας του χρήστη.

Σε μια εφαρμογή ιστού, η έλλειψη της κατάλληλη λειτουργίας για την λήξη της συνεδρίας, μπορεί να αυξήσει την πιθανότητα επιτυχίας ορισμένων επιθέσεων. Ένα μεγάλο χρονικό διάστημα λήξης δίνει άφθονο χρόνο στον εισβολέα, για να βρει ένα έγκυρο αναγνωριστικό συνόδου. Εξάλλου, όσο μεγαλύτερο χρόνο λήξης δίνουμε σε μια συνεδρία τόσες περισσότερες ταυτόχρονες ανοικτές συνεδρίες υφίστανται σε κάθε δεδομένη στιγμή. Αυτό έχει σαν συνέπεια να υπάρχει μεγάλη ποσότητα ανοιχτών συνεδριών, που αυξάνει τις πιθανότητες του εισβολέα να βρει το πιστοποιητικό που ψάχνει, τυχαία.

Μια σωστή προσέγγιση με το θέμα των συνεδριών στις εφαρμογές ιστού είναι, αυτές να ακυρώνονται (timeout) μετά από ένα προκαθορισμένο χρονικό διάστημα αδράνειας και να παράσχουν στον χρήστη την δυνατότητα να ακυρώνουν τη δική τους συνεδρία όποτε αυτοί επιθυμούν (λειτουργία αποσύνδεσης). Η λειτουργία της αποσύνδεσης θα πρέπει να είναι εμφανώς ορατή στο χρήστη.

[19] HTTP://en.wikipedia.org/wiki/Stateless_protocol

4.5.14. Ανεπαρκής προστασία επιπέδου μεταφοράς (Insufficient Transport Layer Protection)

Η ευπάθεια αυτή επιτρέπει την έκθεση των πληροφοριών κατά τη διάρκεια της επικοινωνίας χρήστη - ιστοσελίδας, σε τρίτους, μη έμπιστους χρήστες, παρέχοντας στον επιτιθέμενο την δυνατότητα να υποβιβάσει την ασφάλεια μιας εφαρμογής ιστού και την κλοπή ευαίσθητων πληροφοριών. Οι ιστοσελίδες συνήθως χρησιμοποιούν Secure Socket Layer (SSL) και Transport Layer Security (TLS) για να παρέχουν την απαραίτητη κρυπτογράφηση στο επίπεδο της μεταφοράς. Ωστόσο, οι ιστοσελίδες μπορεί να είναι ευάλωτες σε υποκλοπές κυκλοφορίας και τροποποίησης, στην περίπτωση που οι ρυθμίσεις τους για τη χρήση SSL/TLS δεν έχει γίνει σωστά.

Έλλειψη κρυπτογράφησης στο επίπεδο μεταφοράς. Όταν το επίπεδο μεταφοράς δεν είναι κρυπτογραφημένο, όλη η επικοινωνία μεταξύ της ιστοσελίδας και του χρήστη αποστέλλεται σε απλό, μη κρυπτογραφημένο κείμενο. Αυτό σημαίνει πως η επικοινωνία είναι ευάλωτη επιθέσεις υποκλοπής, έγχυσης (injection) και ανακατεύθυνσης (redirection). Ένας εισβολέας έχει τη δυνατότητα να παρακολουθεί παθητικά την επικοινωνία και να έχει πρόσβαση στα ευαίσθητα δεδομένα που μεταδίδονται, όπως ονόματα χρηστών και κωδικούς πρόσβασης. Επίσης μπορεί να συμμετέχει ενεργά στην συναλλαγή με αφαίρεση ή τροποποίηση του μεταδιδόμενου περιεχομένου, την έγχυση κακόβουλων scripts ή την ώθηση του χρήστη να προσπελάσει απομακρυσμένο μη αξιόπιστο περιεχόμενο. Μπορεί ακόμα να ανακατευθύνει την επικοινωνία με τέτοιο τρόπο ώστε η ιστοσελίδα και ο χρήστης, να μην επικοινωνούν μεταξύ τους, αλλά εν αγνοία τους, με τον εισβολέα.

Χρήση αδύναμων αλγόριθμων κρυπτογράφησης. Παλιότερα, η υψηλής ποιότητας κρυπτογραφία απαγορευόταν να εξαχθεί εκτός Ηνωμένων Πολιτειών Αμερικής. Για αυτό τον λόγο οι δικτυακοί τόποι έχουν ρυθμιστεί να υποστηρίζουν αδύναμες κρυπτογραφικές επιλογές για τους χρήστες που υφίσταντο τον περιορισμό αυτού του είδους. Η χρήση αδύναμων αλγόριθμων κρυπτογράφησης όμως, είναι ευάλωτη σε επιθέσεις, λόγω της σχετικά εύκολης παραβίασης που μπορεί να γίνει κατά την χρήσης τους.

Σήμερα, όλα τα σύγχρονα προγράμματα περιήγησης και ιστοσελίδες χρησιμοποιούν πολύ ισχυρότερη κρυπτογράφηση, αλλά μερικές ιστοσελίδες εξακολουθούν να έχουν τέτοιες ρυθμίσεις, ώστε να εξυπηρετούν τους αδύναμους αλγόριθμους κρυπτογράφησης. Εξαιτίας αυτού, ένας εισβολέας μπορεί να είναι σε θέση να εξαναγκάσει τον χρήστη σε υποβάθμιση κρυπτογράφησης, σε μορφή ασθενέστερη κατά τη σύνδεση του με την ιστοσελίδα, κάτι που του επιτρέπει να την «σπάσει» με σχετικά εύκολο τρόπο. Για τον λόγο αυτό, ο εξυπηρέτης θα πρέπει να ρυθμιστεί με τέτοιο τρόπο, ώστε να δέχεται μόνο ισχυρούς αλγόριθμους κρυπτογράφησης και να μην παρέχει υπηρεσίες σε χρήστες που επιθυμούν να κάνουν χρήση ασθενέστερων. Από την άλλη μεριά, ορισμένες ιστοσελίδες που δεν είναι σωστά ρυθμισμένες, επιλέγουν και κάνουν χρήση, αδύναμων αλγόριθμων κρυπτογράφησης, ακόμη και όταν ο πελάτης μπορεί να υποστηρίξει πολύ ισχυρότερους. Το OWASP προσφέρει έναν οδηγό για τις δοκιμές σε θέματα SSL/TLS, συμπεριλαμβανομένων της αδύναμης και της εσφαλμένης υποστήριξης κρυπτογράφησης.

Χρήση μεικτού περιεχομένου. Υπάρχουν εφαρμογές ιστού οι οποίες εξυπηρετούν ιστοσελίδες που ενώ χρησιμοποιούν προστασία σε επίπεδο μεταφοράς (https), στη συνέχεια, συμπεριλαμβάνουν επιπλέον περιεχόμενο, όπως JavaScript ή εικόνες μέσω μη κρυπτογραφημένου HTTP. Αυτές οι ιστοσελίδες χρησιμοποιούν μεικτό περιεχόμενο και είναι ευάλωτες σε επιθέσεις. Ένας εισβολέας θα μπορούσε να αντικαταστήσει το νόμιμο JavaScript με κακόβουλο. Εύκολα μπορούμε να φανταστούμε τις συνέπειες που θα είχε ο χρήστης στην περίπτωση που η εφαρμογή πλοήγησής του εκτελέσει το Javascript αυτό, στο πλαίσιο της σελίδας HTTPS.

Για όλους τους παραπάνω λόγους, όλο το περιεχόμενο μιας ασφαλούς σελίδας, είτε είναι HTML, είτε JavaScript, εικόνες, CSS, XHR, καθώς και οποιοδήποτε άλλο περιεχόμενο, θα πρέπει να εξυπηρετείται μέσω HTTPS.

4.5.15. Επισφαλής διαμόρφωση εξυπηρέτη (Server Misconfiguration)

Υπάρχουν επιθέσεις που επιτυγχάνουν λόγω της εκμετάλλευσης κάποιων αδυναμιών ή τεχνικών κακού προγραμματισμού που μπορεί να έχει ο εξυπηρέτης. Πολλά πακέτα διανομής, έχουν περιττά αρχεία εργοστασιακής προεπιλογής, καθώς και αρχεία εκμάθησης, συμπεριλαμβανομένων εφαρμογών, αρχεία ρυθμίσεων, scripts, και ιστοσελίδων αναφοράς. Επίσης, έχουν περιττές υπηρεσίες ενεργοποιημένες, όπως η διαχείριση περιεχομένου και η λειτουργία απομακρυσμένης διαχείρισης. Σε κάποιες περιπτώσεις έχουν ενεργοποιημένες λειτουργίες εντοπισμού σφαλμάτων (debugging), αλλά και λειτουργίες διαχειριστή προσβάσιμες από απλούς και ανώνυμους χρήστες. Όλα αυτά τα χαρακτηριστικά, μπορεί να παράσχουν στον επιτιθέμενο τα εργαλεία και τον τρόπο, να παρακάμψει τις μεθόδους ελέγχου ταυτότητας και να αποκτήσει πρόσβαση σε ευαίσθητες πληροφορίες, με αυξημένα προνόμια.

Οι εξυπηρέτες μπορεί να περιλαμβάνουν από προεπιλογή, γνωστούς λογαριασμούς και κωδικούς πρόσβασης καθώς και να έχουν ρυθμιστεί με εσφαλμένα δικαιώματα αρχείων και καταλόγων. Να έχουν εσφαλμένα πιστοποιητικά SSL, προεπιλεγμένα πιστοποιητικά, ρυθμίσεις κρυπτογράφησης, καθώς και εσφαλμένη εφαρμογή ελέγχου ταυτότητας με εξωτερικά συστήματα. Όλα τα παραπάνω μπορεί να θέσουν σε κίνδυνο την εμπιστευτικότητα των πληροφοριών.

Τα μηνύματα λάθους που περιλαμβάνουν λεπτομέρειες του συστήματος μπορεί να οδηγήσουν σε διαρροή δεδομένων, και στην αποκάλυψη πληροφοριών που θα μπορούσαν να χρησιμοποιηθούν για να διαμορφώσει το πλαίσιο της επόμενης επίθεσης.

5. Επισκόπηση διαθέσιμων εργαλείων για έλεγχο ευπαθειών

5.1. Εισαγωγή

Τα εργαλεία τα οποία διαθέτουμε και εν τέλει επιλέγουμε για να πραγματοποιήσουμε μια δοκιμή διείσδυσης σε ένα υπολογιστικό σύστημα ή σε ένα δίκτυο είναι το πιο ουσιαστικό μέρος για την εκτέλεση μιας δοκιμής. Με την βοήθεια τους ο ελεγκτής είναι δυνατό να εντοπίσει και στη συνέχεια να επιδιορθώσει τα τρωτά σημεία που πιθανά να αποκαλυφθούν. Υπάρχουν πολλά διαθέσιμα εργαλεία ελέγχου τρωτότητας, ανοιχτού κώδικα και μη, που εξυπηρετούν διαφορετικούς σκοπούς.

Τον όρο *εργαλεία* μπορούμε να τον ορίσουμε με πολλούς τρόπους. Σε γενικές γραμμές όμως, εργαλείο είναι οτιδήποτε μπορεί να χρησιμοποιηθεί για να εκτελέσει μια αυτοματοποιημένη λειτουργία. Συγκεκριμένα μπορεί να είναι τυποποιημένες εφαρμογές (applications), βοηθητικά προγράμματα (utilities), scripts, προγράμματα ειδικού σκοπού (special-purpose programs) και πρωτόκολλα τα οποία έχουν σχεδιαστεί ειδικά, ώστε να ωθούν ένα σύστημα στα όρια του και να διαμορφώνουν ακραίες συνθήκες, έτσι ώστε να εκμεταλλευθούμε τις αδυναμίες του. Τα εργαλεία, στο πλαίσιο της εκτέλεσης μιας δοκιμής, είναι σχεδιασμένα να συλλέγουν τις απαραίτητες πληροφορίες, να κάνουν επιθέσεις και γενικά να εκτελούν συγκεκριμένες εργασίες με απώτερο σκοπό την αναγνώριση ή και την εκμετάλλευση μιας ευπάθειας.

5.2. Εργαλεία για έλεγχο ευπαθειών

Τα εργαλεία ελέγχου ευπαθειών, χωρίζονται σε αρκετές κατηγορίες, οι οποίες φυσικά δεν είναι δεσμευτικές για κάθε εργαλείο, αλλά αποτελούν την κύρια λειτουργία του καθενός. Τα περισσότερα εργαλεία μπορούν να επιτελέσουν περισσότερους του ενός ελέγχους κάθε φορά, αλλά συνίσταται ισχυρά η εξειδίκευση των εφαρμογών που χρησιμοποιούνται για την αντιμετώπιση του κάθε προβλήματος ξεχωριστά. Έτσι, τόσο η ανακάλυψη αδυναμιών, όσο και η πρόταση λύσεων, από τα εργαλεία, είναι πιο έγκυρη και πιο κατευθυντική για τον διαχειριστή που συντηρεί και ασφαλίζει κάποιο σύστημα.

Τα χαρακτηριστικά των σημαντικότερων από αυτά καθώς και η κατηγορία στην οποία αυτά ανήκουν, παρουσιάζονται και ταξινομούνται, παρακάτω:

5.3. Εργαλεία ανίχνευσης Ευπαθειών (Vulnerability Scanners)

Είναι εργαλεία τα οποία χρησιμοποιούνται για την ανίχνευση αδυναμιών, την αξιολόγηση δικτύων, συστημάτων υπολογιστών και εφαρμογών. Υπάρχουν διάφοροι τύποι εργαλείων ανίχνευσης ευπαθειών και διακρίνονται μεταξύ τους από την εστίαση τους σε συγκεκριμένους στόχους. Αυτά τα εργαλεία συνήθως χρησιμοποιούνται για τη διεξαγωγή σάρωσης και αναγνώρισης ενός δικτύου. Ο στόχος είναι ο προσδιορισμός των ενεργών συσκευών που απαρτίζουν το δίκτυο και η συγκέντρωση πληροφοριών για τις συσκευές αυτές, όπως το είδος και η έκδοση του λειτουργικού συστήματος. Αυτές οι πληροφορίες μπορούν να αναλυθούν περεταίρω για τον εντοπισμό πιθανών τρωτών σημείων και για προσπάθεια απόκτησης πρόσβασης σε αυτά.

Μερικά αντιπροσωπευτικά προγράμματα της κατηγορίας αυτής παρατίθενται παρακάτω:

5.3.1. Nessus

Πρόκειται για ένα από τα πιο δημοφιλή προγράμματα της κατηγορίας αυτής, κυρίως για συστήματα UNIX. Αρχικά ήταν δωρεάν και ανοιχτού κώδικα (open source), στη συνέχεια όμως η εταιρεία «έκλεισε» τον πηγαίο κώδικα. Σήμερα η εταιρεία παρέχει δωρεάν έκδοση του προγράμματος, με περιορισμένες όμως λειτουργίες για οικιακή χρήση. Το Nessus ενημερώνεται διαρκώς και διαθέτει περισσότερα από 55000 πρόσθετα (plugins).

Το πρόγραμμα αρχικά ξεκινά μια επίθεση ανίχνευσης θυρών (port scan attack), προκειμένου να ελέγξει πιθανές ανοιχτές πόρτες της δικτυακής συσκευής που στοχεύει. Στη συνέχεια προσπαθεί να βρει το λειτουργικό σύστημα της συσκευής και την ύπαρξη πιθανών ενημερώσεων ασφαλείας όπως security patches, service packs, security updates. Κατόπιν, χρησιμοποιεί exploits στις ανοιχτές θύρες για να διαπιστώσει τη δυνατότητα απόκτησης πρόσβασης στο σύστημα. Τέλος, εκδίδει αναφορά η οποία περιλαμβάνει τα αποτελέσματα των ερευνών του. Το πρόγραμμα διαθέτει επιλογή μέσω της οποίας μπορεί να επιτεθεί σε ένα σύστημα, μέχρι αυτό να καταλήξει σε άρνηση υπηρεσίας (DoS) ή ακόμα και σε κατάρρευση (crash). Εκατοντάδες exploits ανακαλύπτονται ημερησίως τα οποία διατίθενται προς διάθεση στο κοινό από την παρασκευάστρια εταιρεία, μετά από επτά ημέρες.

5.3.2. Core Impact

Το λογισμικό αυτό είναι ένα από τα πιο ισχυρά εργαλεία της κατηγορίας και ίσως το πιο ακριβό λογισμικό του είδους, γενικά. Το κόστος του ανεβαίνει ανάλογα με τις επιπλέον λειτουργίες που χρειάζεται ο χρήστης. Το λογισμικό προσπαθεί να αντικαταστήσει τον παράγοντα άνθρωπο με αυτοματοποιημένες λειτουργίες, μέσω των οποίων επιτίθεται σε ένα υπολογιστικό σύστημα και στην συνέχεια το εκμεταλλεύεται, δημιουργώντας ένα ασφαλές κανάλι (“secure tunnel”) προκειμένου να επιτεθεί στο επόμενο σύστημα, ενώ ταυτόχρονα φροντίζει να ασφαλίζει τα exploits που έχει χρησιμοποιήσει και να κλειδώνει τα backdoors που βρίσκει ανοιχτά στην πορεία του.

Μερικές από τις σημαντικές λειτουργίες του προγράμματος είναι:

- Ο καθορισμός της ευαισθησίας των χρηστών σε επιθέσεις που πραγματοποιούνται μέσα από κοινωνικά δίκτυα και αξιολόγηση της συνολικής ασφάλειας των συστημάτων.
- Η δυνατότητα πραγματοποίησης δοκιμών διείσδυσης σε κινητές έξυπνες συσκευές (smartphones) με λειτουργικά συστήματα όπως Ios,

Android, Blackberry και απόδειξη πιθανών ευπαθειών που υπάρχουν σε αυτές.

- Η δοκιμή διείσδυσης σε συσκευές που ανήκουν σε ένα δίκτυο.
- Η δοκιμή διείσδυσης στις ενεργές συσκευές ενός δικτύου, εκμεταλλευόμενο το λειτουργικό σύστημα με το οποίο λειτουργούν και ανακαλύπτοντας τα τρωτά τους σημεία.
- Η δοκιμή ανάκτησης κωδικών πρόσβασης, από hashes κωδικών πρόσβασης.
- Οι δοκιμές διείσδυσης που ανιχνεύουν διάφορες αδυναμίες που περιγράφονται και καθορίζονται από το OWASP Top 10.
- Οι δοκιμές διείσδυσης σε ασύρματα δίκτυα.
- Οι δοκιμές διείσδυσης σε δικτυακές συσκευές όπως IPS/IDS και Firewalls.

5.3.3. OpenVAS

Το συγκεκριμένο λογισμικό ανήκει και αυτό στην κατηγορία ανίχνευσης ευπαθειών, είναι ανοιχτού κώδικα, και διανέμεται δωρεάν με άδεια χρήσης GNUGPL. Παίζει σημαντικό ρόλο στο πεδίο της ασφάλειας των πληροφοριακών συστημάτων και ίσως είναι το πιο προηγμένο εργαλείο ανοικτού κώδικα για τη διαχείριση ευπαθειών. Μπορεί να εντοπίσει κενά ασφαλείας σε έναν υπολογιστή, σε ένα δίκτυο ακόμα και σε εφαρμογές. Ενημερώνεται καθημερινά και ο αριθμός των ελέγχων (NVTs) αριθμείται σε πάνω από 30.000 (Απρίλιος 2013).

Το OpenVAS διαθέτει διάφορες υπηρεσίες και εργαλεία. Ο πυρήνας αυτής της αρχιτεκτονικής είναι ο σαρωτής OpenVAS (OpenVAS Scanner). Ο σαρωτής εκτελεί τις πραγματικές δοκιμές ευπάθειας δικτύου (NTVs) ενάντια στις συσκευές – στόχους.

Ο διαχειριστής OpenVAS (OpenVAS manager) είναι η κεντρική υπηρεσία η οποία ενοποιεί τις δυνατότητες του OpenVAS και τον μετατρέπει από έναν απλό διαχειριστή ευπάθειας σε μία πλήρη και ολοκληρωμένη λύση. Επίσης, ο διαχειριστής ελέγχει και τη βάση δεδομένων (sqlite-based), στην οποία αποθηκεύονται κεντρικά όλα τα στοιχεία που προκύπτουν από την σάρωση.

5.4. Εργαλεία ανίχνευσης και εκμετάλλευσης ευπαθειών (Vulnerability Exploitation Tools)

Στην κατηγορία αυτή ανήκουν τα εργαλεία που έχουν ως στόχο την ανακάλυψη και εκμετάλλευση ενδεχόμενων αδυναμιών σε δικτυακές συσκευές και υπολογιστικά συστήματα τα οποία απαρτίζουν ένα πληροφοριακό σύστημα.

5.4.1. Metasploit

Η συγκεκριμένη εφαρμογή έχει ως στόχο την εκτέλεση κακόβουλου κώδικα σε υπολογιστικά συστήματα. Παρέχει στον χρήστη πρόσβαση σε βάση δεδομένων η οποία περιέχει πολλά γνωστά σφάλματα (bugs) για όλα τα λειτουργικά συστήματα. Ο χρήστης, παραμετροποιώντας την εφαρμογή, έχει τη δυνατότητα να εκμεταλλευτεί ένα

σύστημα και να ελέγξει εάν είναι ευάλωτο σε συγκεκριμένα exploits. Στη συνέχεια και σε περίπτωση επιτυχούς πρόσβασης, μπορεί να δώσει τον κώδικα που ο ίδιος επιθυμεί για να εκτελεστεί από το μηχάνημα-στόχο περιμένοντας τα αποτελέσματα. Έχει δε την δυνατότητα να τροποποιήσει και να κωδικοποιήσει τον κώδικα σε διάφορες μορφές, προκειμένου να μην γίνει αντιληπτός από συστήματα IDS.

5.4.2. Social Engineer Toolkit

Είναι ένα εργαλείο κοινωνικής μηχανικής (Social Engineer[20]) ειδικά σχεδιασμένο για να πραγματοποιεί εξειδικευμένες επιθέσεις ενάντια στο ανθρώπινο στοιχείο. Είναι αυτοματοποιημένο και ευέλικτο, ενσωματώνει πολλές από τις λειτουργίες που βρίσκουμε σε άλλα διαθέσιμα προγράμματα του είδους και θεωρείται ένα από τα καλύτερα του είδους του. Μπορεί να δημιουργήσει αυτόματα exploits, κρυμμένα σε ιστοσελίδες ή σε μηνύματα ηλεκτρονικού ταχυδρομείου χρησιμοποιώντας πληροφορίες από βάσεις δεδομένων metasploit. Μπορεί να πραγματοποιήσει επιθέσεις ηλεκτρονικού ψαρέματος (phishing) με τη μέθοδο της κλωνοποίησης ενός δικτυακού τόπου, προκειμένου ο ελεγκτής να λάβει το όνομα χρήση και τους κωδικούς πρόσβασης με τους οποίους κάποιος χρήστης εισέρχεται σε ένα σύστημα.

Βρίσκεται δωρεάν από τον δικτυακό τόπο HTTP:/www.social-engineer.org καθώς και στην διανομή ελέγχου τρωτότητας backtrack.

5.4.3. WebGoat

Είναι μια εσκεμμένα ανασφαλής εφαρμογή βασισμένη σε java, η οποία υποστηρίζεται από το OWASP και είναι σχεδιασμένη με τρόπο ώστε να διδάσκει μαθήματα ασφαλείας για εφαρμογές ιστού. Σε κάθε μάθημα οι χρήστες πρέπει να κατανοήσουν ένα ζήτημα ασφαλείας, αξιοποιώντας ένα πραγματικό ζήτημα ευπάθειας στην εφαρμογή.

Ο πρωταρχικός στόχος της εφαρμογής είναι να δημιουργήσει ένα de-facto διαδραστικό περιβάλλον διδασκαλίας για την ασφάλεια των εφαρμογών ιστού. Ένας από τους μελλοντικούς σκοπούς της ομάδας των προγραμματιστών του WebGoat είναι η επέκταση της εφαρμογής, ώστε να γίνει μια πλατφόρμα αξιολόγησης της ασφάλειας εφαρμογών ιστού, βασισμένη σε java.

5.5. Εργαλεία καταγραφής πακέτων (Packet Analyzer Sniffers)

Στην κατηγορία αυτή ανήκουν προγράμματα τα οποία παρακολουθούν ένα δίκτυο, υποκλέπτοντας πακέτα τα οποία μπορεί να περιέχουν σημαντικές

[20] Το Social Engineer (κοινωνική μηχανική) είναι η πράξη της χειραγώγησης ατόμων με σκοπό την απόσπαση πληροφοριών και έχει τις ρίζες του στον χάκερ και αργότερα σύμβουλο ασφαλείας πληροφορικών συστημάτων Κέβιν Μίτνικ ο οποίος διέδωσε αυτό τον όρο επισημαίνοντας ότι είναι πολύ ευκολότερο να ξεγελάσεις κάποιον να δώσει έναν κωδικό πρόσβασης για ένα σύστημα από το να προσπαθήσεις να τον σπάσεις.

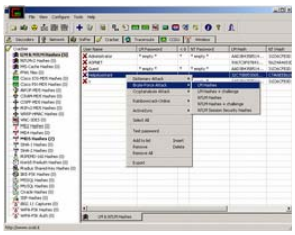
πληροφορίες του χρήστη. Συχνά αποτελούν πολύ μεγάλο κίνδυνο, ιδιαίτερα αν υπάρχει ευάλωτη στην κρυπτανάλυση ή και καθόλου κρυπτογράφηση και στην ανταλλαγή των δεδομένων περιλαμβάνονται συνθηματικά.

5.5.1. Wireshark

Η εφαρμογή αυτή είναι ανοιχτού κώδικα και είναι γνωστή από παλαιότερα και ως Ethereal. Είναι ένας αναλυτής πακέτων (packet analyzer) που έχει ως στόχο την εύρεση εκείνων των πληροφοριών που είναι χρήσιμα στον χειριστή. Μπορεί να διαβάζει και να αναλύει πακέτα δεδομένων τα οποία μεταδίδονται σε πραγματικό χρόνο στο δίκτυο, ή είναι αποθηκευμένα σε κατάλληλο αρχείο (capture). Επιτρέπει στον χρήστη να επεξεργαστεί τα συγκεντρωμένα δεδομένα εισχωρώντας στο επιθυμητό επίπεδο λεπτομερειών που αυτός κάθε φορά επιθυμεί. Υποστηρίζει εκατοντάδες πρωτόκολλα καθώς και εγκατάσταση νέων, με τη λογική των plugins. Ένα από τα πολλά χαρακτηριστικά του είναι, η δυνατότητα μετατροπής των δεδομένων που συλλέγει και αποθηκεύει, με σκοπό την επεξεργασία τους από άλλα προγράμματα.

5.5.2. Cain and Abel

Είναι το πιο ολοκληρωμένο εργαλείο για το λειτουργικό σύστημα Windows της Microsoft, με πάρα πολλές δυνατότητες τόσο σε ασύρματα όσο σε ενσύρματα δίκτυα. Μπορεί να πραγματοποιήσει υποκλοπή λογαριασμών και κωδικών πρόσβασης, καθώς και να παρακολουθήσει την κίνηση σε ένα δίκτυο με καταγραφή και αναφορές σε XML και HTML. Αρχικά, σαρώνει το δίκτυο στο οποίο βρίσκεται, και παρουσιάζει όλες τις ενεργές συσκευές που ανήκουν σε αυτό. Κατόπιν, με κατάλληλες τεχνικές, επιτυγχάνει την καταγραφή όλων των δρομολογημένων πακέτων του δικτύου και μετά από κάποια ώρα συλλογής στοιχείων μπορεί να παρουσιάσει ονόματα χρηστών και συνθηματικά, κλειδιά ασύρματων δικτύων καθώς και κάνει ανάλυση συνομιλιών VoIP[21]. Σε περίπτωση που τα συνθηματικά, ή οι μεταδιδόμενες πληροφορίες είναι κρυπτογραφημένες, είναι εφοδιασμένο με password cracker, το οποίο με χρήση των κατάλληλων αλγορίθμων μπορεί να πραγματοποιήσει και την κατάλληλη αποκρυπτογράφηση.



Η ανάπτυξη του έγινε με την προοπτική να είναι χρήσιμο για τους διαχειριστές δικτύων, εκπαιδευτικούς, συμβούλους ασφαλείας, ελεγκτές ασφαλείας και οποιονδήποτε άλλον που δύναται να το χρησιμοποιήσει για ηθικούς σκοπούς.

5.5.3. Tcpdump

Είναι ένας κοινός αναλυτής πακέτων ο οποίος λειτουργεί άνευ παραθυρικού περιβάλλοντος, με γραμμή εντολών. Η

[21] Voice over IP: Μια ομάδα πρωτοκόλλων – τεχνολογιών, η οποία προσφέρει φωνητική συνομιλία σε πραγματικό χρόνο πάνω από δίκτυα IP (over Internet Protocol) όπως το διαδίκτυο.

συνηθέστερη χρήση του είναι για πραγματοποίηση αποσφαλμάτωσης (dubugging) εφαρμογών που χρησιμοποιούν πρωτόκολλα για να επικοινωνούν με άλλα συστήματα, με σκοπό να βρεθούν πιθανά προβλήματα και να απομονωθούν. Επίσης μπορεί να χρησιμοποιηθεί για την παρακολούθηση συνδέσεων σε μη ασφαλή πρωτόκολλα και μεθόδους επικοινωνίας όπως είναι το HTTP και το TELNET, υποκλέπτοντας ονόματα χρηστών, κωδικούς πρόσβασης και διάφορα άλλα ευαίσθητα δεδομένα.

Ανήκει στην κατηγορία του λογισμικού ανοιχτού κώδικα και διανέμεται σύμφωνα με την άδεια BSD.

5.5.4. Ettercap

Είναι μια ολοκληρωμένη σουίτα, εξειδικευμένη σε επιθέσεις τύπου man in the middle. Λειτουργεί με μεγάλο αριθμό πρωτοκόλλων και μπορεί να εκτελέσει πληθώρα λειτουργιών όπως ωτακοή (sniffing) σε πρωτόκολλα SSH, HTTPS, SSL, sniffing σε GRE Tunnel, OS Fingerprint, συλλογή κωδικών πρόσβασης, ενεργών υπηρεσιών και εφαρμογών. Επίσης υπάρχει η δυνατότητα να κάνει παρεμβολή και να διακόψει την επικοινωνία μεταξύ hosts καθώς και να παρέμβει σε αιτήματα DNS.

Ανήκει στην κατηγορία του λογισμικού ανοιχτού κώδικα και διανέμεται δωρεάν.

5.6. Εργαλεία καταγραφής και παραμετροποίησης πακέτων (Packet Crafting Tools)

Είναι εργαλεία που επιτρέπουν στους ελεγκτές δικτύων να εξετάσουν το σύνολο των κανόνων που απαρτίζουν μια συσκευή firewall, προκειμένου να ανακαλύψουν σημεία εισόδου στο σύστημα ή το δίκτυο. Αυτό μπορεί να επιτευχθεί, με εργαλεία τα οποία εμπεριέχουν γεννήτριες και αναλυτές πακέτων και δημιουργούν την κατάλληλη κίνηση προκειμένου να ελέγξουν τη "συμπεριφορά" των συσκευών του ελεγχόμενου δικτύου.

Οι δικτυακές συσκευές που στοχεύουν τα εργαλεία αυτά μπορεί να είναι είτε το firewall του δικτύου, είτε το IDS, η στοίβα TCP/IP, ο δρομολογητής (Router), ο μεταγωγέας (Switch) και οποιαδήποτε άλλη ενεργή δικτυακή συσκευή.

5.6.1. Netcat

Είναι μια απλή εφαρμογή, η οποία διαβάζει και γράφει δεδομένα στις συνδέσεις ενός δικτύου χρησιμοποιώντας τα κατάλληλα πακέτα, είτε αυτά είναι TCP είτε UDP. Περιέχει πλούσια χαρακτηριστικά αποσφαλμάτωσης (debugging) καθώς και εργαλείο έρευνας και μπορεί να δημιουργήσει σχεδόν οποιοδήποτε είδος σύνδεσης που θα χρειαστεί ο χρήστης. Στον κατάλογο των χαρακτηριστικών του περιλαμβάνονται λειτουργίες όπως port scanning, μεταφορά αρχείων και ακρόαση θυρών (port listening). Μπορεί να χρησιμοποιηθεί ως κερκόπορτα (backdoor) ενός δικτύου και για αυτούς τους λόγους συχνά αποκαλείτε ως ο «Ελβετικός σουγιάς για το TCP/IP».

5.6.2. Hping

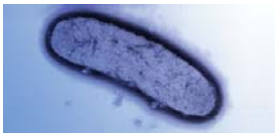
Είναι ένας αναλυτής πακέτων TCP/IP (packet assembler/analyzer) ο οποίος δεν διαθέτει παραθυρικό περιβάλλον αλλά λειτουργεί μέσα από γραμμή εντολών (command line). Είναι σε θέση να στείλει τροποποιημένα αιτήματα πακέτων ICMP echo, UDP και TCP και να παρουσιάσει τις απαντήσεις. Ενσωματώνει την λειτουργία traceroute με εύχρηστο τρόπο και κατακερματισμό πακέτων IP (IP fragmentation). Είναι ιδιαίτερα χρήσιμο σε χρήστες που προσπαθούν να πραγματοποιήσουν traceroute ή ping πίσω από ένα firewall το οποίο μπλοκάρει αυτές τις προσπάθειες τους.

Συνήθως, χρησιμοποιείται από τους ελεγκτές δικτύων για να πραγματοποιήσουν χαρτογράφηση ενός δικτύου αλλά και για την καταγραφή του συνόλου των κανόνων του τείχους προστασίας αυτού. Επίσης, προσφέρεται για εκμάθηση και πειραματισμό του πρωτοκόλλου TCP/IP.

Είναι ανοικτού κώδικα και μπορεί να το βρει κανείς από την επίσημη ιστοσελίδα με άδεια GPL. Δυστυχώς έχει σταματήσει η ανάπτυξη και η ενημέρωση του από το 2005.

5.6.3. Yersinia

Είναι ένα χαμηλού επιπέδου εργαλείο επίθεσης, χρήσιμο για τεχνικές ελέγχου διείσδυσης, το οποίο έχει σχεδιαστεί με σκοπό να εκμεταλλευτεί τις αδυναμίες που παρουσιάζονται στα πρωτόκολλα των δικτύων. Μπορεί να πραγματοποιήσει πολλών ειδών διαφορετικές επιθέσεις μέσω διαφορετικών πρωτοκόλλων, όπως, το να αναλάβει τον ρόλο της “root” συσκευής σε ένα σενάριο Spanning tree[22] (Spanning Tree Protocol), να δημιουργήσει εικονικούς γείτονες στο CDP[23] (Cisco Discovery Protocol), να γίνει η ενεργή συσκευή σε ένα HSRP[24] (Hot standby Routing Protocol) κάνοντας παραποίηση αιτημάτων DHCP, καθώς και άλλων χαμηλού επιπέδου επιθέσεων.



[22] **Spanning Tree**: Είναι ένα Layer 2 πρωτόκολλο που έχει γίνει αποδεκτό από τον οργανισμό IEEE ως το πρότυπο 802.1d. Η υπηρεσία αυτή, τρέχει σε δικτυακές συσκευές όπως routers και switches και ο κύριος σκοπός της είναι να εξασφαλίσει ότι δεν δημιουργούνται διπλοί δρόμοι (loops) όταν υπάρχουν παραπάνω από μια συνδέσεις στις συσκευές αυτές. Αξίζει να σημειωθεί ότι τα loops θεωρούνται «θανατηφόρα» σε ένα δίκτυο.

[23] **Cisco Discovery Protocol**: Είναι ένα Layer 2 πρωτόκολλο εντοπισμού συσκευών, το οποίο τρέχει σε όλες τις συσκευές της εταιρείας cisco. Χρησιμοποιώντας το, μια συσκευή μπορεί να «διαφημίσει» την ύπαρξη της προς άλλες συσκευές (και αντίστροφα) καθώς και να στέλνει πληροφορίες για την κατάσταση της συσκευής, την IP διεύθυνση, την θύρα διασύνδεσης και άλλα.

[24] **Hot Standby Routing Protocol**: Είναι ένα Layer 2 πρωτόκολλο το οποίο χρησιμοποιείται από δικτυακές συσκευές για να επιτευχθεί σχεδόν 100% χρόνος λειτουργικότητας (up-time). Παρέχει δικτυακή εφεδρεία σε ip δίκτυα, διασφαλίζοντας ότι η κίνηση θα αναδρομολογηθεί σε περίπτωση αστοχίας υλικού. Πρακτικά με την τεχνική αυτή, χρησιμοποιούμε 2 συσκευές με κοινή ip και MAC διεύθυνση οι οποίες λειτουργούν σε εφεδρεία. Σε κάθε περίπτωση, η μια συσκευή λειτουργεί ως ενεργή (active) και η άλλη λειτουργεί σε κατάσταση αναμονής (preempt). Το HSRP αναλαμβάνει όταν η ενεργή συσκευή αστοχήσει να θέσει σε λειτουργία την συσκευή αναμονής.

5.7. Εργαλεία σπασίματος συνθηματικών (Password crackers)

Στην κρυπτανάλυση και στην επιστήμη της ασφάλειας των υπολογιστών, το σπάσιμο των συνθηματικών (password cracking) είναι η διαδικασία της ανάκτησης των κωδικών πρόσβασης από δεδομένα τα οποία έχουν αποθηκευτεί ή μεταδίδονται από ένα υπολογιστικό σύστημα. Τα προγράμματα τα οποία χρησιμοποιούν διάφορες τεχνικές και μεθόδους για την αποκρυπτογράφηση των κωδικών των χρηστών, τα κατατάσσουμε στην κατηγορία αυτή.

5.7.1. Aircrack

Είναι μια σουίτα εργαλείων η οποία χρησιμοποιείται από ελεγκτές δικτύων για την αποκωδικοποίηση αλγορίθμων WEP και WPA που χρησιμοποιούνται σε δίκτυα 802.11/a/b/g[25]. Υλοποιεί τους καλύτερους γνωστούς αλγόριθμους για να ανακτήσει τα κλειδιά ασύρματων δικτύων μετά από την παρακολούθηση και συγκέντρωση αρκετών κρυπτογραφημένων πακέτων.

Η σουίτα **aircrack-ng** περιλαμβάνει πολλά διακριτά εργαλεία ελέγχου ασύρματων δικτύων, συμπεριλαμβανομένων των **airodump** (πρόγραμμα σύλληψης πακέτων), το **aireplay** (πρόγραμμα για έκχυση πακέτων), το **aircrack** (πρόγραμμα για «σπάσιμο» πρωτοκόλλων ασφαλείας WEP, WPA-PSK) και το **airdecap** (πρόγραμμα αποκρυπτογράφησης αρχείων καταγραφής WEP/ WPA).

Ανήκει στην κατηγορία του λογισμικού ανοιχτού κώδικα και διανέμεται σύμφωνα με την άδεια GPL v2, BSD 3 Clause και Open SSL.

5.7.2. THC Hydra

Η πρώτη έκδοση του λογισμικού δημιουργήθηκε πριν από περίπου δέκα χρόνια από μια γερμανική ομάδα που ονομαζόταν “The Hacker’s Choice”. Η ομάδα χρησιμοποίησε την μέθοδο της «επίθεσης λεξικού» (dictionary attack), για το σπάσιμο ασθενών συνθηματικών. Είχε απλοϊκή σχεδίαση και ο κύριος στόχος της ήταν να δείξει στο κοινό, πόσο σημαντική είναι η παράλειψη της χρήσης ισχυρών κωδικών.

Σήμερα, το λογισμικό αυτό χρησιμοποιείται σε περιπτώσεις που χρειάζεται να κάνουμε επιθέσεις ωμής βίας για να παραβιάσουμε υπηρεσίες που απαιτούν απομακρυσμένο έλεγχο ταυτότητας. Μπορεί να εκτελέσει ταχείες επιθέσεις τύπου λεξικού (dictionary attack), με χρήση πάρα πολλών πρωτοκόλλων, όπως, telnet, ftp, http, https, http-proxy, mysql, rlogin, snmp, vnc, pop3, imap, icq, ldap, cisco enable, cisco aaa κ.ά.

Το λογισμικό διατίθεται δωρεάν από την ιστοσελίδα της THC Hydra σύμφωνα με την άδεια, GPLv3.

[25]Το 802.11/a/b/g ανήκει στην 802.11 οικογένεια προτύπων της IEEE, για ασύρματα τοπικά δίκτυα (WLAN), τα οποία είναι ευρύτερα γνωστά ως WiFi. Ο αρχικός σκοπός ήταν επέκταση τους 802.3 (Ethernet) πρωτοκόλλου της ενσύρματης δικτύωσης, στην ασύρματη περιοχή.

5.7.3. Medusa

Το λογισμικό Medusa προορίζεται για χρήστες ή για ελεγκτές δικτύων που επιθυμούν να πραγματοποιήσουν ταχείες, παράλληλες και αρθρωτές επιθέσεις ωμής βίας. Ο στόχος του είναι οι υπηρεσίες εκείνες που επιτρέπουν απομακρυσμένη ταυτοποίηση. Είναι ένα εργαλείο που η χρήση του γίνεται από γραμμή εντολών και υποστηρίζει πολλά πρωτόκολλα όπως ftp, http, imap, ssh, ssh v2, mysql, pop3, snmp, telnet, vnc κ.ά.

Τα πιο βασικά χαρακτηριστικά της εφαρμογής είναι:

- Η δοκιμή ωμής βίας (brute-force), η οποία μπορεί να εκτελεστεί παράλληλα από πολλούς υπολογιστές και πολλούς χρήστες με πολλούς κωδικούς ασφαλείας, ταυτόχρονα.
- Η ευέλικτη είσοδος χρήστη, με την οποία στοχευμένη πληροφορία μπορεί να σταλεί με πολλούς τρόπους. Για παράδειγμα, κάθε στοιχείο εισόδου, μπορεί να είναι είτε μια ενιαία καταχώρηση είτε ένα αρχείο που περιέχει πολλαπλές εισόδους είτε συνδυασμός αυτών.
- Ο αρθρωτός σχεδιασμός. Κάθε υπηρεσία υπάρχει ως ανεξάρτητο αρχείο (.mod file). Αυτό σημαίνει ότι για την επέκταση της λίστας των υποστηριζόμενων υπηρεσιών δεν τροποποιούμε τον κορμό της εφαρμογής.

Είναι λογισμικό ανοιχτού κώδικα και διανέμεται δωρεάν.

5.7.4. RainbowCrack

Το RainbowCrack βασίζεται στην θεωρία κρυπτανάλυσης του Philippe Oechslin.

Το πρόγραμμα μπορεί να βρει κωδικούς που βασίζονται στους αλγόριθμους κερματισμού LM, MD5 και SHA1. Ανήκει στην νέα γενιά προγραμμάτων εύρεσης κωδικών πρόσβασης και έχει γίνει ευρέως γνωστό για την ταχύτητα με την οποία οι κωδικοί πρόσβασης οι οποίοι είναι κρυπτογραφημένοι με αυτούς τους αλγόριθμους, μπορούν να ανακαλυφθούν.

Για την λειτουργία του το πρόγραμμα χρησιμοποιεί ειδικού τύπου πίνακες (Rainbow tables), οι οποίοι λαμβάνουν υπόψιν το αντιστάθμισμα χρόνου – μνήμης (time-memory trade off). Με την μέθοδο αυτή γίνεται η εύρεση του μη κρυπτογραφημένου κωδικού από ένα κρυπτογραφημένο ο οποίος παράγεται από μια μονόδρομη συνάρτηση hash και έτσι μειώνεται αισθητά ο χρόνος που απαιτείται. Η διαδικασία υπολογισμού των πινάκων συνήθως παίρνει ένα απαιτούμενο χρονικό διάστημα, αλλά από την στιγμή που θα δημιουργηθούν οι πίνακες, με τη μέθοδο που προαναφέρθηκε, η εύρεση είναι εκατοντάδες φορές ταχύτερη από την παραδοσιακή μέθοδο της εξαντλητικής αναζήτησης, την οποία χρησιμοποιούν άλλα προγράμματα του είδους.

5.8. Εργαλεία ανίχνευσης ευπαθειών σε εξυπηρέτες ιστοσελίδων (Web Vulnerability Scanners)

Τα εργαλεία της κατηγορίας είναι πολύ δημοφιλή σήμερα, επειδή οι εφαρμογές ιστού επιτρέπουν στους χρήστες να έχουν μια διαδραστική εμπειρία στο διαδίκτυο και με αυτό τον τρόπο εκθέτουν τα προσωπικά τους δεδομένα δημόσια. Αυτές οι εφαρμογές δεν προβάλουν μόνο στατικές σελίδες, αλλά είναι σε θέση να επιτρέψουν στον χρήστη να δημιουργήσει προσωπικούς λογαριασμούς, να προσθέσει περιεχόμενο, να κάνει ερωτήματα σε βάσεις δεδομένων καθώς και να πραγματοποιήσει κάθε είδους συναλλαγή. Οι σύγχρονες ιστοσελίδες, κατά τη διαδικασία παροχής των υπηρεσιών τους συχνά συλλέγουν, αποθηκεύουν και χρησιμοποιούν προσωπικά ή και ευαίσθητα προσωπικά δεδομένα. Οι χρήστες με την σειρά τους επωφελούνται από την ευκολία των εφαρμογών αυτών και παραβλέπουν τον κίνδυνο που ενέχει η διάθεση των προσωπικών τους δεδομένων στις εφαρμογές αυτές, είτε από επιθέσεις χάκερ είτε από χρησιμοποίηση των πληροφοριών αυτών από τις ίδιες τις εταιρείες.

Οι προγραμματιστές των εφαρμογών αυτών, χρησιμοποιούν τα εργαλεία της κατηγορίας και έχουν ως σκοπό την εύρεση πιθανών προβλημάτων ασφαλείας, αρχιτεκτονικών αδυναμιών και exploits εφαρμογών ιστού. Από τις βασικές λειτουργίες τους είναι η διερεύνηση για πιθανές ανοιχτές πόρτες, οι οποίες δεν θα έπρεπε να είναι ανοιχτές, για ευάλωτες σελίδες που φιλοξενούνται στο εξυπηρέτη, καθώς και η προσπάθεια για την εκμετάλλευσης αυτών προς όφελος του επιτιθέμενου.

5.8.1. W3af

Το W3af είναι ένα εξαιρετικά δημοφιλές, ισχυρό και ευέλικτο πλαίσιο που έχει ως στόχο την εξεύρεση και εκμετάλλευση των τρωτών σημείων των εφαρμογών ιστού, προκειμένου ο χρήστης να δημιουργήσει μια ασφαλή εφαρμογή ή ο ελεγκτής να προτείνει λύσεις για να γίνει ασφαλής μια εφαρμογή. Παρέχει σαρωτή ευπαθειών καθώς και εργαλείο εκμετάλλευσης των ευπαθειών εφαρμογών ιστού. Επίσης, κατά τις δοκιμές διείσδυσης σε μια εφαρμογή, το πρόγραμμα παρέχει στον ελεγκτή χρήσιμες πληροφορίες σχετικά με τις ευπάθειες των εφαρμογών. Είναι εύκολο στη χρήση, επεκτείνεται συνεχώς και διαθέτει δεκάδες plugins εκμετάλλευσης.

Η εφαρμογή αναπτύχθηκε με τη χρήση Python και είναι ανοιχτού κώδικα σύμφωνα με άδεια GPLv2.0. Είναι διαθέσιμη για όλα τα δημοφιλή λειτουργικά συστήματα όπως Windows, Linux, Mac OS X, FreeBSD, OpenBSD και ο χρήστης έχει την δυνατότητα να λειτουργήσει το πρόγραμμα είτε σε γραφικό περιβάλλον είτε σε γραμμή εντολών.

5.8.2. WebScarab

Το Webscarab είναι ένα πλαίσιο (framework) για την ανάλυση εφαρμογών ιστού, οι οποίες επικοινωνούν χρησιμοποιώντας HTTP και HTTPS πρωτόκολλα. Είναι γραμμένο σε γλώσσα Java και για αυτό τον λόγο μπορεί να λειτουργήσει σε πολλές πλατφόρμες. Η εφαρμογή λειτουργεί με μια σειρά από

διάφορα plugins τα οποία προσφέρουν συγκεκριμένες λειτουργίες σε αυτό. Στην πιο συνηθισμένη του χρήση λειτουργεί ως intercepting proxy, επιτρέποντας στον χρήστη να επανεξετάσει και να τροποποιήσει τα αιτήματα που δημιουργούνται από την εφαρμογή ιστού του χρήστη πριν από την αποστολή των αιτημάτων στον εξυπηρέτη, καθώς και να επανεξετάσει και να τροποποιήσει τις απαντήσεις του εξυπηρέτη, προτού αυτές ληφθούν από την εφαρμογή ιστού. Επίσης, είναι σε θέση να παρακολουθήσει τις επικοινωνίες HTTP και HTTPS και να τα επανεξετάσει τα αιτήματα και τις απαντήσεις που περνούν από την εφαρμογή, αν το επιθυμεί αυτό ο χρήστης.

Το εργαλείο αυτό έχει σχεδιαστεί με σκοπό να χρησιμοποιηθεί από χρήστες οι οποίοι έχουν γνώσεις προγραμματισμού. Μπορεί να αποκαλύψει τον τρόπο λειτουργίας μιας εφαρμογής ιστού και να επιτρέψει στον υπεύθυνο ασφαλείας να προσδιορίσει τις αδυναμίες μιας εφαρμογή καθώς και να προτείνει διορθώσεις στον υπεύθυνο ανάπτυξης.

5.8.3. Skipfish

Το Skipfish, είναι μια εφαρμογή ασφαλείας με αυτοματοποιημένες λειτουργίες η οποία έχει σχεδιαστεί για να αναγνωρίζει και να ανακαλύπτει πιθανά τρωτά σημεία σε εφαρμογές ιστού. Είναι εύκολη στη χρήση, ανιχνεύει αυτόματα και με εξαιρετική ταχύτητα τις εφαρμογές ιστού για προβλήματα ασφαλείας. Επίσης, ερευνά για την ύπαρξη κενών ασφαλείας σε διάφορα επίπεδα, όπως είναι οι υπερχειλίσεις μνήμης.

Το εργαλείο είναι ανοιχτού κώδικα, ο οποίος υπάρχει διαθέσιμος στο google code και εκδοχές της εφαρμογής υπάρχουν για πολλές πλατφόρμες όπως, Linux, FreeBSD, Windows (Cygwin) και Mac OS X.

5.8.4. Netsparker

Είναι ένας σαρωτής ασφαλείας εφαρμογών ιστού για την εύρεση και την εκμετάλλευση πιθανών αδυναμιών σε εφαρμογές. Το Netsparker είναι ένα χρήσιμο εργαλείο τόσο για προγραμματιστές, όσο για δοκιμαστές ασφαλείας εφαρμογών ιστού (penetration testers). Παρέχει εύχρηστο περιβάλλον χρήστη, ταχύτατη διαδικασία σάρωσης των εφαρμογών και προσφέρει πλήρη υποστήριξη για εφαρμογές AJAX και Javascript.

Είναι το μοναδικό λογισμικό του είδους το οποίο χρησιμοποιεί ενσωματωμένη μηχανή εκμετάλλευσης των αδυναμιών, αφήνοντας χρόνο στον ελεγκτή να χρησιμοποιήσει τις δεξιότητες του για την διόρθωση των λαθών και όχι για την αναζήτηση αυτών. Δηλαδή, παρέχει τις ακριβείς πληροφορίες που χρειάζονται για να εργαστούν οι ελεγκτές γρήγορα και αποτελεσματικά.

Εκτός από την αυτόματη αναζήτηση αδυναμιών, το λογισμικό επιτρέπει και την χειροκίνητη αναζήτηση, δίνοντας την δυνατότητα στον ελεγκτή να δει τις πραγματικές επιπτώσεις μιας επίθεσης με πολύ απλό τρόπο.

5.8.5. Firebug

Το Firebug είναι ένα εργαλείο για την ανάπτυξη ιστοσελίδων, που διευκολύνει τον εντοπισμό, την επεξεργασία και την αποσφαλμάτωση σε πραγματικό χρόνο των

CSS, Javascript κ.ά. των ιστοσελίδων που μελετάμε με αυτό. Με το λογισμικό αυτό, οι προγραμματιστές εφαρμογών ιστού έχουν την δυνατότητα να κατανοήσουν τι συμβαίνει στην ιστοσελίδα που παρακολουθούν ή κατασκευάζουν, τον χρόνο που απαιτείται για την φόρτωση της, τα υπάρχοντα λάθη, τις κλήσεις των συναρτήσεων από την εφαρμογή πλοήγησης και τις αιτήσεις URL, όπως π.χ. για ανάκτηση εξωτερικών CSS και αρχείων εικόνας.

Εκτός από τον εντοπισμό σφαλμάτων και την ανάλυση των επιδόσεων των ιστοσελίδων, το Firebug είναι χρήσιμο εργαλείο για τον έλεγχο ασφαλείας εφαρμογών ιστού. Το λογισμικό γράφτηκε από έναν από τους δημιουργούς του Firefox και η ομάδα εργασίας που επιβλέπει την περεταίρω ανάπτυξη του διαθέτει δυο μεγάλες υλοποιήσεις. Την επέκταση (add-on) για το Mozilla Firefox, καθώς και μια απλουστευμένη έκδοση (Firebug lite) για άλλες εφαρμογές πλοήγησης.

Το λογισμικό είναι ανοιχτού κώδικα και διανέμεται δωρεάν με άδεια BSD.

5.9. Rootkit Detectors

Ένα *rootkit*, είναι λογισμικό που επιτρέπει την συνεχή πρόσβαση σε έναν υπολογιστή με προνόμια υπερχρήστη, ενώ κρύβει ενεργά την παρουσία του από τους διαχειριστές με το να ενσωματώνεται σε βασικά αρχεία του λειτουργικού συστήματος ή άλλων εφαρμογών. Ο όρος rootkit είναι μια συνένωση των λέξεων "root" (το παραδοσιακό όνομα του λογαριασμού διαχειριστή σε λειτουργικά συστήματα τύπου Unix) και της λέξης "kit".

Τυπικά, ένας εισβολέας εγκαθιστά ένα rootkit σε έναν υπολογιστή μόλις αποκτήσει πρόσβαση σε επίπεδο υπερχρήστη, είτε με την αξιοποίηση γνωστών κενών στην ασφάλεια στου λειτουργικού είτε με την απόκτηση ενός κωδικού πρόσβασης (είτε με απευθείας επίθεση στην κρυπτογράφηση, είτε μέσω της κοινωνικής μηχανικής). Ένα rootkit συνήθως ενσωματώνεται σε κάποιο από τα βασικά αρχεία του λειτουργικού συστήματος και με αυτόν τον τρόπο αποκτά τον πλήρη έλεγχο όλου του συστήματος και επομένως και τον έλεγχο του οποιουδήποτε αντιϊκού λογισμικού.

Τα rootkit detectors είναι λογισμικά τα οποία ανιχνεύουν τα rootkits. Η ανίχνευση ενός rootkit είναι δύσκολη επειδή μπορεί να είναι σε θέση να αλλάξει ακόμα και το λογισμικό που προορίζεται για την εύρεσή του. Οι μέθοδοι ανίχνευσης περιλαμβάνουν τη χρήση ενός αναπληρωματικού αξιόπιστου λειτουργικού συστήματος, τη χρήση σύγκρισης των αρχείων του λειτουργικού με βάση αποθηκευμένα έμπιστα αντίγραφα ή τον έλεγχο υπογραφών των αρχείων των συστημάτων τα οποία, όμως, πρέπει να έχουν ληφθεί πριν την «επίθεση».

Πιο συγκεκριμένα, μερικά αντιπροσωπευτικά προγράμματα της κατηγορίας αυτής παρατίθενται παρακάτω:

5.9.1. Sysinternals

Το λογισμικό αυτό, εμπεριέχει πολλά μικρά βοηθητικά προγράμματα των Windows, τα οποία είναι πολύ χρήσιμα για έλεγχο ασφαλείας χαμηλού επιπέδου.

Κάποια από αυτά παρέχονται ελεύθερα και περιλαμβάνουν τον πηγαίο τους κώδικα, ενώ άλλα παρέχονται με κάποιο κόστος. Τα ποιο διαδεδομένα από αυτά είναι:

- **ProcessExplorer:** Βοηθητικό πρόγραμμα που επιτρέπει να βλέπουμε τα ανοιχτά αρχεία, τους καταλόγους καθώς και τις υπηρεσίες (processes) που εκτελούνται.
- **PsTools:** Μας επιτρέπει να διαχειριστούμε (εκτέλεση, αναστολή, κλείσιμο) τοπικές και απομακρυσμένες υπηρεσίες.
- **Autoruns:** Μας επιτρέπει να ανακαλύψουμε ποιες υπηρεσίες είναι προγραμματισμένες να εκτελεστούν κατά τη διαδικασία εκκίνησης (boot up) του συστήματος και κατά την διαδικασία της εισόδου (login) στο σύστημα.
- **RootkitRevealer:** Το πρόγραμμα αυτό ανιχνεύει το μητρώο και τα API's του συστήματος για τυχόν αποκλίσεις, οι οποίες μπορούν να υποδεικνύουν την παρουσία rootkit.
- **TCPView**: Για την παρατήρηση των παραμέτρων της κίνησης TCP και UDP που χρησιμοποιείται από κάθε διαδικασία, όπως για παράδειγμα το Netstat στο Unix.

Από την εποχή που το Sysinternals υπήρχε μόνο για Linux, σε πολλά από τα εργαλεία, δόθηκε και ο πηγαίος τους κώδικας. Μετέπειτα, όταν το απέκτησε η Microsoft αφαιρέθηκε το μεγαλύτερο μέρος του κώδικα, παρά τις διαβεβαιώσεις για το αντίθετο.

5.9.2. HijackThis

Το λογισμικό επιθεωρεί τις ρυθμίσεις του λειτουργικού συστήματος καθώς και την εφαρμογή πλοήγησης του υπολογιστή για εσφαλμένες ή ύποπτες καταχωρήσεις

στο μητρώο και στον σκληρό δίσκο και δημιουργεί ένα αρχείο καταγραφής με την τρέχουσα κατάσταση. Στη συνέχεια, αυτοματοποιημένα εργαλεία του λογισμικού, αναλύουν τις παραμέτρους του αρχείου καταγραφής, το συγκρίνουν με προεπιλεγμένες ρυθμίσεις και κάνουν τις κατάλληλες συστάσεις προς τον χρήστη, ή τον ρωτούν εάν επιθυμεί το λογισμικό να καθαρίσει αυτόματα τις λάθος καταχωρήσεις.

Η χρήση των εργαλείων αυτών, πρέπει να αποφεύγεται από άπειρους χρήστες γιατί ένα αρχείο καταγραφής δεν είναι και τόσο εύκολο να αναλυθεί. Τα αποτελέσματα τέτοιων εργαλείων, θεωρούνται επικίνδυνα, μη ακριβή και γενικά όχι αξιόπιστα και για αυτούς τους λόγους τα αρχεία αυτά θα πρέπει να αναλύονται από εκπαιδευμένους αναλυτές.

Πρόκειται για μια εφαρμογή ανοιχτού κώδικα η οποία διανέμεται δωρεάν από την εταιρεία TrendMicro.

6. Παρουσίαση του τρόπου λειτουργίας του Webscarab και αναφορά των πιθανών ευπαθειών κατά τον έλεγχο στους εξυπηρέτες του Τμήματος Κοινωνικής & Εκπαιδευτικής Πολιτικής

6.1. Παρουσίαση του WebScarab

Το WebScarab είναι ένα εξελισσόμενο project υπό την αιγίδα του OWASP. Ο στόχος του είναι να παράσχει ένα ελεύθερο εργαλείο στους υπεύθυνους ανάπτυξης και τους επιθεωρητές εφαρμογών ιστού, ώστε να κατανοήσουν τη λειτουργία των εφαρμογών του συγκεκριμένου είδους και να αναγνωρίσουν τα πιθανά προβλήματα, τα οποία θα μπορούσαν να προκαλέσουν τη δυσλειτουργία των εφαρμογών αυτών. Το WebScarab λειτουργεί υπό την άδεια του GNU General Public License v2.

Όπως προαναφέρθηκε, το Webscarab είναι ένα πλαίσιο (framework) για την ανάλυση εφαρμογών ιστού, οι οποίες επικοινωνούν χρησιμοποιώντας τα πρωτόκολλα HTTP και HTTPS. Είναι γραμμένο σε γλώσσα Java και για αυτό τον λόγο μπορεί να λειτουργήσει σε πολλές πλατφόρμες. Η εφαρμογή λειτουργεί με μια σειρά από διάφορα **πρόσθετα (plugins)** τα οποία προσφέρουν συγκεκριμένες λειτουργίες σε αυτό. Στην πιο συνηθισμένη του χρήση λειτουργεί ως intercepting proxy, επιτρέποντας στον χρήστη να επανεξετάσει και να τροποποιήσει τα αιτήματα που δημιουργούνται από την εφαρμογή ιστού που χρησιμοποιεί πριν από την αποστολή των αιτημάτων στον εξυπηρέτη, καθώς και να επανεξετάσει και να τροποποιήσει τις απαντήσεις του εξυπηρέτη, προτού αυτές ληφθούν από την εφαρμογή ιστού. Επίσης, είναι σε θέση να παρακολουθεί τις επικοινωνίες HTTP και HTTPS και να τα επανεξετάζει τα αιτήματα και τις απαντήσεις που περνούν από την εφαρμογή, αν το επιθυμεί ο χρήστης.

6.2. Εγκατάσταση

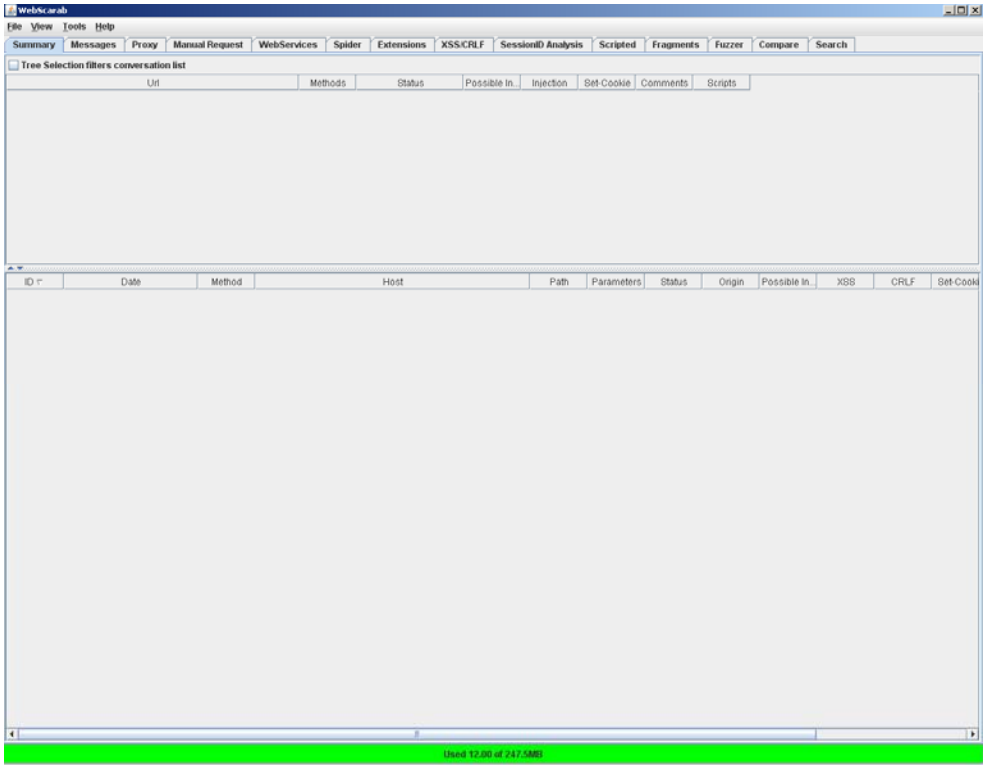
Την εφαρμογή μπορούμε να την εγκαταστήσουμε σε υπολογιστές με λειτουργικό σύστημα Windows ή Linux ή MacOS ή σε οποιοδήποτε λειτουργικό που υποστηρίζει Java. Για να εκτελεστεί χρειάζεται Java JRE (Java Runtime Environment) αρκεί να είναι μεταγενέστερη της έκδοσης 1.4.

Μπορούμε να μεταφορτώσουμε την εφαρμογή από τη ιστοσελίδα του οργανισμού OWASP, ή απευθείας από την ιστοσελίδα HTTP://sourceforge.net/project/showfiles.php?group_id=64424&package_id=61823 επιλέγοντας το αρχείο “webscarab-installer-20070504-1631.jar”. Εφόσον κάνουμε την λήψη του αρχείου, εγκαθιστάμε την εφαρμογή και μπορούμε πλέον να το χρησιμοποιήσουμε.

6.3. Ρυθμίσεις περιβάλλοντος εργασίας

Το Webscarab χαρακτηρίζεται από την πολυλειτουργικότητα του και ως εκ τούτου, ένας νέος χρήστης πρέπει να δώσει προσοχή σε κάποια βασικά σημεία του.

Στην (Εικόνα 4) φαίνεται το περιβάλλον της εφαρμογής κατά την εκκίνηση της.



Εικόνα 4. Η οθόνη εκκίνησης του Webscarab

Παρακάτω, σημειώνονται μερικές περιοχές που αξίζει να σταθούμε και να επεξηγήσουμε. Η γραμμή εργαλείων παρέχει πρόσβαση στα παράθυρα:

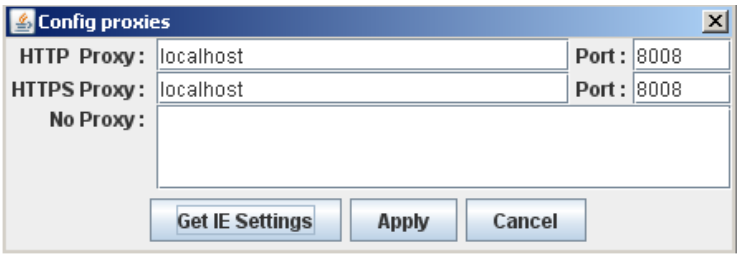
- της κύριας όψης (“summary”),
- των μηνυμάτων καταγραφής (“log messages”),
- και στα διάφορα πρόσθετα της εφαρμογής (“plugins”).

Το παράθυρο της κύριας όψης χωρίζεται σε δυο μέρη. Στο πάνω μέρος βρίσκεται ένας πίνακας μορφής δέντρου που δείχνει τις ιστοσελίδες που επισκεπτόμαστε καθώς και ορισμένα χαρακτηριστικά των URL, ενώ στο κάτω μέρος βρίσκεται ένας πίνακας που δείχνει όλες τι συνομιλίες που περνούν μέσα από την εφαρμογή.

Προκειμένου να ξεκινήσουμε να χρησιμοποιούμε την εφαρμογή πρέπει να ρυθμίσουμε την εφαρμογή ιστού που χρησιμοποιούμε να λειτουργεί με το Webscarab ως εξυπηρέτης αντιπροσώπευσης (upstream proxy server).

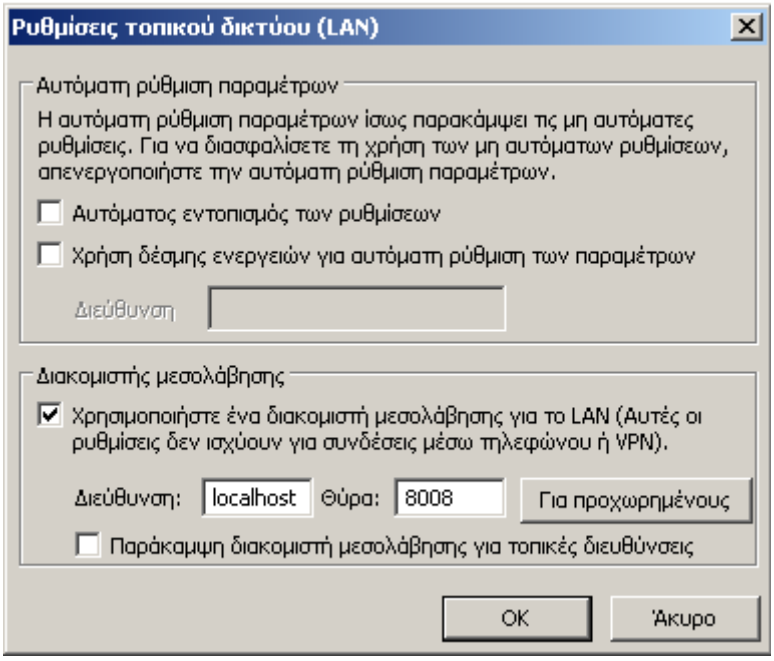
6.3.1. Ρυθμίσεις εξυπηρέτη αντιπροσώπευσης (Upstream Proxy)

Για τους χρήστες του λειτουργικού συστήματος Windows, το WebScarab υποστηρίζει την αυτόματη εισαγωγή των ρυθμίσεων του εξυπηρέτη αντιπροσώπευσης (proxy server) που χρησιμοποιεί ο Internet Explorer, μέσω του κατάλληλου πλαισίου διαλόγου. Αυτό το χαρακτηριστικό γνώρισμα απαιτεί ένα JNI plugin/DLL, το οποίο πρέπει να περιλαμβάνεται στη μεταβλητή PATH. Εάν το αρχείο DLL βρίσκεται στη σωστή θέση, τότε μια επιλογή με τίτλο "Get IE Settings", θα εμφανιστεί όπως βλέπουμε και στην (Εικόνα 5). Εάν το αρχείο DLL δεν υπάρχει ή δεν βρίσκεται στη σωστή θέση/κατάλογο, τότε η επιλογή αυτή δεν θα εμφανιστεί.



Εικόνα 5. Καρτέλα διαμόρφωσης του proxy μέσα από το περιβάλλον του Webscarab

Σε περίπτωση που στο δίκτυο μας δεν χρησιμοποιούμε proxy server, όπως στην περίπτωση των δοκιμών που έγιναν στα πλαίσια της πτυχιακής αυτής εργασίας, πρέπει να χρησιμοποιήσουμε ως προεπιλογή τη θύρα 8008 στο localhost[26] για την ρύθμιση του proxy server, διότι η θύρα 8008 είναι αυτή η οποία χρησιμοποιεί το webscarab. Έτσι, πρέπει να διαμορφώσουμε κατάλληλα τον Internet Explorer ούτως ώστε να προωθεί τα αιτήματα προς εξυπηρέτες ιστού στο Webscarab. Για να πραγματοποιηθεί αυτό πρέπει να μεταβούμε στα «εργαλεία→ Επιλογές Internet→ Συνδέσεις→ Ρυθμίσεις LAN. Στη συνέχεια να επιλέξουμε τη χρήση proxy server (Εικόνα 6) και να δηλώσουμε ως διεύθυνση το localhost και θύρα την 8008. Αντίστοιχες ρυθμίσεις μπορούν να γίνουν και για άλλες εφαρμογές πλοήγησης (Firefox, Chrome κ.λπ.).



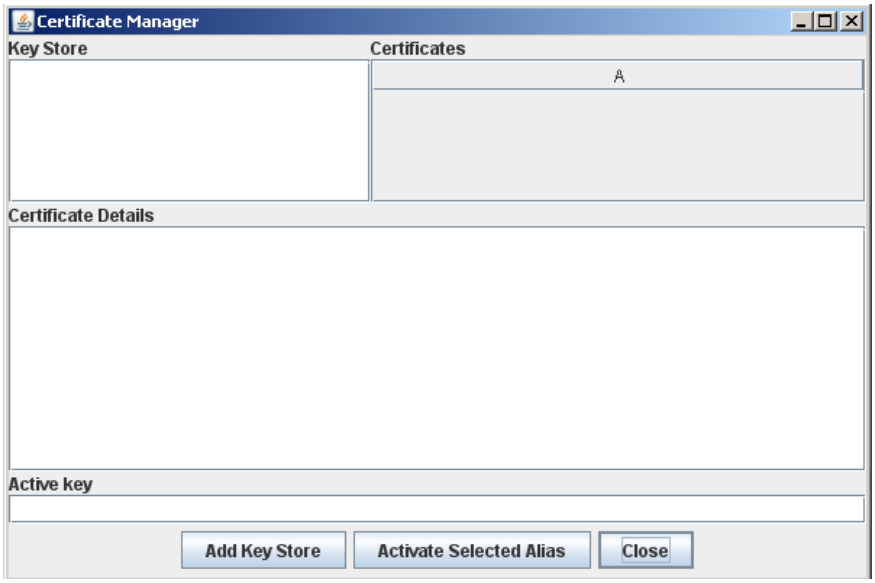
Εικόνα 6. Διαμόρφωση ρυθμίσεων του τοπικού δικτύου και χρήση proxy, από το περιβάλλον του Internet Explorer

Όταν ολοκληρώσουμε τις απαραίτητες ρυθμίσεις ανοίγουμε την εφαρμογή και καλούμε μια ιστοσελίδα. Εάν όλα έχουν ρυθμιστεί σωστά, θα παρατηρήσουμε τις συνομιλίες να δρομολογούνται μέσα από τον proxy του Webscarab.

[26] Το όνομα **localhost** χρησιμοποιείται για να δηλώσει τον ίδιο υπολογιστή. Αντιστοιχεί στη διεύθυνση IPv4: 127.0.0.1 και IPv6: ::1 και συνήθως χρησιμοποιείται για loopback testing.

6.3.2. Πιστοποιητικά πελάτη (Client-side Certificates)

Το WebScarab παρέχει υποστήριξη για πιστοποιητικά SSL πελατών. Προς το παρόν, μπορεί να έχει πρόσβαση σε κλειδιά και πιστοποιητικά, που αποθηκεύονται σε μορφή αρχείων PKCS#12, καθώς επίσης και PKCS#11 (από την έκδοση της 1.5 της Java και μετέπειτα).



Εικόνα 7. Καρτέλα διαχειριστή πιστοποιητικών μέσα από το περιβάλλον του Webscarab

Για να κάνουμε χρήση του πιστοποιητικού, πρέπει ανοίξουμε την επιλογή ”certificate manager”, να ορίσουμε τη θέση που βρίσκεται το πιστοποιητικό και να δώσουμε έναν κωδικό για την μετέπειτα πρόσβαση μας σε αυτό. Η λειτουργία αυτή επιτυγχάνεται μέσω της επιλογής Tools → Certificates. Στη συνέχεια, για να ενεργοποιήσουμε το πιστοποιητικό που θέλουμε, το επιλέγουμε και κάνουμε κλικ στο “Activate selected Alias”.

6.4. Διαχείριση συνόδων από το Webscarab (Webscarab session management)

Η εφαρμογή χρησιμοποιεί την έννοια της συνόδου (session) για να συσχετίσει τις διάφορες συνομιλίες (conversations). Η ύπαρξη των συνόδων είναι αναγκαία κατά τη χρήση της εφαρμογής, διότι κάθε συνομιλία καταγράφεται κατά τη διάρκεια της συνόδου, προκειμένου να αναζητηθεί και να αναγνωστεί όταν είναι αυτό επιθυμητό.

Οι σύνοδοι αποθηκεύονται στον κατάλογο του συστήματος αρχείων του λειτουργικού συστήματος του χρήστη. Όταν μια νέα σύνοδος δημιουργείται, ο χρήστης μπορεί να επιλέξει είτε ένα κατάλογο που ήδη υπάρχει, είτε να δημιουργήσει έναν νέο, δίνοντας το όνομα του καταλόγου.

Κατά την εκκίνηση της εφαρμογής, ζητείται από τον χρήστη να επιλέξει το είδος της συνόδου που επιθυμεί αυτός να χρησιμοποιήσει. Η προεπιλογή είναι η δημιουργία μιας προσωρινή συνόδου, στον κατάλογο που ορίζεται από τη ρύθμιση *${java.io.tmpdir}* και με όνομα προσωρινού αρχείου *(.tmp)* που ακολουθεί το μοτίβο *webscarabnnnnn.tmp*. Ο προαναφερθείς κατάλογος διαγράφεται αυτόματα όταν ο χρήστης εξέρχεται από την εφαρμογή.

Αν υπάρχει επιθυμία διατήρησης του αρχείου καταγραφής (audit record), για μελλοντική χρήση, πρέπει ο χρήστης να ξεκινήσει νέα σύνοδο πριν από την δημιουργία οποιαδήποτε συνομιλίας.

6.5. Πρόσθετα του Webscarab (Webscarab Plugins)

Το Webscarab χρησιμοποιεί μια σειρά από πρόσθετα (plugins) τα οποία τα χρησιμοποιεί για να δημιουργεί και να αναλύει συνομιλίες.

Το plugins, που δημιουργεί συνομιλίες, χρησιμοποιεί μια συγκεκριμένη μέθοδο προκειμένου να αποφασίσει ποιους πόρους θα ζητήσει από τον εξυπηρέτη, πώς θα παραμετροποιήσει τις επικεφαλίδες του αιτήματος αυτού κ.ά. και κατόπιν υποβάλλει το αίτημα του. Στην συνέχεια όταν λάβει την απάντηση από τον εξυπηρέτη, εκτελεί κάποιο υπολογισμό και αποφασίζει, εάν πρέπει να υποβάλει την συνομιλία στο framework.

Όλες οι συνομιλίες, που υποβάλλονται στο framework, διανέμονται σε όλα τα plugins, που είναι εγκατεστημένα. Στην συνέχεια, κάθε plugin, κάνει τις αναλύσεις που είναι προγραμματισμένο.

Τα plugins του WebScarab είναι:

- Proxy
- Manual Request
- Spider
- SessionID Analysis
- Scripted
- Fragments
- Compare
- Fuzzer
- Search
- XSS/CRLF

6.5.1. Το Πρόσθετο του εξυπηρέτη αντιπροσώπευσης (The Proxy Plugin)

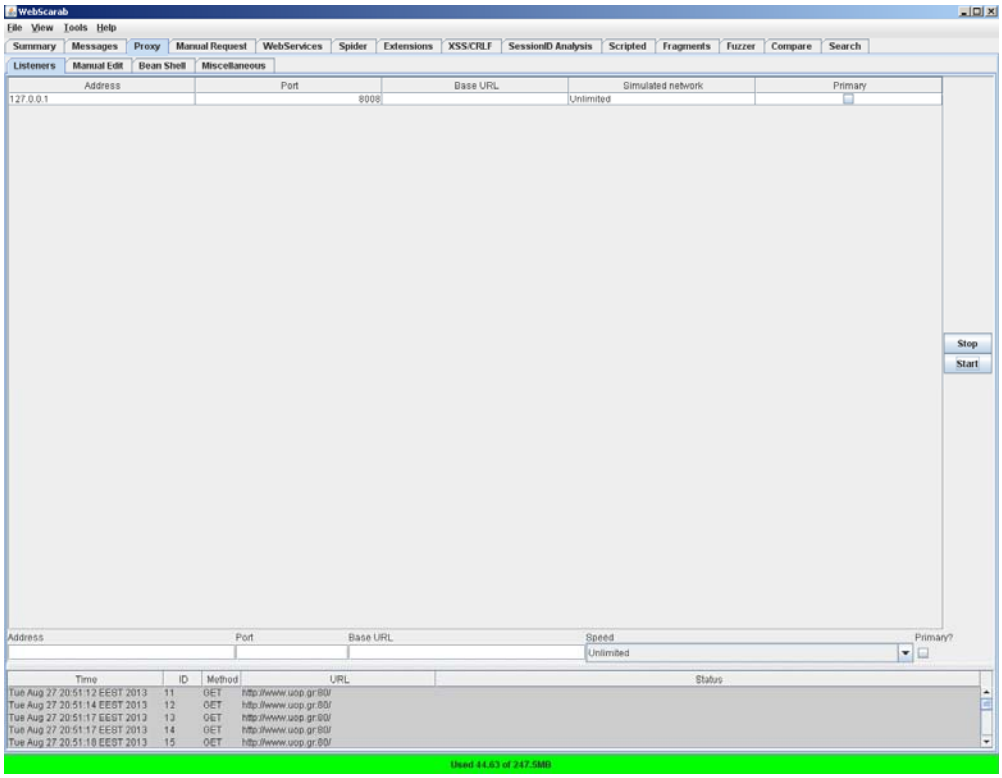
Το πρόσθετο του εξυπηρέτη αντιπροσώπευσης εγκαθιστά έναν εξυπηρέτη αντιπροσώπευσης HTTP, ο οποίος εξ ορισμού παρακολουθεί («ακούει») τη θύρα 8008 στο μηχάνημα localhost. Σε περίπτωση που χρησιμοποιηθεί αυτός ο ακροατής (listener), πρέπει να διαμορφώσουμε την εφαρμογή ιστού που χρησιμοποιούμε, έτσι ώστε να χρησιμοποιεί το WebScarab ως upstream proxy. Όταν αυτό πραγματοποιηθεί, οποιοδήποτε αίτημα στείλει η εφαρμογή ιστού, αυτό θα δρομολογηθεί μέσω του webScarab, θα καταγραφεί και θα αναλυθεί.

Επίσης, το WebScarab παρέχει υποστήριξη για παρεμπόδιση των αιτημάτων HTTPS που είναι κρυπτογραφημένα με SSL. Εάν η εφαρμογή ιστού έχει διαμορφωθεί με τρόπο τέτοιο ώστε να χρησιμοποιεί το WebScarab για αιτήματα SSL, τότε το WebScarab θα χρησιμοποιήσει το δικό του μη έμπιστο πιστοποιητικό SSL, για να διαπραγματευτεί μια κρυπτογραφημένη σύνοδο με την εφαρμογή ιστού, προκειμένου να αναγνωστεί το αίτημα που η εφαρμογή ιστού υποβάλλει, στον "ασφαλή" εξυπηρέτη. Φυσικά, από τη στιγμή που το πιστοποιητικό που

χρησιμοποιεί το Webscarab δεν το εμπιστεύεται η εφαρμογή ιστού, η εφαρμογή **πρέπει** να μας προειδοποιήσει ότι αυτό δεν είναι έμπιστο και να μας δώσει την επιλογή να συνεχίσουμε ή να σταματήσουμε.

6.5.1.1. Ακροατές εξυπηρέτη αντιπροσώπευσης (Proxy listeners)

Η εφαρμογή υποστηρίζει πολλαπλούς ακροατές (listeners) HTTP. Η λειτουργία των ακροατών είναι να μας επιτρέψουν να καθορίσουμε την διεύθυνση IP καθώς και την θύρα στην οποία «ακούμε». Οι ακροατές του Webscarab, εξ ορισμού ακούνε μόνο στο localhost, έτσι ώστε να μειωθούν οι πιθανότητες ανεπιθύμητης ακρόασης από ανώνυμους χρήστες.



Εικόνα 8. Καρτέλα επιλογών του Proxy Listener μέσα από το περιβάλλον του Webscarab

Μέσω των επιλογών, μπορούμε να καθορίσουμε μια «διεύθυνση βάσης» (“base address”) για τον ακροατή. Αυτή, καθοδηγεί τον ακροατή να λειτουργήσει ως ένας αντίστροφος εξυπηρέτης αντιπροσώπευσης (reverse proxy) με μορφή HTTP ή HTTPS διεύθυνσης (URL). Στην περίπτωση αυτή θα λειτουργήσει ως εξυπηρέτης ιστού και όχι ως εξυπηρέτης αντιπροσώπευσης και θα κατασκευάσει το URL, με τον συνδυασμό της βάσης URL και της διαδρομής που εμφανίζεται στην γραμμή αιτήματος. Στην περίπτωση που η βάση URL είναι ένα HTTPS URL τότε αμέσως θα διαπραγματευτεί μια σήραγγα SSL (SSL tunnel), πριν από την προσπάθεια ανάγνωσης του αιτήματος από την εφαρμογή ιστού.

6.5.1.2. Ενεργές συνομιλίες (Active Conversations)

Ο πίνακας των ενεργών συνομιλιών βρίσκεται στην καρτέλα Proxy Listeners και μας εμφανίζει τα πιο πρόσφατα καθώς και τα τρέχοντα αιτήματα που έχουν χειριστεί οι ακροατές. Ο πίνακας αυτός, είναι χρήσιμος στην περίπτωση που επιχειρούμε να αντιμετωπίσουμε προβλήματα (troubleshooting) που αφορούν π.χ. τη συνδεσιμότητα.

6.5.1.3. Πρόσθετα εξυπηρέτη αντιπροσώπευσης(Proxy Plugins)

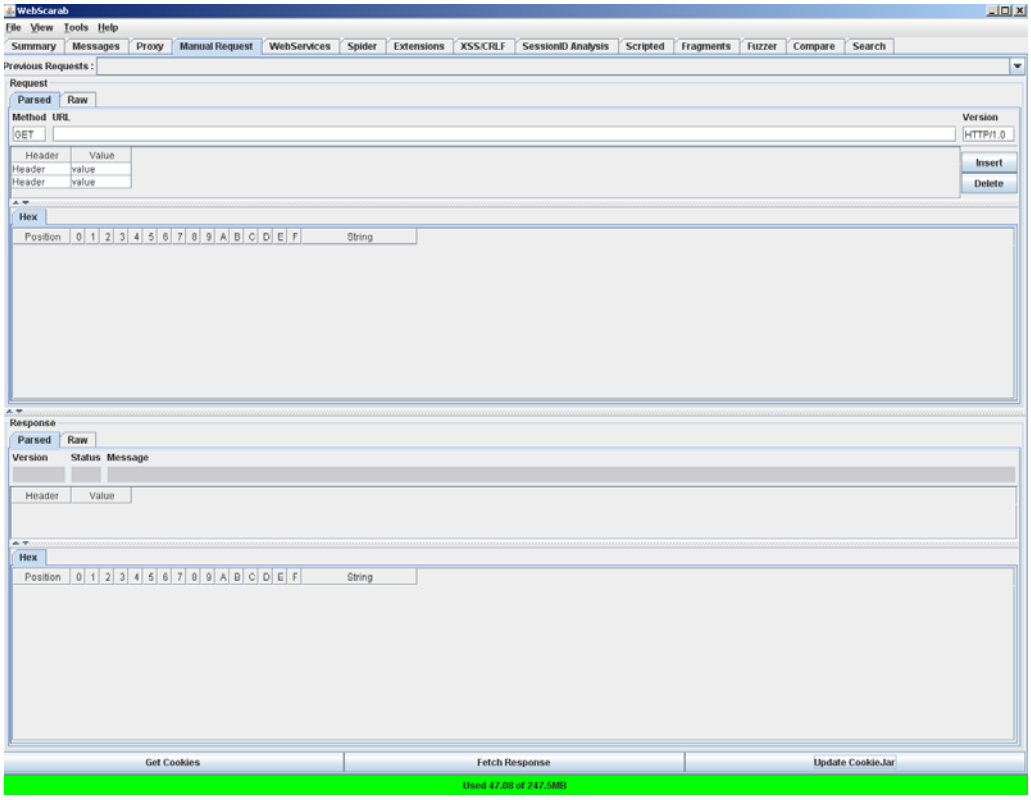
Ο χρήστης, εφόσον διαθέτει τις κατάλληλες γνώσεις προγραμματισμού, μπορεί να κατασκευάσει τα δικά του πρόσθετα εξυπηρέτη αντιπροσώπευσης, τα οποία θα πραγματοποιούν τις δικές του τροποποιήσεις στα αιτήματα και τις απαντήσεις κατά τη μεταφορά των αιτημάτων μέσω του εξυπηρέτη αντιπροσώπευσης.

Μερικά υπάρχοντα διαθέσιμα πρόσθετα είναι:

- **Manual Intercept:** Επιτρέπει στον χρήστη να υποκλέψει αιτήματα από την εφαρμογή ιστού καθώς και τις απαντήσεις προς τον εξυπηρέτη ιστού, να τις επιθεωρήσει και να τις τροποποιήσει πριν τις μεταδώσει. Είναι ένα χρήσιμο πρόσθετο ιδιαίτερα στην περίπτωση που χρειάζεται να υποβάλλουμε μια φόρμα σε ένα εξυπηρέτη, αλλά η επικύρωση της Javascript απορρίπτει τις τιμές που στέλνουμε. Υπάρχει δυνατότητα αλλαγής του αιτήματος μετά την εκτέλεση της επικύρωσης.
- **Bean Shell:** Επιτρέπει προκαθορισμένες τροποποιήσεις ενός αιτήματος και της απόκρισης. Είναι ένα χρήσιμο πρόσθετο στη περίπτωση που χρειάζεται να εκτελέσουμε την ίδια τροποποίηση σε πολλαπλές συνομιλίες ή στην περίπτωση που η τροποποίηση είναι σύνθετη. Κατά τη χρήση του πρόσθετου, έχουμε πλήρη πρόσβαση στο αίτημα πριν αυτό σταλεί στον εξυπηρέτη, καθώς και πλήρη πρόσβαση στην απάντηση πριν αυτή σταλεί στην εφαρμογή ιστού.
- **Miscellaneous plugins:** Ένας αριθμός από πρόσθετα που λειτουργούν στις συνδιαλέξεις που πραγματοποιούνται μέσω αντιπροσώπευσης.
    - **Reveal hidden fields**: Αλλάζει οποιαδήποτε πεδία εμπεριέχονται σε φόρμες χαρακτηρισμένες ως “hidden”, σε πεδία τύπου “text”. Με τον τρόπο αυτό, αναγκάζει την εφαρμογή ιστού να εμφανίσει αυτά τα πεδία, διευκολύνοντας το χρήστη να τα παρατηρήσει και να τα τροποποιήσει.
    - **Prevent caching:** Αφαιρεί κάθε επικεφαλίδα (header) “if-modified-since” ή παρόμοιες επικεφαλίδες από τα αιτήματα, για να εξασφαλίσει ότι το Webscarab έχει πάντα ένα αντίγραφο του σώματος (body) της απάντησης, αντί να επιτρέπει στην εφαρμογή ιστού να χρησιμοποιεί ένα τοπικά αποθηκευμένο αντίγραφο.
    - **Inject cookies:** Επιτρέπει στο Webscarab να παρακάμψει τυχόν cookies που είναι αποθηκευμένα στην εφαρμογή ιστού ή να της παρέχει cookies που δεν έχει. Τα cookies ανακτώνται από την λίστα “Shared cookies” και φιλτράρονται για την χρησιμοποίηση τους από την εφαρμογή, σύμφωνα με τις ιδιότητες του domain και του path.
    - **Collect cookies:** Εξάγει κάθε επικεφαλίδα (header) “set-cookie” από τις απαντήσεις που περνούν μέσω του εξυπηρέτη αντιπροσώπευσης και τις προσθέτει στην λίστα “shared-cookies”. Οι απαντήσεις αυτές, μπορούν να χρησιμοποιηθούν αργότερα από άλλα πρόσθετα, αν αυτό είναι επιθυμητό.

6.5.2. Το πρόσθετο χειροκίνητου αιτήματος (The Manual Request Plugin)

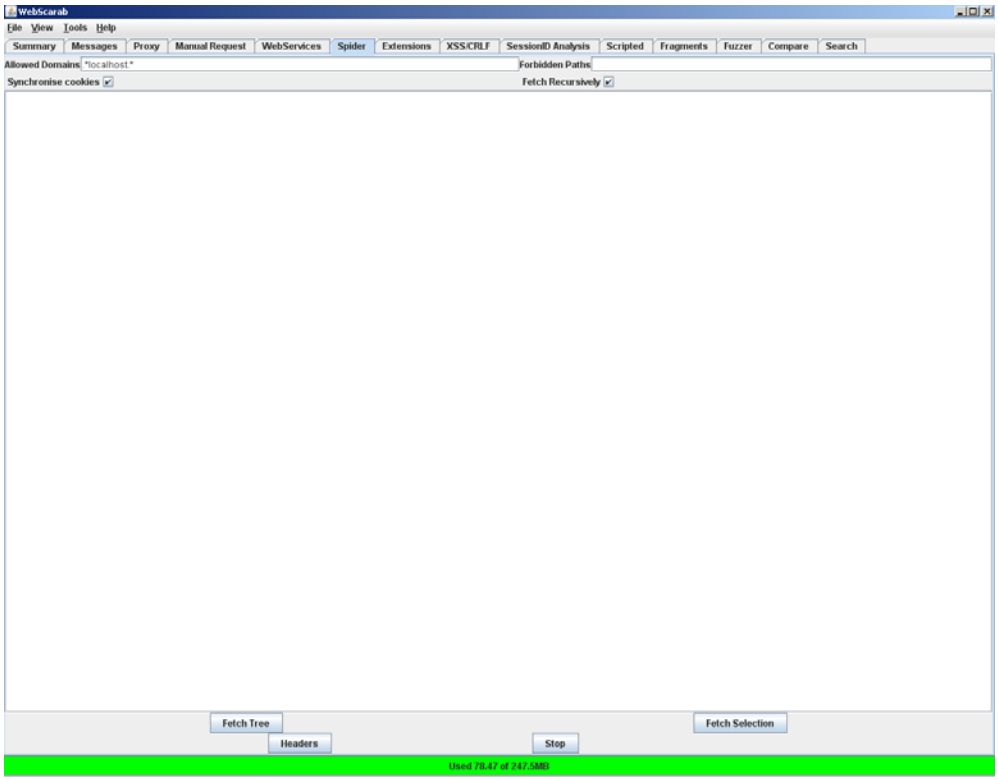
Επιτρέπει στο χρήστη να στείλει χειροκίνητα ένα αίτημα στον εξυπηρέτη. Είναι επίσης δυνατόν, να σταλεί κατ' επανάληψη ένα προηγούμενο αίτημα επιλέγοντάς το από κατάλληλη επιλογή του μενού. Προηγούμενα αιτήματα που έχουν φορτωθεί στον editor, μπορούν να υποστούν επεξεργασία πριν σταλούν στον εξυπηρέτη. Με την επιλογή "Fetch Response", το Webscarab στέλνει το αίτημα στον κατάλληλο εξυπηρέτη και αποθηκεύει την συζήτηση για ανάλυση από άλλα πρόσθετα του προγράμματος. Με την επιλογή "Get cookies" ανακτά τα cookies που σχετίζονται με τη διεύθυνση URL από την λίστα "shared cookies" και τα προσθέτει στο αίτημα. Τέλος με την επιλογή "Update cookie Jar" αναζητά επικεφαλίδες "set-cookie" από την απάντηση που ανακτήθηκε και τις προσθέτει στη λίστα "shared cookies".



Εικόνα 9. Καρτέλα επιλογών του Manual Request μέσα από το περιβάλλον του Webscarab

6.5.3. Το πρόσθετο Spider (The Spider Plugin)

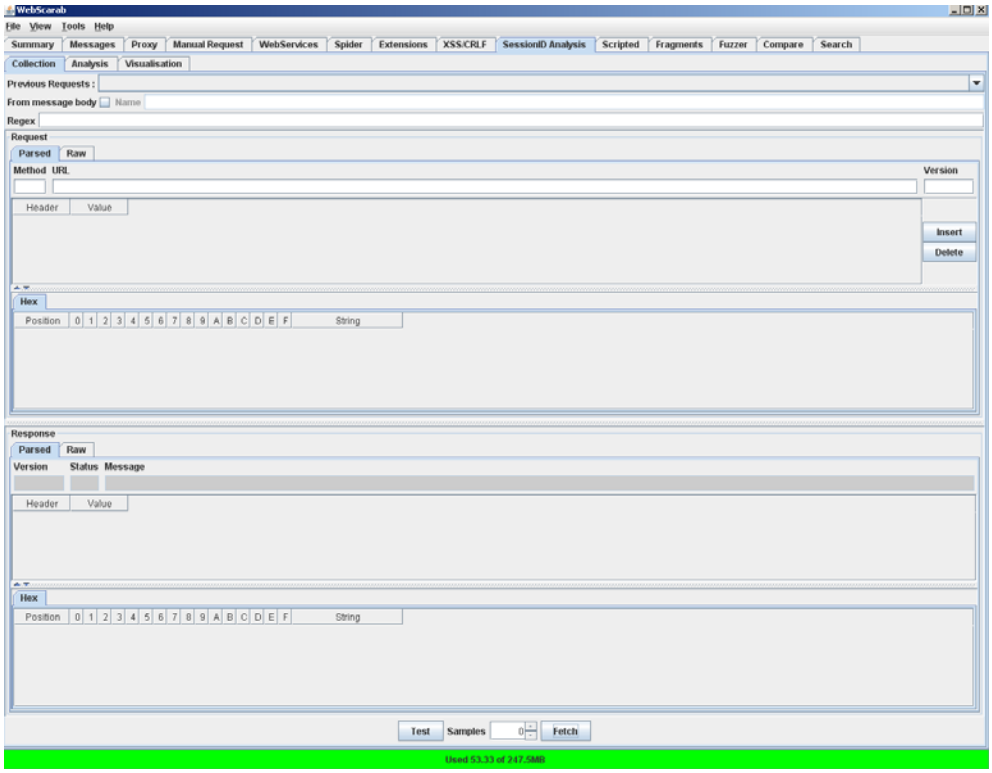
Αναλύει τις απαντήσεις, ώστε να αναγνωρίσει τυχόν συνδέσμους (links) στο κυρίως σώμα της απάντησης, ή στην επικεφαλίδα "Location". Εάν το URL δεν εμφανίζεται, τότε προστίθεται σε ένα δέντρο και αυτόματα μπορεί να μεταφορτωθεί όταν ζητηθεί. Το Webscarab έχει δυο τρόπους να «συλλαμβάνει» (fetching) τους συνδέσμους οι οποίοι δεν είναι ορατοί. Τον "Fetch Tree" με τον οποίο απαριθμεί όλους τους άγνωστους συνδέσμους στον επιλεγμένο κόμβο και τους τοποθετεί σε μια ουρά για ανάκτηση κατ' απαίτηση, και τον "Fetch Selection"με τον οποίο τοποθετεί σε ουρά μόνο τους επιλεγμένους κόμβους για ανάκτηση κατ' απαίτηση.



Εικόνα 10. Καρτέλα επιλογών του πρόσθετου Spider μέσα από το περιβάλλον του Webscarab

6.5.4. Session ID Analysis

Είναι ένα χρήσιμο πρόσθετο, το οποίο μας δείχνει την ευκολία με την οποία ένας επιτιθέμενος μπορεί να πραγματοποιήσει μια επιτυχή επίθεση τύπου ωμής βίας (Brute Force) στο sessionid του θύματος.



Εικόνα 11. Καρτέλα επιλογών του πρόσθετου session ID Analysis μέσα από το περιβάλλον του Webscarab

Συλλέγει ένα δείγμα με τα αναγνωριστικά συνόδων (session identifiers), πιθανά από ένα cookie της εφαρμογής ιστού, το οποίο συνήθως είναι ορισμένο σε ένα κρυφό πεδίο μιας φόρμας, μιας HTML σελίδας. Το Webscarab μαρκάρει όλα τα session id που συλλέγει με την ημερομηνία και την ώρα και στη συνέχεια, εκτελώντας κατάλληλους υπολογισμούς στη τιμή του string, το μετατρέπει σε ένα αριθμό, σχεδιάζοντας την τιμή αυτή συναρτήσει του χρόνου σε γραφική παράσταση. Η απεικόνιση των τιμών σε γραφική παράσταση, κάνει πιο εύκολη την διαδικασία για τον χρήστη.

- **Collecting sessionids:** Υπάρχουν δύο τρόποι για τη συλλογή των sesionids:
  - Συλλογή από ένα cookie στις επικεφαλίδες της απάντησης
  - Συλλογή από το ταίριασμα μιας κανονικής έκφρασης, συγκρινόμενο με το σώμα της απάντησης

  Όσον αφορά την πρώτη περίπτωση, πρέπει να μην έχουμε επιλεγμένη την επιλογή “from message body”, ενώ για τον δεύτερο τρόπο, πρέπει να την έχουμε επιλεγμένη.
- **Session ID Analysis:** Ενώ η διαδικασία συλλογής των sessionid’s συνεχίζεται μπορούμε να αλλάξουμε καρτέλα, να μεταβούμε στην καρτέλα Analysis και να δούμε τις τιμές που συλλέγονται. Επιλέγοντας το sessionid που μας ενδιαφέρει, μπορούμε να δούμε σε στήλες το χρόνο που αυτό συλλέχθηκε, την τιμή του string, την υπολογισμένη τιμή η οποία μπορεί να αλλάξει με την πάροδο του χρόνου και τη διαφορά μεταξύ των επομένων υπολογισμένων τιμών.
- **Session ID Visualization:** Στην συγκεκριμένη καρτέλα μπορούμε να δούμε τη γραφική παράσταση του επιλεγμένου sessionid. Κάνοντας χρήση της γραφικής παράστασης, είναι πιο εύκολο για τον χρήστη να προσδιορίσει τα γραμμικά ή τα επαναλαμβανόμενα μοτίβα στα συλλεχθέντα sessionid’s.

6.5.5. Πρόσθετο εκτέλεσης script (The scripted plugin)

Έχει ως στόχο να δώσει στους χρήστες τη δυνατότητα να δημιουργούν σενάρια δοκιμών, χρησιμοποιώντας τις λειτουργίες του Webscarab.

Η υποστήριξη των scripts παρέχεται από το Bean Scripting Framework (BSF)[27] το οποίο πρακτικά σημαίνει πως τα scripts μπορούν να γραφούν σε οποιαδήποτε γλώσσα που υποστηρίζεται από το BSF, ενώ η γλώσσα προεπιλογής είναι η Beanshell[28].

[27] Μέθοδος που επιτρέπει την εισαγωγή scripting μέσα σε κώδικα της Java.Παρέχει ένα σύνολο από κλάσεις της Java οι οποίες παρέχουν υποστήριξη σε εφαρμογές που δέχονται scripting. Μερικά παραδείγματα γλωσσών που μπορούν να χρησιμοποιηθούν σε συνδυασμό με την FSB είναι οι Jython και Tcl, JRuby και Groovy.

Είναι δημιούργημα της IBM και η ανάπτυξη του συνεχίζεται από τη Apache Software Foundation.

[28] HTTP://WWW.beanshell.org

Μετά την εκτέλεση του script, το πρόσθετο εξάγει ένα αντικείμενο Java και μας επιτρέπει να αλληλεπιδράσουμε με άλλες λειτουργίες ή άλλα πρόσθετα του Webscarab για άλλου είδους αναλύσεις.

6.5.6. The fragments plugin

Η λειτουργία του πρόσθετου είναι να αναλύει τις απαντήσεις HTML και να αναζητά scripts και σχόλια. Μπορούμε να δούμε εάν υπάρχουν κρυμμένοι σύνδεσμοι ή κώδικας αποσφαλμάτωσης, καθώς και να κατανοήσουμε τον τρόπο με τον οποίο οι σελίδες πρέπει να λειτουργούν.

6.5.7. Το πρόσθετο σύγκρισης (The Compare plugin)

Επιτρέπει στο χρήστη να κρίνει το βαθμό της διαφοράς μεταξύ διαφόρων απαντήσεων. Είναι μια χρήσιμη λειτουργία, ειδικά στην περίπτωση που έχουμε κάνει ένα αριθμό αιτημάτων για μια συγκεκριμένη διεύθυνση URL και θα θέλαμε να αξιολογήσουμε τα αποτελέσματα. Στον πίνακα των αποτελεσμάτων εμφανίζονται δυο στήλες. Η πρώτη στήλη, αυτή της συνομιλίας, μας δείχνει τη συνομιλία και η δεύτερη στήλη, αυτή της διαφοράς, μας δείχνει τον αριθμό των λέξεων που διαφέρουν στις απαντήσεις.

6.5.8. Το πρόσθετο Fuzzer (The Fuzzer[29] plugin)

Μας επιτρέπει να δίνουμε ένα συνδυασμό τιμών σε ένα εξυπηρέτη. Η ιδέα είναι ότι μπορούμε να ρυθμίσουμε τη μέθοδο του αιτήματος, το βασικό URL χωρίς οποιαδήποτε παράμετρο, την έκδοση του αιτήματος, τις επικεφαλίδες και μια λίστα παραμέτρων. Η **παράμετρος** ορίζεται, από τη **θέση** της (Path, Fragment, Query, Cookie, Body), το **όνομα** της, τον **τύπο** της, την **προκαθορισμένη τιμή** που έχει, την **Fuzz προτεραιότητα** και της μια **Fuzz πηγή**.

- Η **προκαθορισμένη τιμή** είναι η τιμή που θα υποβληθεί, εφόσον δεν έχει οριστεί καμία πηγή Fuzz,
- Η **Fuzz προτεραιότητα** καθορίζει το πώς συνδυάζονται οι διάφορες Fuzz πηγές: αν όλες οι προτεραιότητες έχουν την ίδια τιμή, τότε ο αριθμός των ερωτημάτων που θα υποβληθούν θα είναι ο αριθμός των στοιχείων από την πιο σύντομη Fuzz πηγή. Αν έχουν διαφορετικές τιμές, ο αριθμός των ερωτημάτων που δημιουργούνται θα είναι το εξαγόμενο του αριθμού των στοιχείων σε κάθε επίπεδο.

Το πρόσθετο, επεξεργάζεται τις παραμέτρους με συγκεκριμένη σειρά. Path, Fragment, Query, Cookie, Body.

6.5.9. Το πρόσθετο της αναζήτησης (The search plugin)

Μας δίνει την δυνατότητα να εκτελέσουμε αυθαίρετα script beanshell, για να εντοπίσουμε «ενδιαφέρουσες» συνομιλίες. Μας παρέχεται το αίτημα, η απάντηση καθώς και η προέλευση της συνομιλίας και χρησιμοποιώντας τις μεθόδους των

[29] Fuzzing: Είναι μια τεχνική ελέγχου λογισμικού που παρέχει τυχαία δεδομένα (“fuzz”) ως είσοδο σε κάποιο πρόγραμμα.

κλάσεων μπορούμε να επιστρέψουμε μια σωστή (true) ή λάθος (false) τιμή. Η σωστή τιμή υποδηλώνει ότι μια ενδιαφέρουσα συνομιλία πρέπει να εμφανιστεί και η λανθασμένη τιμή υποδηλώνει πως δεν θα εμφανιστεί.

6.5.10. Το πρόσθετο XSS/CRL (XSS/CRLF plugin)

Εξετάζει αιτήματα HTTP τα οποία είναι ύποπτα για τις ευπάθειες cross-site scripting και έκχυσης CRLF (HTTP response splitting). Το πρόσθετο αυτό, αναλύει όλες τις συνομιλίες HTTP που περνούν από την εφαρμογή παθητικά και επιθεωρεί κάθε αίτημα και απόκριση προκειμένου να ελέγξει:

- Κάθε τιμή των παραμέτρων GET/POST που απεικονίζεται στο σώμα της απόκρισης HTTP, κάτι που πιθανότατα μας υποδηλώνει ευπάθεια XSS,
- Κάθε τιμή των παραμέτρων GET/POST που απεικονίζεται στις επικεφαλίδες HTTP των αποκρίσεων, κάτι που πιθανότατα υποδηλώνει ευπάθεια CRLF.

Ο πίνακας που εμφανίζεται στο άνω μισό μέρος του παραθύρου του πρόσθετου, δείχνει όλα τα αιτήματα HTTP τα οποία είναι ύποπτα για τις ευπάθειες που μελετάμε. Επιλέγοντας το πλήκτρο “Check”, θα επιχειρήσει να πραγματοποιήσει έναν έλεγχο για να διαπιστώσει εάν οι κατάλληλες συμβολοσειρές για ευπάθειες XSS/CRLF περνάνε επιτυχώς. Αν ο έλεγχος σε συγκεκριμένη συνομιλία είναι επιτυχής (π.χ. αποδεκτό string, δημιουργία νέας HTTP επικεφαλίδας), η συνομιλία θα εμφανιστεί στο κάτω μέρος του πίνακα. Αν οι έλεγχοι πραγματοποιηθούν χωρίς επιτυχία, ο πίνακας θα παραμείνει κενός.

Πιο συγκεκριμένα, για τη διαπίστωση ύπαρξης της ευπάθειας XSS, το πρόσθετο στέλνει ένα ελεγχόμενο δοκιμαστικό XSS string και ελέγχει το κυρίως σώμα της απάντησης για την ύπαρξη του δοκιμαστικού string. Αν υπάρξει, τότε η υπό έλεγχο εφαρμογή ίσως είναι ευάλωτη στην ευπάθεια.

Για τη διαπίστωση ευπάθειας HTTP response splitting, το πρόσθετο στέλνει ξανά ένα ελεγχόμενο δοκιμαστικό CRLF string, το οποίο υποτίθεται πως θα δημιουργήσει μια νέα HTTP επικεφαλίδα, ως τιμή μιας δυνητικά ευάλωτης παραμέτρου που ελέγχει τις επικεφαλίδες ων HTTP αποκρίσεων. Αν η επικεφαλίδα υπάρξει, τότε η υπό έλεγχο εφαρμογή ίσως είναι ευάλωτη στην ευπάθεια.

6.6. Μετρήσεις ασφαλείας και ανάλυση στοιχείων του δικτύου του Τμήματος Κοινωνικής & Εκπαιδευτικής Πολιτικής του Πανεπιστημίου Πελοποννήσου

Παρακάτω, στον Πίνακας 3, παρουσιάζεται η κατηγοριοποίηση των ελέγχων που πρέπει να γίνουν από τους ελεγκτές εφαρμογών ιστού, σύμφωνα με τη μέθοδο δοκιμής διείσδυσης του οργανισμού OWASP. Η συγκεκριμένη κατηγοριοποίηση του οδηγού ελέγχου του OWASP (OWASP Testing Guide) περιλαβαίνει ένα πλαίσιο ελέγχου διείσδυσης (penetration testing) βασισμένο σε βέλτιστες πρακτικές, το οποίο μπορούν να χρησιμοποιούν οι χρήστες στους οργανισμούς τους, και έναν οδηγό ελέγχου διείσδυσης που περιγράφει τεχνικές για τον έλεγχο των πιο συνηθισμένων ζητημάτων ασφάλειας σε διαδικτυακές εφαρμογές και υπηρεσίες διαδικτύου.

Ως οδηγός χρησιμοποιήθηκε η έκδοση 3 του οδηγού ελέγχου (OWASP Testing Guide Version 3), διότι η έκδοση 4 βρισκόταν έκδοση σε μορφή σχεδίου (draft). Ο Οδηγός Ελέγχου περιγράφει λεπτομερώς και το γενικό Πλαίσιο Ελέγχου (Testing Framework) αλλά και τις τεχνικές που απαιτούνται για να εφαρμοστεί στην πράξη αυτό το πλαίσιο.

Οι έλεγχοι πραγματοποιήθηκαν σε πραγματικές συνθήκες στο δίκτυο του Τμήματος Κοινωνικής & Εκπαιδευτικής Πολιτικής του Πανεπιστημίου Πελοποννήσου, ακολουθώντας την τεχνική “black box”. Απαραίτητες προϋποθέσεις ήταν ότι σε καμία περίπτωση δεν έπρεπε να παρεμποδιστεί η λειτουργία του δικτύου και των υπηρεσιών που παρέχονται πάνω από αυτό, καθώς και η εύρυθμη λειτουργία του Τμήματος.

Τα αποτελέσματα, έδειξαν την πραγματική κατάσταση ασφαλείας του ιστοχώρου και αξιολογήθηκε η αποτελεσματικότητα της άμυνας σε μια ενδεχόμενη επίθεση. Ωστόσο, δεν πραγματοποιήθηκε το σύνολο των ελέγχων, διότι κάποιοι έλεγχοι απαιτούν άμεση παρέμβαση στον εξυπηρέτη της εφαρμογής ενώ κάποιοι άλλοι, απαιτούν πρόσβαση τοπικά στην εφαρμογή. Γενικά, λάβαμε υπόψη μας την κρισιμότητα της εφαρμογής, η οποία αποτελεί φορέα γνώσης και εκπαίδευσης και σε καμία περίπτωση δεν θελήσαμε να δημιουργήσουμε πρόβλημα στην ασφάλεια και στην απρόσκοπτη λειτουργία της.

Κατηγορία (Category)	Αρ. Αναφοράς (Ref.Number)	Όνομα Ελέγχου (Test name)	Ευπάθεια που ελέγχεται (Vulnerability)
Συλλογή Πληροφοριών (Information Gatherning)	OWASP-IG-001	Spiders, Robots and Crawlers	-
	OWASP-IG-002	Search Engine Discovery/Reconnaissance	-
	OWASP-IG-003	Identify application entry points	-
	OWASP-IG-004	Testing for Web Application Fingerprint	-
	OWASP-IG-005	Application Discovery	-
	OWASP-IG-006	Analysis of Error Codes	Information Disclosure
Έλεγχος Διαχείρισης Διαμόρφωσης (Configuration Management Testing)	OWASP-CM-001	SSL/TLS Testing (SSL Version, Algorithms, Key length, Digital Cert. Validity)	SSL Weakness
	OWASP-CM-002	DB Listener Testing	DB Listener weak
	OWASP-CM-003	Infrastructure Configuration Management Testing	Infrastructure Configuration management weakness
	OWASP-CM-004	Application Configuration Management Testing	Application Configuration management weakness
	OWASP-CM-005	Testing for File Extensions Handling	File extensions handling
	OWASP-CM-006	Old, backup and unreferenced files	Old, backup and unreferenced files
	OWASP-CM-007	Infrastructure and Application Admin Interfaces	Access to Admin interfaces
	OWASP-CM-008	Testing for HTTP Methods and XST	HTTP Methods enabled, XST permitted, HTTPVerb
Έλεγχος Αυθεντικοποίησης (Authentication	OWASP-AT-001	Credentials transport over an encrypted channel	Credentials transport over an encrypted channel
	OWASP-AT-002	Testing for user enumeration	User enumeration

Κατηγορία (Category)	Αρ. Αναφοράς (Ref.Number)	Όνομα Ελέγχου (Test name)	Ευπάθεια που ελέγχεται (Vulnerability)
Testing)	OWASP-AT-003	Testing for Guessable (Dictionary) User Account	Guessable user account
	OWASP-AT-004	Brute Force Testing	Credentials Brute forcing
	OWASP-AT-005	Testing for bypassing authentication schema	Bypassing authentication schema
	OWASP-AT-006	Testing for vulnerable remember password and pwd reset	Vulnerable remember password, weak pwd reset
	OWASP-AT-007	Testing for Logout and Browser Cache Management	Logout function not properly implemented, browser cache weakness
	OWASP-AT-008	Testing for CAPTCHA	Weak Captcha implementation
	OWASP-AT-009	Testing Multiple Factors Authentication	Weak Multiple Factors Authentication
	OWASP-AT-010	Testing for Race Conditions	Race Conditions vulnerability
Έλεγχος Διαχείρισης Συνόδου (Session Management)	OWASP-SM-001	Testing for Session Management Schema	Bypassing Session Management Schema, Weak Session Token
	OWASP-SM-002	Testing for Cookies attributes	Cookies are set not ‘HTTP Only’, ‘Secure’, and no time validity
	OWASP-SM-003	Testing for Session Fixation	Session Fixation
	OWASP-SM-004	Testing for Exposed Session Variables	Exposed sensitive session variables
	OWASP-SM-005	Testing for CSRF	CSRF
Έλεγχος Εξουσιοδότησης (Authorization Testing)	OWASP-AZ-001	Testing for Path Traversal	Path Traversal
	OWASP-AZ-002	Testing for bypassing authorization schema	Bypassing authorization schema
	OWASP-AZ-003	Testing for Privilege Escalation	Privilege Escalation

Κατηγορία (Category)	Αρ. Αναφοράς (Ref.Number)	Όνομα Ελέγχου (Test name)	Ευπάθεια που ελέγχεται (Vulnerability)
Έλεγχος Επιχειρησιακής Λογικής (Business Logic Testing)	OWASP-BL-001	Testing for business logic	Bypassable business logic
Έλεγχος Επικύρωσης Δεδομένων (Data Validation Testing)	OWASP-DV-001	Testing for Reflected Cross Site Scripting	Reflected XSS
	OWASP-DV-002	Testing for Stored Cross Site Scripting	Stored XSS
	OWASP-DV-003	Testing for DOM based Cross Site Scripting	DOM XSS
	OWASP-DV-004	Testing for Cross Site Flashing	Cross Site Flashing
	OWASP-DV-005	SQL Injection	SQL Injection
	OWASP-DV-006	LDAP Injection	LDAP Injection
	OWASP-DV-007	ORM Injection	ORM Injection
	OWASP-DV-008	XML Injection	XML Injection
	OWASP-DV-009	SSI Injection	SSI Injection
	OWASP-DV-010	XPath Injection	XPath Injection
	OWASP-DV-011	IMAP/SMTP Injection	IMAP/SMTP Injection
	OWASP-DV-012	Code Injection	Code Injection
	OWASP-DV-013	OS Commanding	OS Commanding
	OWASP-DV-014	Buffer overflow	Buffer overflow
	OWASP-DV-015	Incubated vulnerability Testing	Incubated vulnerability
	OWASP-DV-016	Testing for HTTP Splitting/Smuggling	HTTP Splitting, Smuggling
Έλεγχος Άρνησης	OWASP-DS-001	Testing for SQL Wildcard Attacks	SQL Wildcard vulnerability

Κατηγορία (Category)	Αρ. Αναφοράς (Ref.Number)	Όνομα Ελέγχου (Test name)	Ευπάθεια που ελέγχεται (Vulnerability)
Παροχής Υπηρεσιών (Denial of Service Testing)	OWASP-DS-002	Locking Customer Accounts	Locking Customer Accounts
	OWASP-DS-003	Testing for DoS Buffer Overflows	Buffer Overflows
	OWASP-DS-004	User Specified Object Allocation	User Specified Object Allocation
	OWASP-DS-005	User Input as a Loop Counter	User Input as a Loop Counter
	OWASP-DS-006	Writing User Provided Data to Disk	Writing User Provided Data to Disk
	OWASP-DS-007	Failure to Release Resources	Failure to Release Resources
	OWASP-DS-008	Storing too Much Data in Session	Storing too Much Data in Session
Έλεγχος Υπηρειών Ιστού (Web Services Testing)	OWASP-WS-001	WS Information Gathering	-
	OWASP-WS-002	Testing WSDL	WSDL Weakness
	OWASP-WS-003	XML Structural Testing	Weak XML Structure
	OWASP-WS-004	XML content-level Testing	XML content-level
	OWASP-WS-005	HTTP GET parameters/REST Testing	WS HTTP GET parameters /REST
	OWASP-WS-006	Naughty SOAP attachments	WS Naughty SOAP attachments
	OWASP-WS-007	Replay Testing	WS Replay Testing
Έλεγχος AJAX (AJAX Testing)	OWASP-AJ-001	AJAX Vulnerabilities	-
	OWASP-AJ-002	AJAX Testing	AJAX weakness

Πίνακας 3. Λίστα κατηγοριοποίησης ελέγχων του OWASP

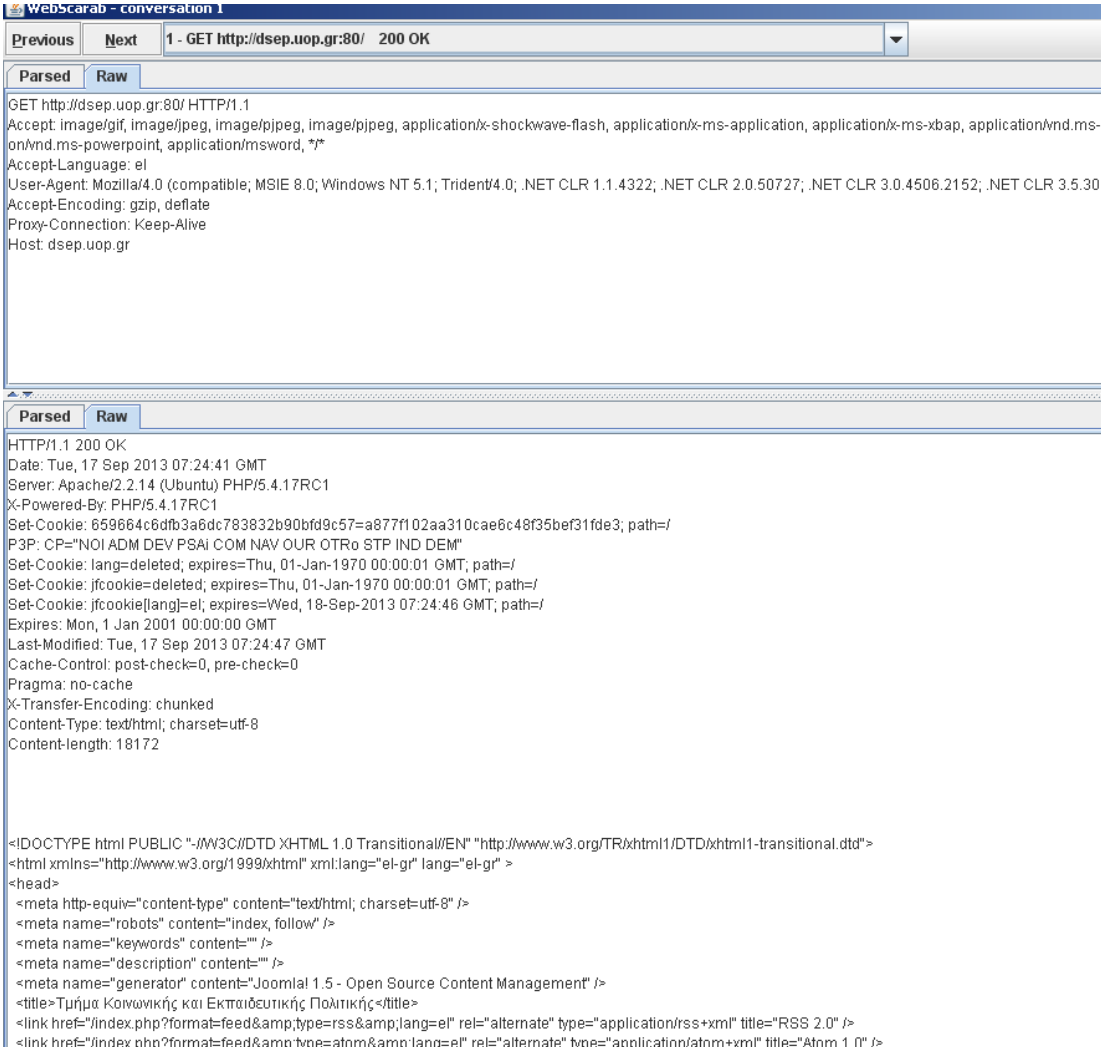
6.7. Συλλογή πληροφοριών (Information Gathering)

Η συλλογή όσο το δυνατόν περισσοτέρων πληροφοριών (Information Gathering) σχετικά με την εφαρμογή-στόχο και η ανάλυση αυτών, είναι το πρώτο και αναγκαίο βήμα για μια δοκιμή διείσδυσης.

6.7.1. Λήψη αποτυπωμάτων, Αρ. Αναφοράς OWASP-IG-004

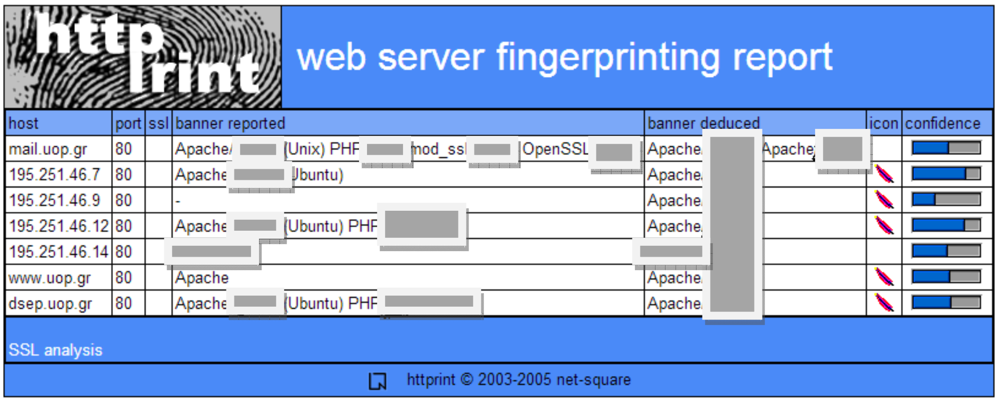
Στην διαδικασία της συλλογής των πληροφοριών “fingerprint” έγινε έλεγχος χειροκίνητα με το Webscarab, στη συνέχεια έγινε διερεύνηση και διασταύρωση των αποτελεσμάτων με μια αυτοματοποιημένη μέθοδο και τέλος χρησιμοποιήθηκε και μια on-line μέθοδος όπως ακριβώς αναφέρεται στο OWASP Testing Guide V3. Με όλες τις μεθόδους, λήφθηκαν τα ίδια αποτελέσματα.

Στην (Εικόνα 12) φαίνεται ένα στιγμιότυπο με τις πληροφορίες που έχει συλλέξει η εφαρμογή Webscarab κατά την πλοήγηση μας στον ιστοχώρο του Τμήματος Κοινωνικής & Εκπαιδευτικής Πολιτικής του Πανεπιστημίου Πελοποννήσου. Μπορούμε να δούμε αρκετές πληροφορίες όπως για το λειτουργικό σύστημα του εξυπηρέτη, τον εξυπηρέτη ιστού, την έκδοση PHP κ.ά.



Εικόνα 12. Πληροφορίες του ιστοχώρου HTTP://dsep.uop.gr όπως φαίνονται από το Webscarab

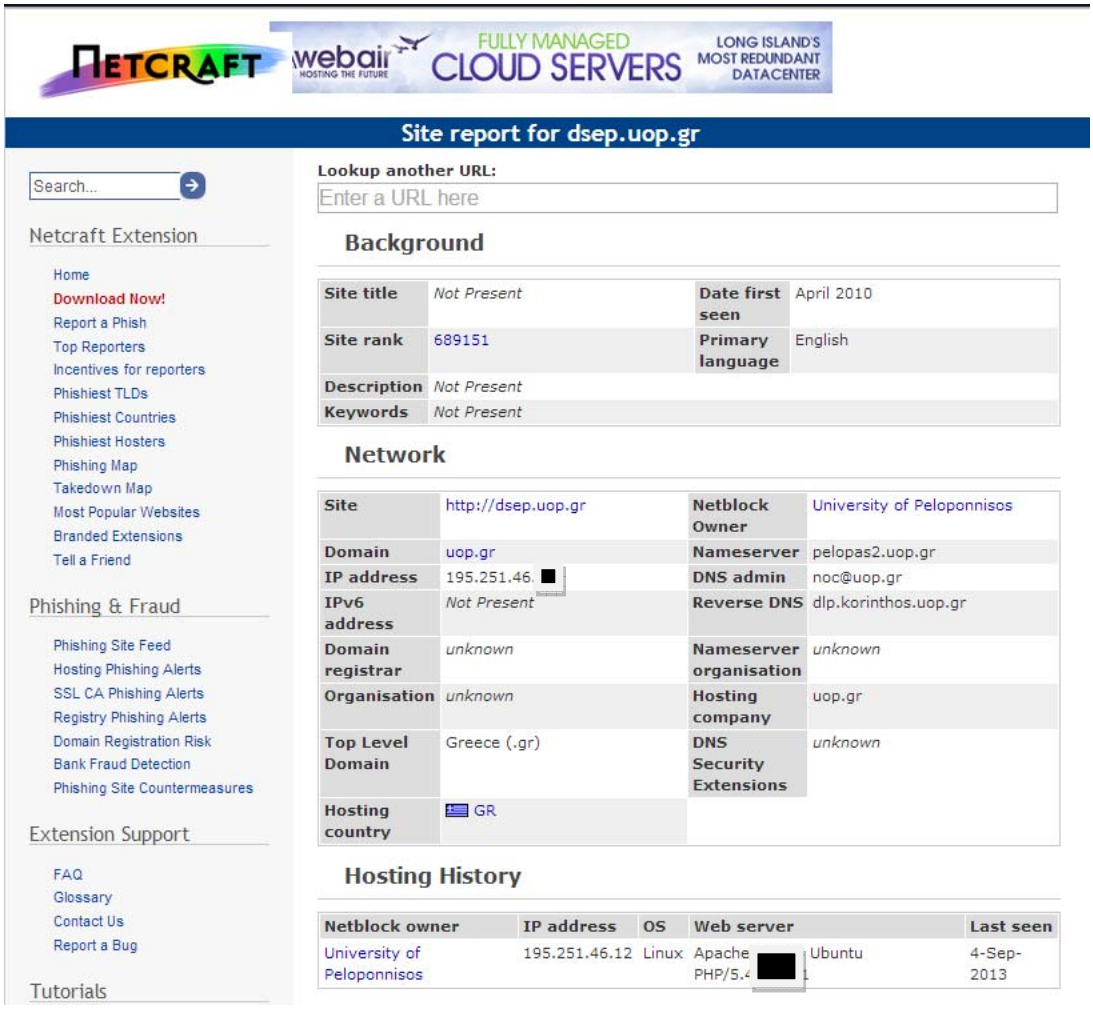
Στον έλεγχο με την αυτόματη μέθοδο, έγινε χρήση του εργαλείου “httpprint” και τα αποτελέσματα που λάβαμε παρατίθενται στην (Εικόνα 13):



Εικόνα 13. Αποτελέσματα αυτοματοποιημένου ελέγχου με το “httpprint” στο υποδίκτυο 195.251.46.0 /28

Στον έλεγχο με την on-line μέθοδο χρησιμοποιήσαμε το εργαλείο “Netcraft”. Το εργαλείο αυτό μας επιτρέπει να ανακτήσουμε πληροφορίες σχετικά με το

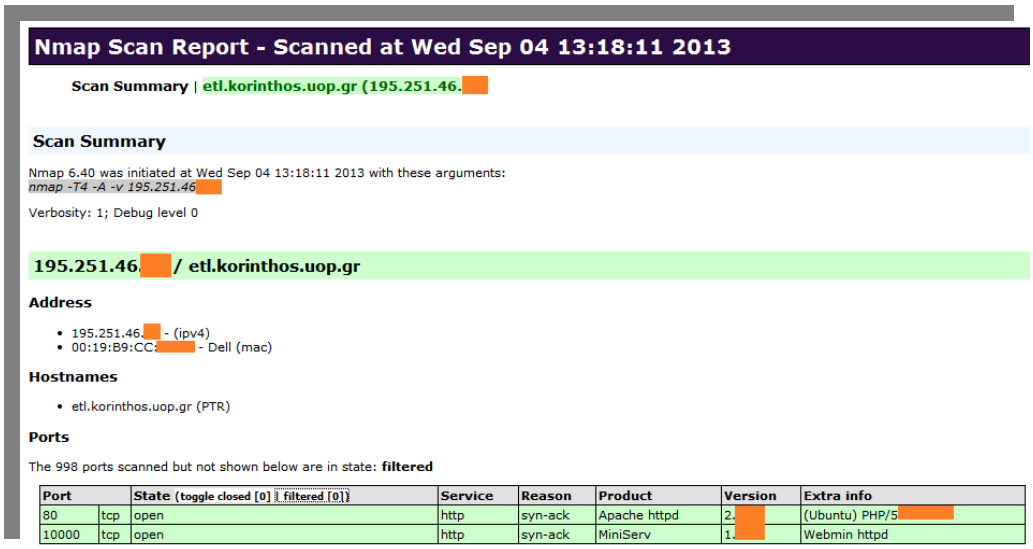
«ιστοσελίδα - στόχο» που θέλουμε να ελέγξουμε καθώς και άλλες πληροφορίες. Τα αποτελέσματα που λάβαμε παρατίθενται στην (Εικόνα 14):



Εικόνα 14. Αποτελέσματα on-line ελέγχου με το "Netcraft" στον ιστοχώρο HTTP://dsep.uop.gr

6.7.2. Εντοπισμός εφαρμογών, Αρ. Αναφοράς OWASP-IG-005

Ένας άλλος βασικός έλεγχος που πραγματοποιήσαμε κατά τις δοκιμές, ήταν ο έλεγχος των ανοιχτών θυρών και των πρωτοκόλλων, στον εξυπηρέτη του ιστοχώρου HTTP://dsep.uop.gr. Η εξακρίβωση των ανοιχτών θυρών καθώς και οι εκδόσεις των πρωτοκόλλων που «τρέχουν» σε ένα εξυπηρέτη είναι πολύ σημαντική για την αξιολόγηση της ασφάλειας, διότι πολλές εφαρμογές – και ιδιαίτερα συγκεκριμένες εκδόσεις αυτών- έχουν τρωτά σημεία και γνωστές στρατηγικές επίθεσης είναι διαθέσιμες για την εκμετάλλευση τους από κακόβουλους χρήστες. Με την αξιοποίηση των πληροφοριών για τα τρωτά σημεία των εφαρμογών, μπορεί ένας επιτιθέμενος να αποκτήσει απομακρυσμένο έλεγχο ή να εκμεταλλευτεί την εφαρμογή και τα δεδομένα της. Έτσι, χρησιμοποιήθηκε το εργαλείο "Nmap" προκειμένου να ανακαλύψουμε τις επιθυμητές πληροφορίες στον ιστοχώρο του Τμήματος. Στην (Εικόνα 15) παρατίθενται τα αποτελέσματα που λάβαμε.



Nmap Scan Report - Scanned at Wed Sep 04 13:18:11 2013

Scan Summary | **etl.korinthos.uop.gr (195.251.46.[redacted])**

Scan Summary

Nmap 6.40 was initiated at Wed Sep 04 13:18:11 2013 with these arguments:
nmap -T4 -A -v 195.251.46[redacted]

Verbosity: 1; Debug level 0

195.251.46[redacted] / etl.korinthos.uop.gr

Address

- 195.251.46[redacted] - (ipv4)
- 00:19:B9:CC[redacted] - Dell (mac)

Hostnames

- etl.korinthos.uop.gr (PTR)

Ports

The 998 ports scanned but not shown below are in state: **filtered**

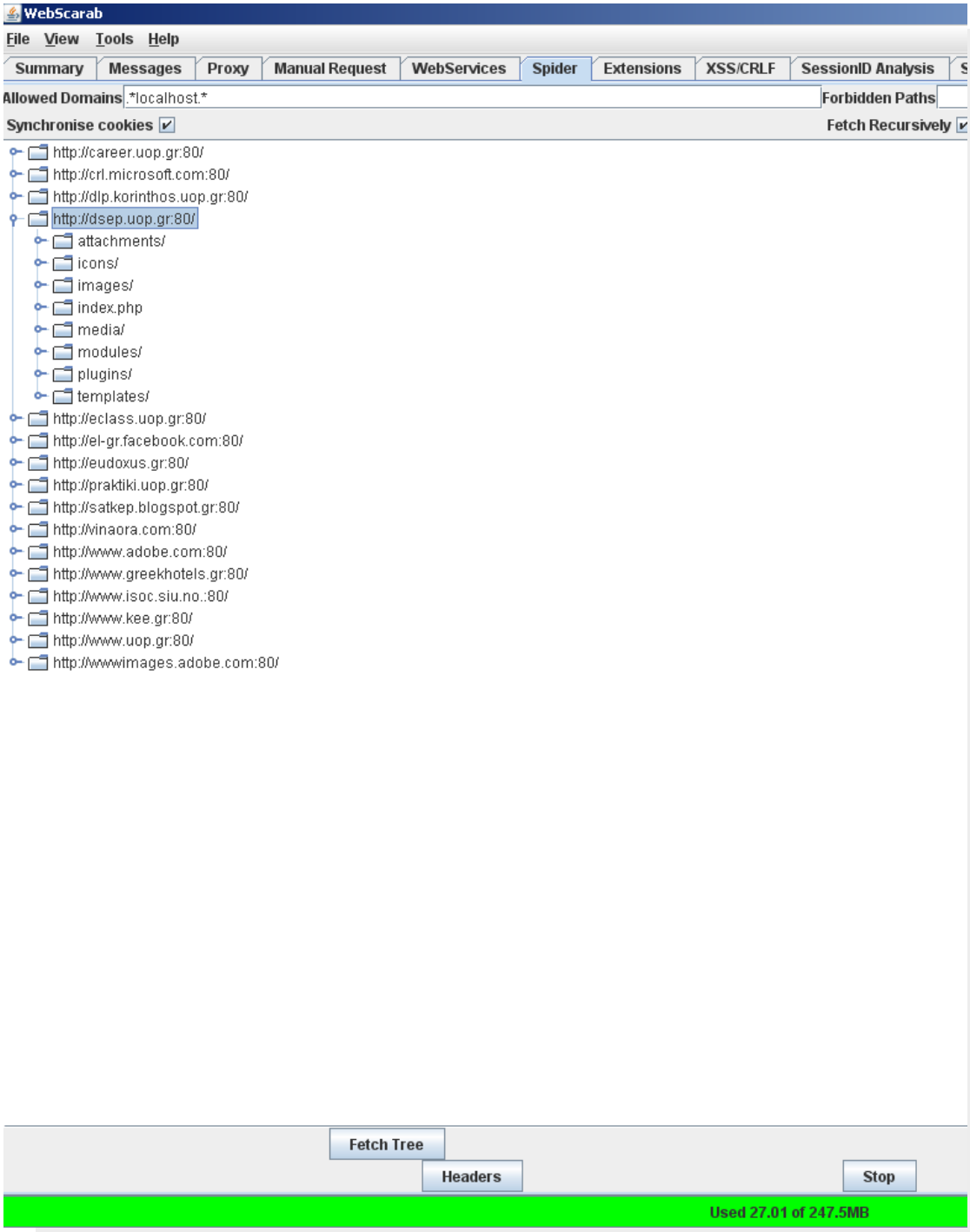
Port		State (toggle closed [0] | filtered [0])	Service	Reason	Product	Version	Extra info
80	tcp	open	http	syn-ack	Apache httpd	2.[redacted]	(Ubuntu) PHP/5[redacted]
10000	tcp	open	http	syn-ack	MiniServ	1.[redacted]	Webmin httpd

Εικόνα 15. Αποτελέσματα ελέγχου με το "Netmap" στον ιστοχώρο HTTP://dsep.uop.gr

6.7.3. Spiders, Robots and Crawlers, Αρ. Αναφοράς OWASP-IG-001

Κατά τη διενέργεια των ελέγχων αυτών, ανιχνεύουμε όλες τις συνδέσεις που υπάρχουν σε μια σελίδα, με στόχο να δημιουργήσουμε έναν χάρτη της εφαρμογής ο οποίος θα περιλαμβάνει όλα τα σημεία πρόσβασης σε αυτή.

Χρησιμοποιήσαμε το πρόσθετο του Webscarab "Spider" και τα αποτελέσματα παρατίθενται στην (Εικόνα 16).



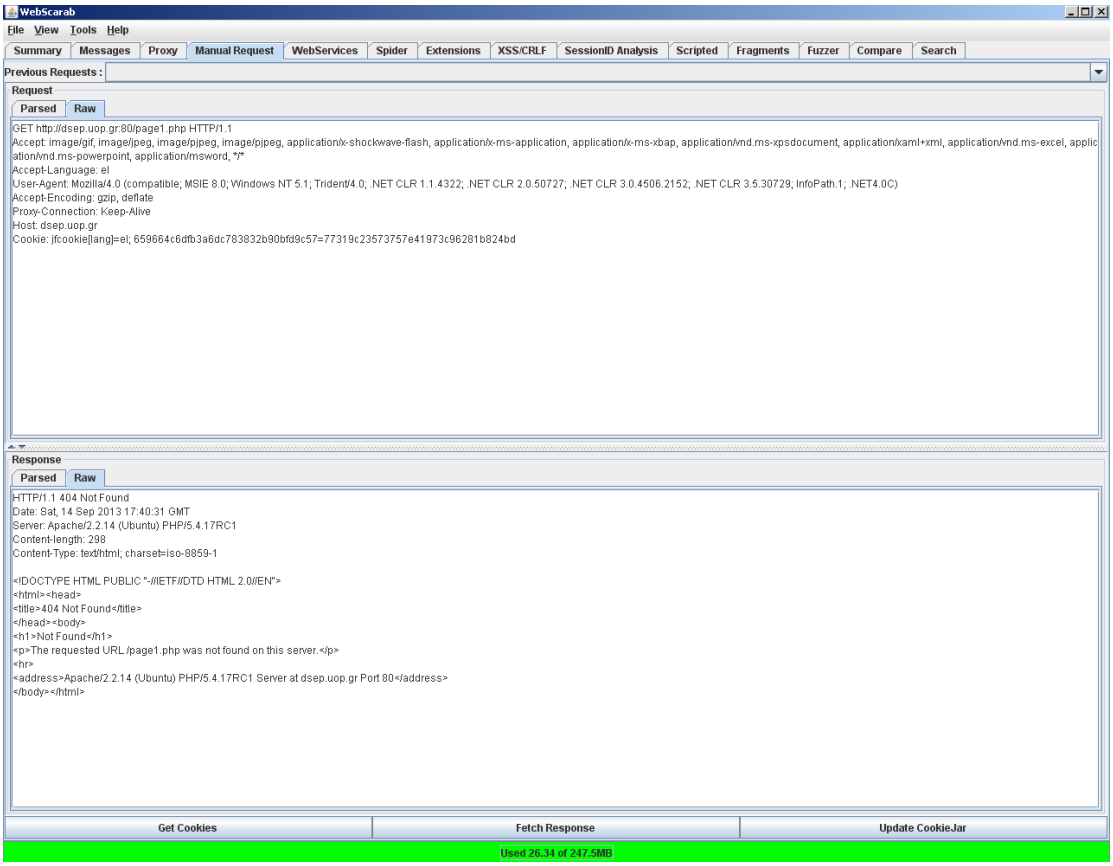
Εικόνα 16. Πληροφορίες των συνδέσεων του ιστοχώρου HTTP://dsep.uop.gr όπως φαίνονται από το πρόσθετο "Spider" του Webscarab

6.7.4. Έλεγχος για λάθη στον κώδικα, Αρ. Αναφοράς OWASP-IG-006

Συχνά, κατά την διάρκεια μιας δοκιμής διείσδυσης, ερχόμαστε αντιμέτωποι με αρκετά λάθη στον κώδικα της εφαρμογής ή του εξυπηρέτη ιστού. Είναι πιθανό να προκαλέσουμε τέτοιου είδους λάθη χρησιμοποιώντας συγκεκριμένο αίτημα, είτε χειροκίνητα είτε με εργαλεία, ειδικά για το σκοπό αυτό. Τα συμπεράσματα που αποκομίζουμε με το συγκεκριμένο έλεγχο είναι χρήσιμα επειδή αποκαλύπτουν πολλές πληροφορίες σχετικά με τις βάσεις δεδομένων, με τα σφάλματα των εφαρμογών (bugs), και με άλλα στοιχεία που συνδέονται άμεσα με τις εφαρμογές

ιστού. Ένα κοινό λάθος είναι η αναζήτηση μιας λάθους τοποθεσίας που έχει ως αποτέλεσμα την απάντησης “HTTP 404 Not Found” από την εφαρμογή ιστού. Αυτό μπορεί να είναι ένα κοινό μήνυμα, αλλά μας δείχνει αρκετές πληροφορίες σχετικά με το λειτουργικό σύστημα, την έκδοσης του εξυπηρέτη ιστού κ.ά.

Στον έλεγχο που πραγματοποιήσαμε αιτηθήκαμε την ανύπαρκτη σελίδα *page1.php.* Το αίτημα που δόθηκε ήταν GET HTTP://dsep.uop.gr:80/page1.php. Η εφαρμογή αποκρίθηκε σωστά, και η εφαρμογή ιστού αποκρίθηκε με μήνυμα “404: Not Found”. Στην (Εικόνα 17) φαίνεται το αποτέλεσμα του αιτήματος της ανύπαρκτης ιστοσελίδας.



Εικόνα 17. Αποτελέσματα χειροκίνητου ελέγχου με το “Webscarab” κατά το αίτημα ανύπαρκτης ιστοσελίδας

6.8. Έλεγχος Αυθεντικοποίησης (Authentication Testing)

Η αυθεντικοποίηση, εξαρτάται από έναν ή περισσότερους παράγοντες ελέγχου ταυτότητας. Στην ασφάλεια των υπολογιστών, ο έλεγχος της ταυτότητας είναι η διαδικασία για την προσπάθεια επιβεβαίωσης της ψηφιακής ταυτότητας του αποστολέα μιας επικοινωνίας. Ένα παράδειγμα τέτοιας διαδικασίας είναι η διαδικασία σύνδεσης (login). Εξετάζοντας το σχήμα ελέγχου ταυτότητας σημαίνει πως κατανοούμε τον τρόπο λειτουργίας αυτής της διαδικασίας και μπορούμε να χρησιμοποιήσουμε τις πληροφορίες που αποκομίσαμε για να προσπαθήσουμε παρακάμψουμε τον μηχανισμό αυτό.

6.8.1. Έλεγχος για επίθεση ωμής βίας, Αρ. Αναφοράς OWASP-AT-004

Για να πραγματοποιήσουμε επίθεση ωμής βίας, σημαντικό είναι να ανακαλύψουμε το είδους της μεθόδου ελέγχου ταυτότητας που χρησιμοποιείται από την εφαρμογή, επειδή ανάλογα με τη μέθοδο διαφοροποιούνται και οι τεχνικές καθώς και τα εργαλεία που θα χρησιμοποιηθούν.

Οι πιο συνηθισμένοι μέθοδοι είναι:

- HTTP Authentication
- HTML Form-Based Authentication

Το Webscarab δεν μας παρέχει κάποιο εργαλείο για να εξακριβώσουμε τη μέθοδο που χρησιμοποιεί ένας ιστοχώρος, αλλά με μια απλή εξέταση του κώδικα μπορούμε να το διαπιστώσουμε.

Ο ιστοχώρος HTTP://dsep.uop.gr τον οποίο εξετάζουμε δεν διαθέτει λειτουργία αυθεντικοποίησης σε καμία λειτουργία του. Γι' αυτό το λόγο ο έλεγχος που κάναμε ήταν στον ιστοχώρο HTTP://mail.uop.gr. Από τον έλεγχο διαπιστώσαμε, όπως φαίνεται και στην (Εικόνα 18), πως το είδος ελέγχου ταυτότητας της εφαρμογής web mail του Πανεπιστημίου Πελοποννήσου είναι HTML Form-based Authentication.

```
<!-- IMP: Copyright 2001, The Horde Project. IMP is under the GPL. -->
<!-- Horde Project: http://horde.org/ | IMP: http://horde.org/imp/ -->
<!--    GNU Public License: http://www.fsf.org/copyleft/gpl.html   -->
<html lang="el-GR"><head>
<title>Mail :: Καλωσήρθατε στο UoP Webmail</title>
<link href="/css.php?app=imp" rel="stylesheet" type="text/css" />
</head>

<body onload="setFocus()">
<script language="JavaScript" type="text/javascript">
<!--

function setFocus()
{
    document.implogin.imapuser.focus();
}

function submit_login()
{
    if (document.implogin.server[document.implogin.server.selectedIndex].value.substr(0, 1) == "_") {
        return false;
    }
    if (document.implogin.imapuser.value == "") {
        alert('Παρακαλώ δώστε το ονομα και συνθηματικό');
        document.implogin.imapuser.focus();
        return false;
    } else if (document.implogin.pass.value == "") {
        alert('Παρακαλώ δώστε το ονομα και συνθηματικό');
        document.implogin.pass.focus();
        return false;
    } else {
        return true;
    }
}
//-->
</script>
```

Εικόνα 18. Εξέταση κώδικα της εφαρμογής Webmail του Πανεπιστημίου Πελοποννήσου

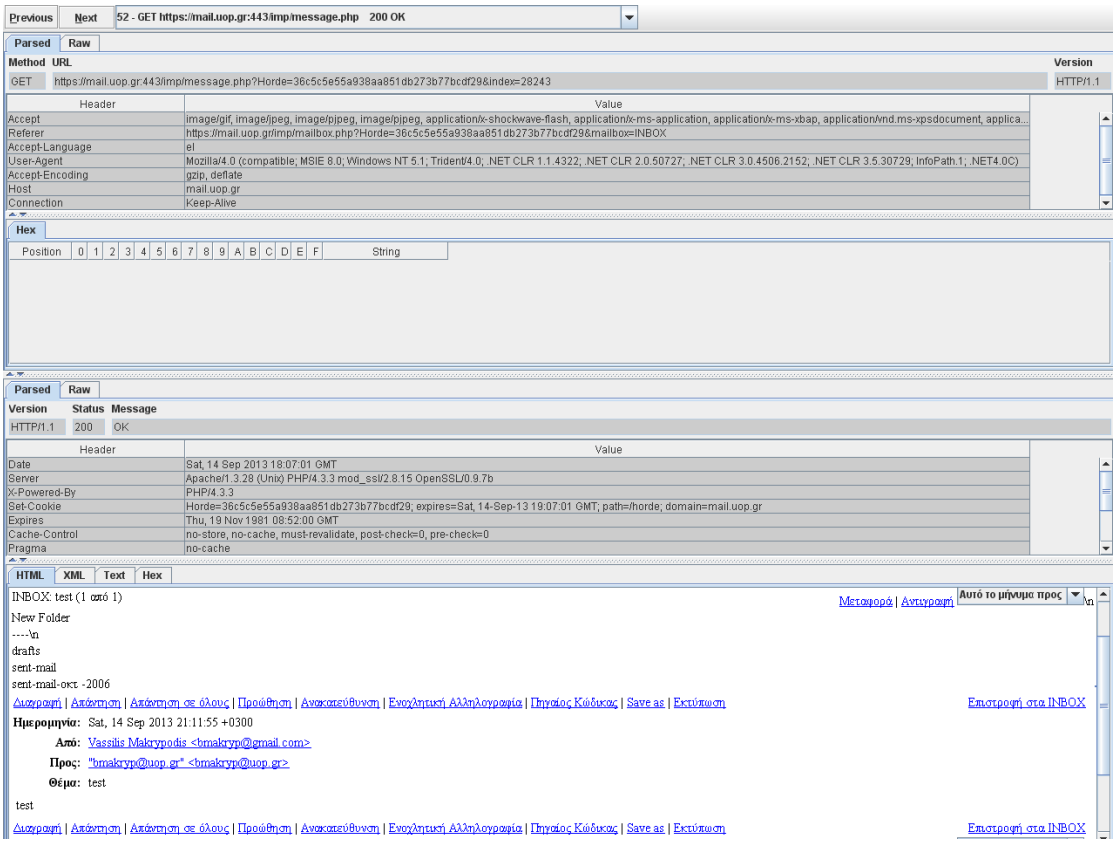
6.8.2. Έλεγχος για παράκαμψη της αυθεντικοποίησης, Αρ. Αναφοράς OWASP-AT-005

Οι περισσότερες εφαρμογές απαιτούν έλεγχο ταυτότητας στη σελίδα εισόδου προκειμένου ο χρήστης να αποκτήσει πρόσβαση σε ευαίσθητες πληροφορίες. Ωστόσο, υπάρχει πιθανότητα, από αμέλεια ή από κακή εκτίμηση των απειλών, κάποιος χρήστης να ζητήσει μια εσωτερική σελίδα παρακάμπτοντας τη αρχική

σελίδα εισόδου. Στα πλαίσια της εργασίας αυτής, θα πραγματοποιήσουμε τους παρακάτω ελέγχους παράκαμψης.

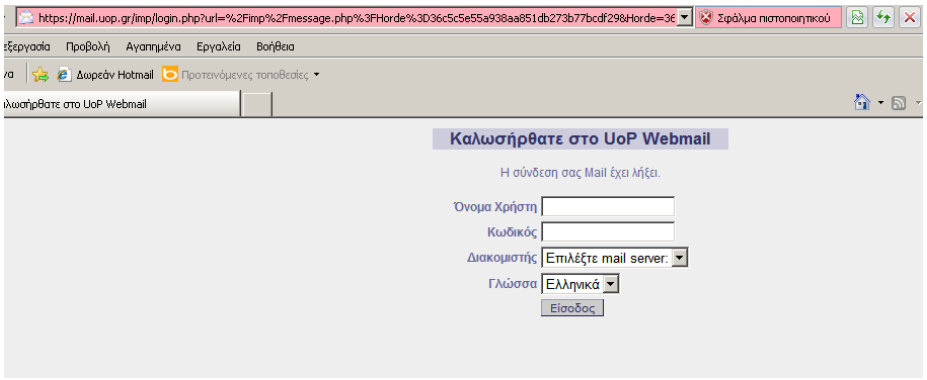
- **Άμεσο αίτημα σελίδων** *(Direct page request)*. Στην περίπτωση αυτή, αν ο χρήστης αιτηθεί μια επόμενη σελίδα από την σελίδα εισόδου και το σύστημα θεωρήσει ότι ο χρήστης έχει ήδη πιστοποιηθεί και του δώσει πρόσβαση, τότε θα έχει παρακάμψει τον έλεγχο εισόδου.
- **Πρόβλεψη αναγνωριστικών συνόδου** (Session ID Prediction). Κάποιες εφαρμογές ιστού, διαχειρίζονται την αυθεντικοποίηση χρησιμοποιώντας αναγνωριστικά περιόδου (session ID's). Αν τα αναγνωριστικά περιόδου είναι προβλέψιμα, είναι πιθανό κάποιος κακόβουλος χρήστης να βρει ένα έγκυρο αναγνωριστικό και να αποκτήσει μη εξουσιοδοτημένη πρόσβαση στην εφαρμογή.
- **Έκχυση SQL σε φόρμα αυθεντικοποίησης** (SQL injection). Μια ευρέως γνωστή τεχνική επίθεσης κατά την οποία ο κακόβουλος χρήστης με έκχυση SQL μπορεί να παρακάμψει την HTML φόρμα αυθεντικοποίησης.

Για να ελέγξουμε αν στην εφαρμογή Web Mail HTTP://mail.uop.gr, υπάρχει πρόβλημα στην ευπάθεια **άμεσο αίτημα σελίδων**, κάναμε είσοδο με ένα κανονικό λογαριασμό χρήστη και πιστοποιηθήκαμε. Επιλέξαμε ένα εισερχόμενο email *(https://mail.uop.gr:443/imp/message.php?Horde=36c5c5e55a938aa851db273b77bcdf29&index=28243)* και παρακολουθήσαμε την συνομιλία, μέσα από το Webscarab όπως φαίνεται στην (Εικόνα 19). Στη συνέχεια κάναμε έξοδο από την εφαρμογή του Web Mail και ολοκληρώσαμε τη διαδικασία αποσύνδεσης από αυτή την σύνοδο.



Εικόνα 19. Παρακολούθηση συνομιλίας μέσα από το περιβάλλον του Webscarab.

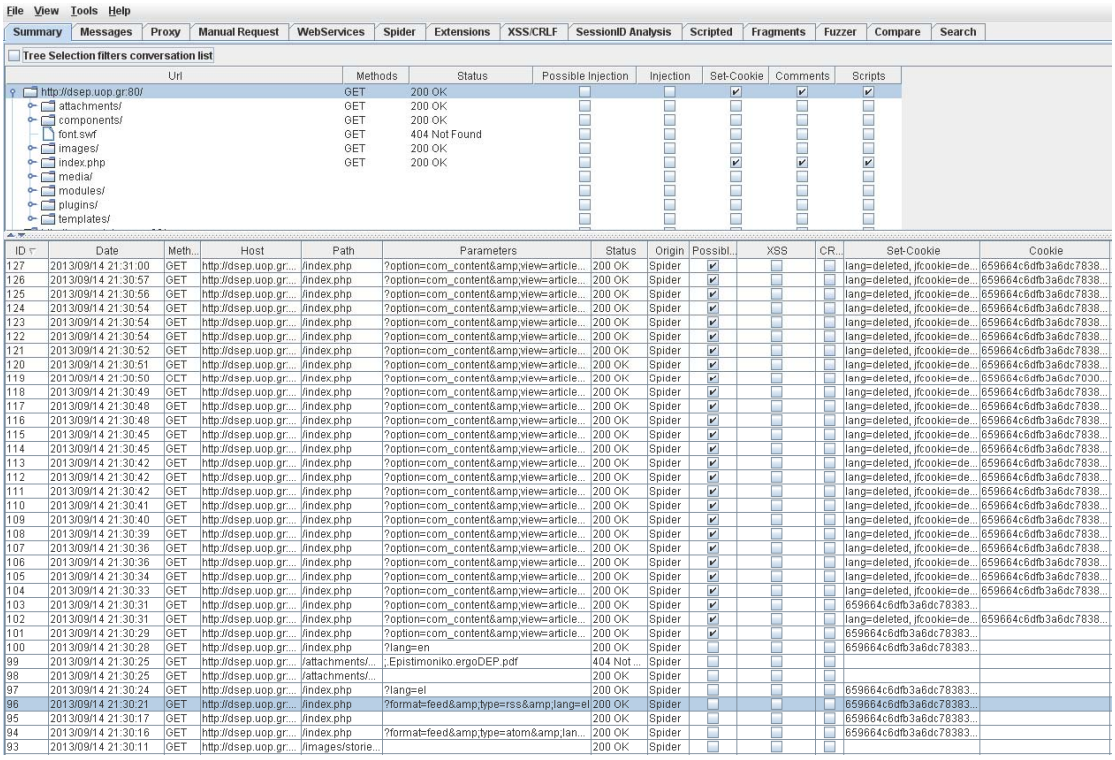
Έχοντας πλέον αποσυνδεθεί από την εφαρμογή του Web Mail, πληκτρολογούμε στην εφαρμογή ιστού, το URL του email που λάβαμε στην προηγούμενη σύνοδο. Το αποτέλεσμα ήταν να εμφανιστεί στην οθόνη μας το μήνυμα «Η σύνδεση σας Mail έχει λήξει» όπως αυτό φαίνεται και στην (Εικόνα 20) που σημαίνει πως δεν καταφέραμε να παρακάμψουμε την οθόνη αυθεντικοποίησης της εφαρμογής και πως πρέπει να δώσουμε ξανά τα διαπιστευτήρια μας για νέα είσοδο.



Εικόνα 20. Αποτυχημένη απόπειρα σύνδεσης κατά τον έλεγχο της παράκαμψης αυθεντικοποίησης

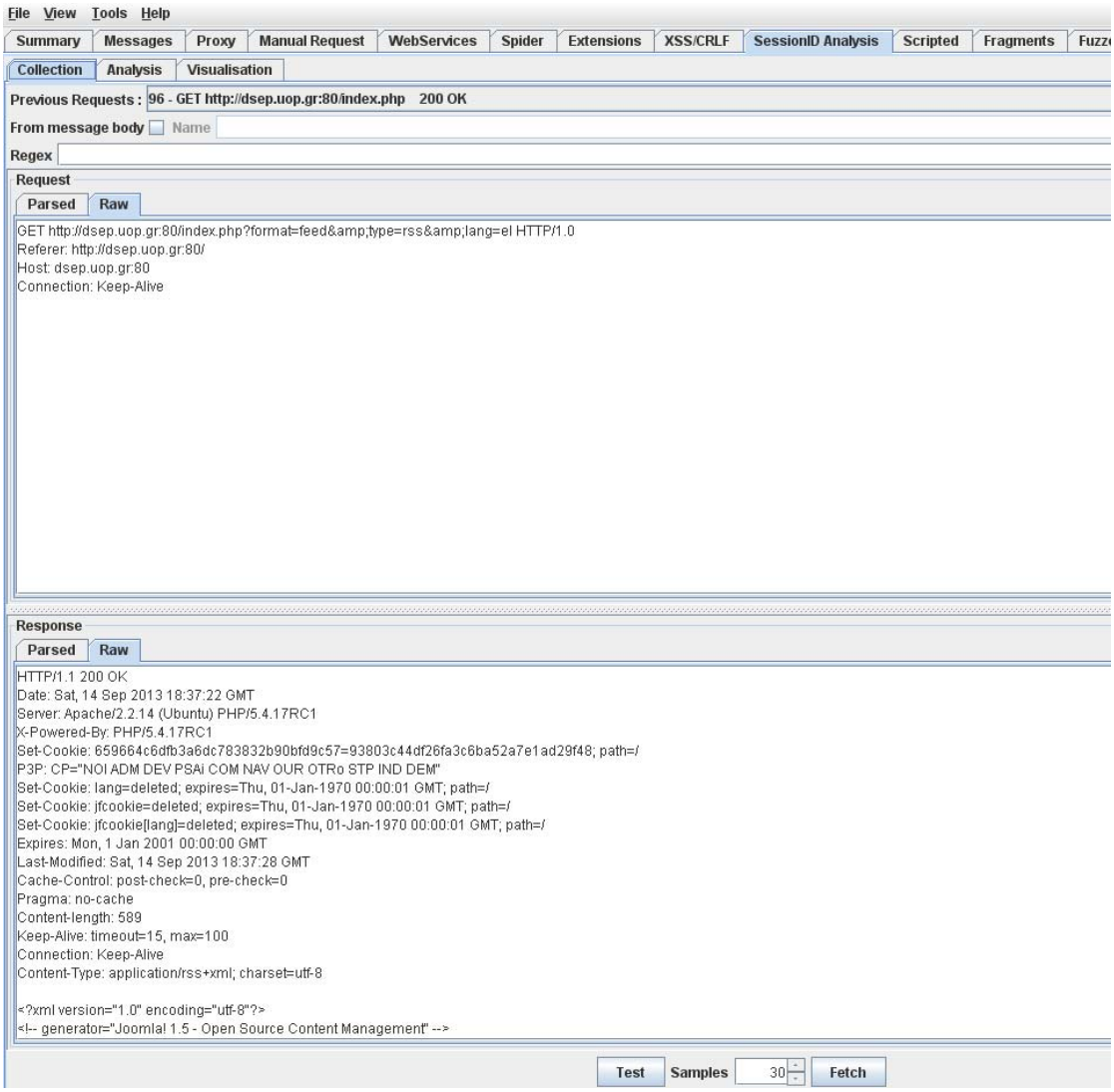
Κατά το έλεγχο που πραγματοποιήσαμε για να διαπιστώσουμε αν ο ιστοχώρος έχει **προβλέψιμα αναγνωριστικά συνόδου** διαπιστώσαμε πως αυτό δεν ισχύει.

Συλλέξαμε μερικά *Session Id's* τα οποία περιλαμβάνουν cookies και ειδικότερα από αυτές τις συνομιλίες οι οποίες έχουν συμπληρωμένες τις τιμές στη στήλη "Set-cookie" όπως φαίνεται στην (Εικόνα 21).



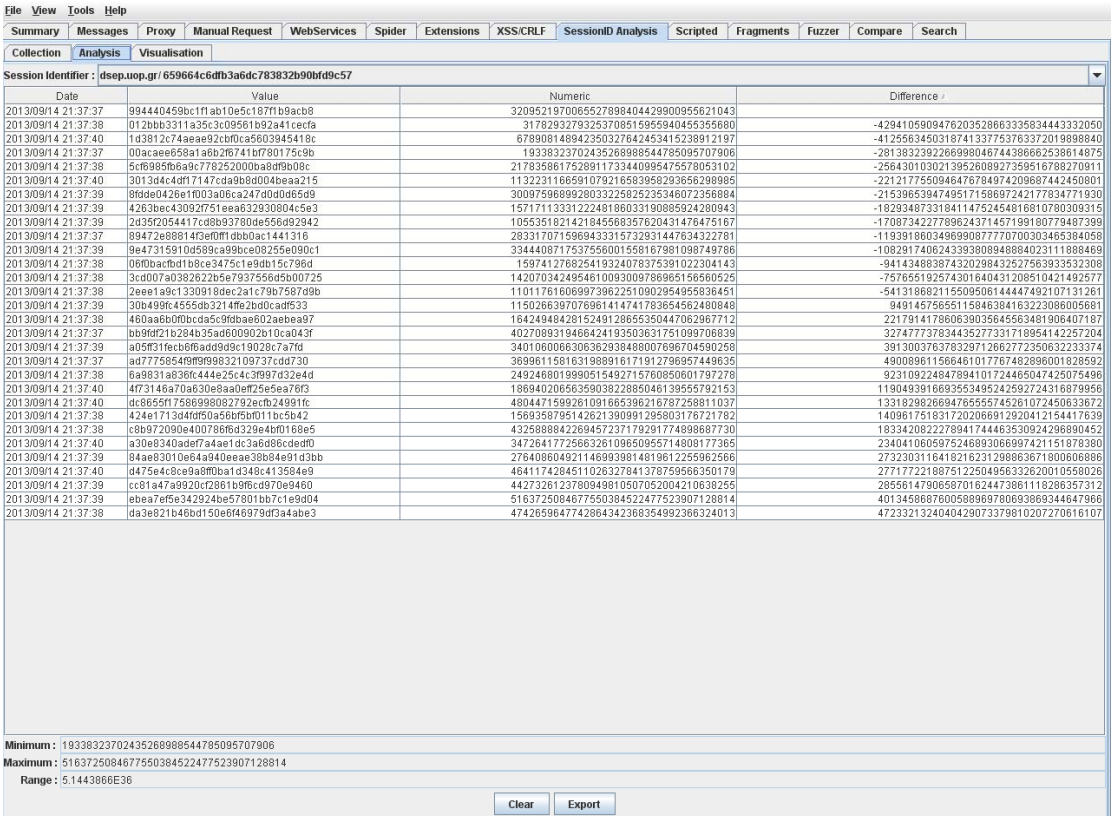
Εικόνα 21. Συνομιλίες οι οποίες περιλαμβάνουν cookies.

Αφού εντοπίσαμε τη συνομιλία που θέλαμε να αναλύσουμε, η οποία περιείχε τα χαρακτηριστικά που αναφέραμε, μεταβήκαμε στο πρόσθετο "SesionID Analysis" και επιλέξαμε την καρτέλα "Collection". Διαμορφώσαμε τον αριθμό των δειγμάτων που θέλαμε, στην προκειμένη περίπτωση 30, και επιλέξαμε "test" για να εξαχθεί το αποτέλεσμα στα υπόλοιπα εργαλεία όπως φαίνεται στην (Εικόνα 22).



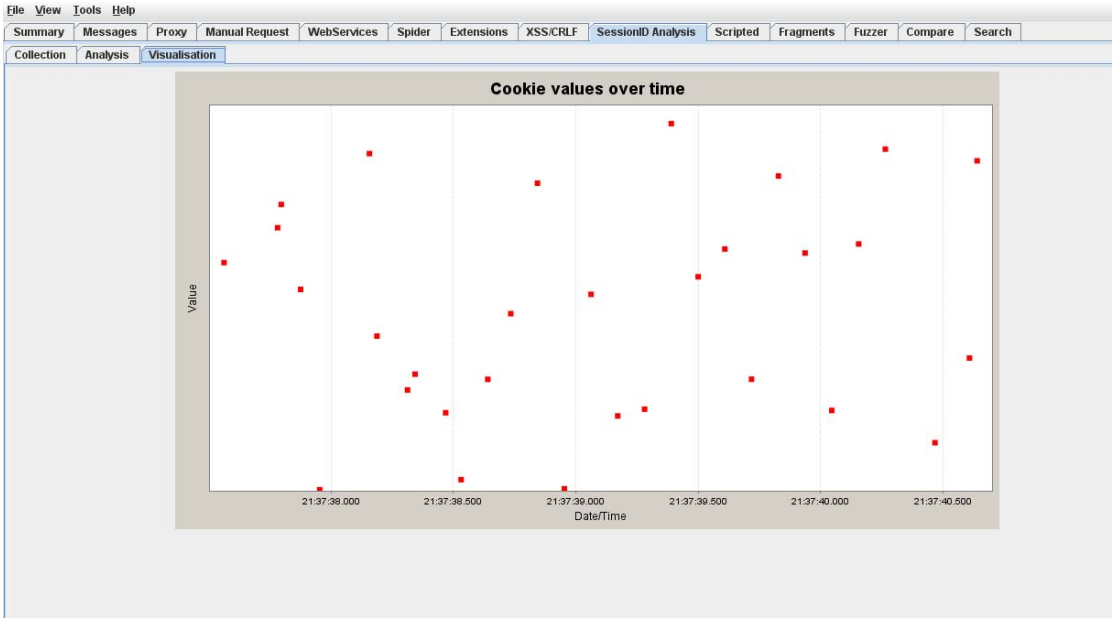
Εικόνα 22. Το πρόσθετο SessionID Analysis σε συγκεκριμένη συνομιλία

Μεταβαίνοντας στην καρτέλα “Analysis” (Εικόνα 23), μπορούμε να παρατηρήσουμε μερικά χαρακτηριστικά από τα cookies που συλλέχθηκαν, όπως την ημερομηνία και ώρα, την πραγματική και αριθμητική τιμή και την διαφορά αυτών.



Εικόνα 23. Αποτελέσματα από την ανάλυση των αναγνωριστικών περιόδου

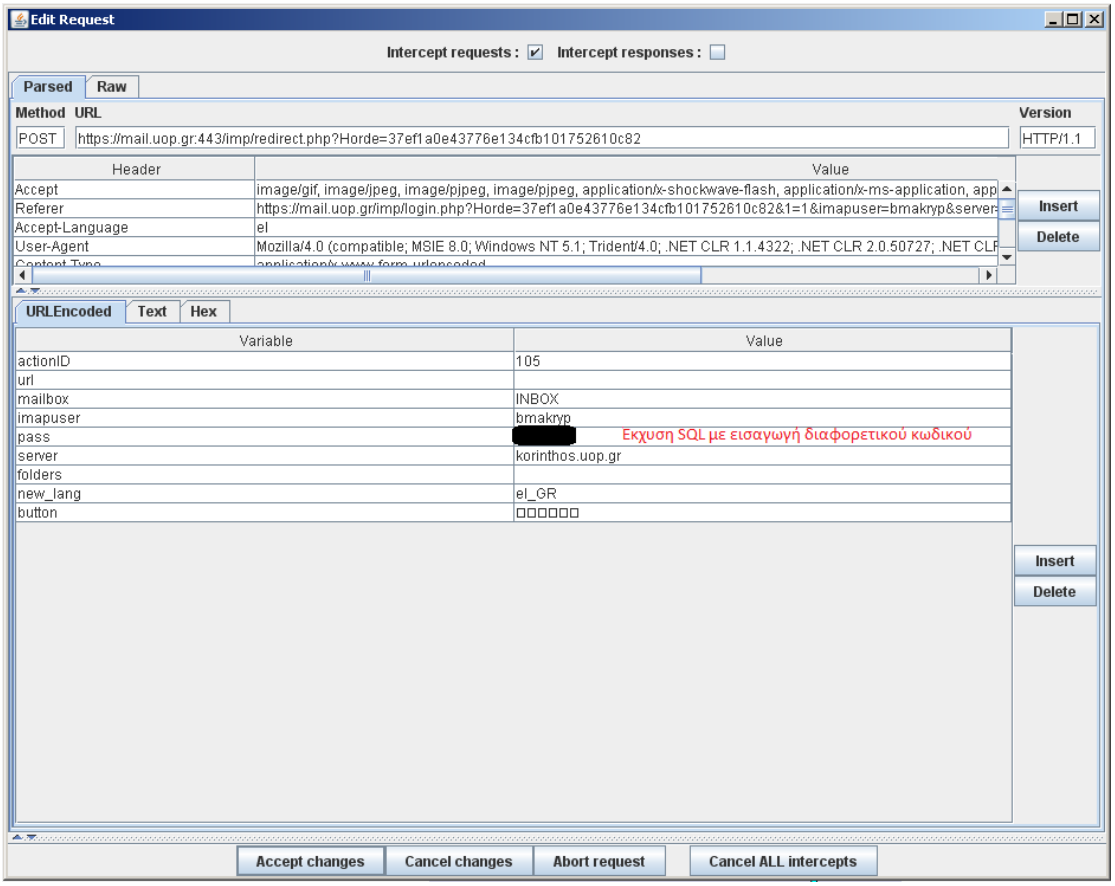
Μεταβαίνοντας στην τρίτη και τελευταία καρτέλα του πρόσθετου, την καρτέλα “Visualization”, (Εικόνα 24) λαμβάνουμε και σε μορφή γραφήματος πλέον, τα προηγούμενα αποτελέσματα. Εδώ βλέπουμε εύκολα, την τυχαία σχέση που υπάρχει στα αναγνωριστικά περιόδου, αποκλείοντας με αυτό τον τρόπο την τυχαία πρόβλεψη, η οποία θα ήταν πιθανή σε περίπτωση, που παρατηρούσαμε γραμμική ή επαναληπτική σχέση.



Εικόνα 24. Γράφημα των τιμών των cookies συναρτήσει του χρόνου

Για να εξακριβώσουμε αν μπορούμε να **παρακάμψουμε μια HTML φόρμα αυθεντικοποίησης**, αρχικά πραγματοποιήσαμε είσοδο στην εφαρμογή Web Mail.

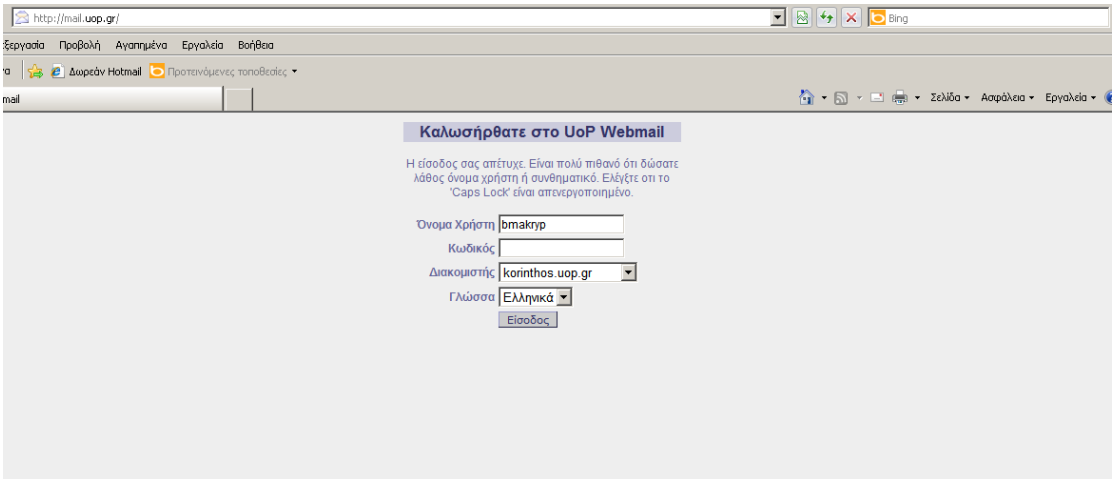
Μέσα από τις οθόνες διαλόγων του Webscarab παρατηρήσαμε τα στοιχεία που δώσαμε κατά την διαδικασία εισόδου, που μόλις πραγματοποιήσαμε. Στην καρτέλα "Edit request" (Εικόνα 25) η οποία μας επιτρέπει να τροποποιήσουμε κάποιο αίτημα, παραποιήσαμε τον κωδικό πρόσβασης που πληκτρολογήσαμε, με κάποιο άλλο διαφορετικό.



Εικόνα 25. Προσπάθεια πραγματοποίησης έκχυσης SQL

Αφού αλλάξαμε τον κωδικό πρόσβασης, επιλέξαμε το "Accept changes" για να επικυρώσουμε το αίτημα μας στην προσπάθεια σύνδεσης με την υπηρεσία. Το αποτέλεσμα που λάβαμε στην απάντηση του εξυπηρέτη, όπως φαίνεται στην (Εικόνα 26), ήταν η απαγόρευση εισόδου σε αυτόν, με αίτιο, αυτό του πιθανού λάθος στον κωδικό πρόσβασης.

Από όλα τα παραπάνω μπορούμε να συμπεράνουμε πως η εφαρμογή δεν εμφανίζει αυτή την ευπάθεια.

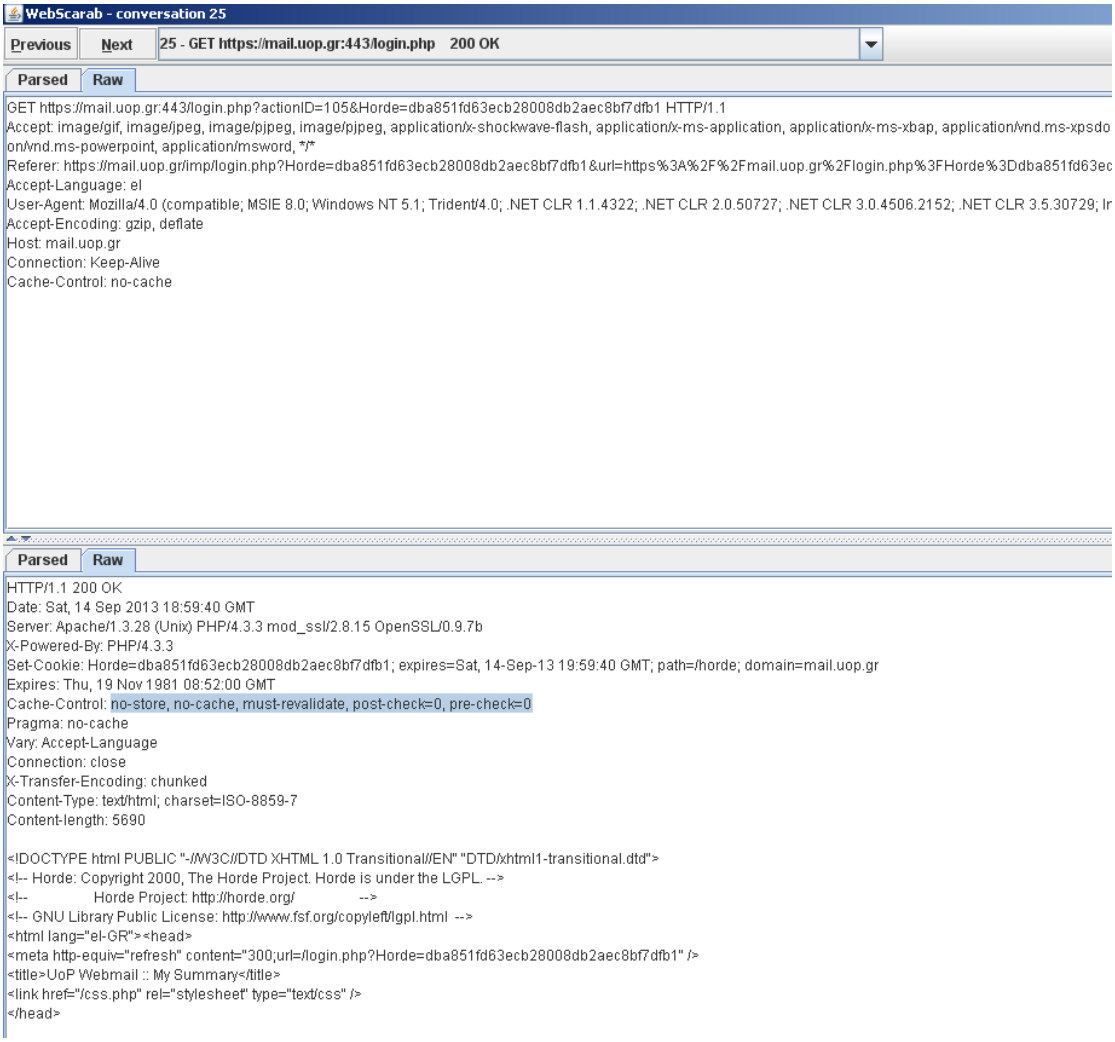


Εικόνα 26. Άρνηση πρόσβασης στην εφαρμογή μετά από έκχυση SQL

6.8.3. Έλεγχος για ευπάθεια απομνημόνευσης κωδικού, Αρ. Αναφοράς OWASP-AT-006

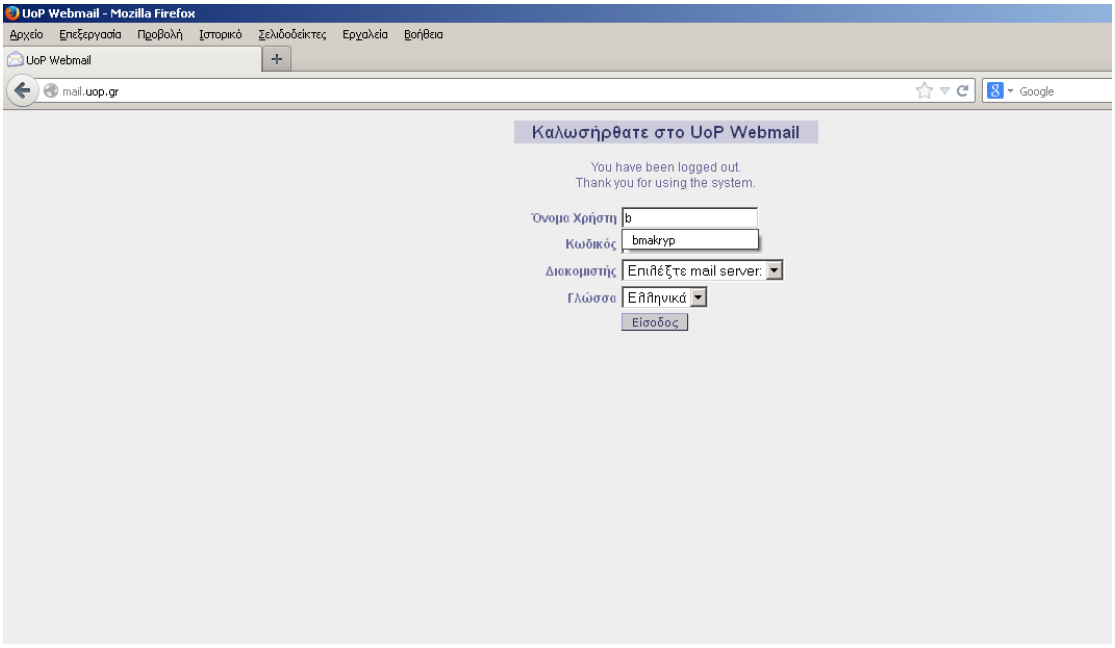
Ένα κοινό χαρακτηριστικό των εφαρμογών ιστού οι οποίες παρέχουν υπηρεσία ταυτοποίησης χρηστών, είναι η δυνατότητα να παρέχουν κάποιες λειτουργίες προκειμένου να κάνουν πιο εύκολη την πλοήγηση. Ένα τέτοιο χαρακτηριστικό, είναι η ερώτηση για απομνημόνευση τους κωδικού πρόσβασης *(password)* σε τοπικό επίπεδο και η αποθήκευση του ονόματος χρήστη *(username),* έτσι ώστε σε επόμενες επισκέψεις στον ιστοχώρο να δίνεται στο χρήστη προ-δακτυλογραφημένο. Αυτό, σαν χαρακτηριστικό είναι εξαιρετικά φιλικό για το μέσο χρήστη, ταυτόχρονα όμως εισάγει ένα ελάττωμα, γιατί ο λογαριασμός χρήστη γίνεται εύκολα προσβάσιμος από οποιονδήποτε που χρησιμοποιεί τη συγκεκριμένη ηλεκτρονική συσκευή.

Για να Ελέγξουμε την εφαρμογή Web Mail του Πανεπιστημίου κάναμε εισαγωγή των στοιχείων του λογαριασμού και πραγματοποιήσαμε είσοδο στην εφαρμογή. Ελέγχοντας μέσω του Webscarab τη συνδιάλεξη διαδικασίας εισόδου, παρατηρήσαμε (Εικόνα 27) πως στον κώδικα δεν υπάρχει κάποια δήλωση για την απαγόρευση της αποθήκευσης του ονόματος χρήστη και του κωδικού πρόσβασης στην εφαρμογή. Υπάρχουν δε, οι εξής παράμετροι στο cache-control. **No-store, no-cache, must-revalidate, post-check=0, pre-check=0.**



Εικόνα 27. Παρατήρηση του cache-control της εφαρμογής που παρατηρούμε

Στη συνέχεια κάναμε έξοδο και πραγματοποιήσαμε ξανά τη διαδικασία επανασύνδεσης. Παρατηρήσαμε (Εικόνα 28) πως στο όνομα χρήση υπάρχει προ-συμπληρωμένο το όνομα χρήστη που είχαμε δώσει στη προηγούμενη συνεδρία.

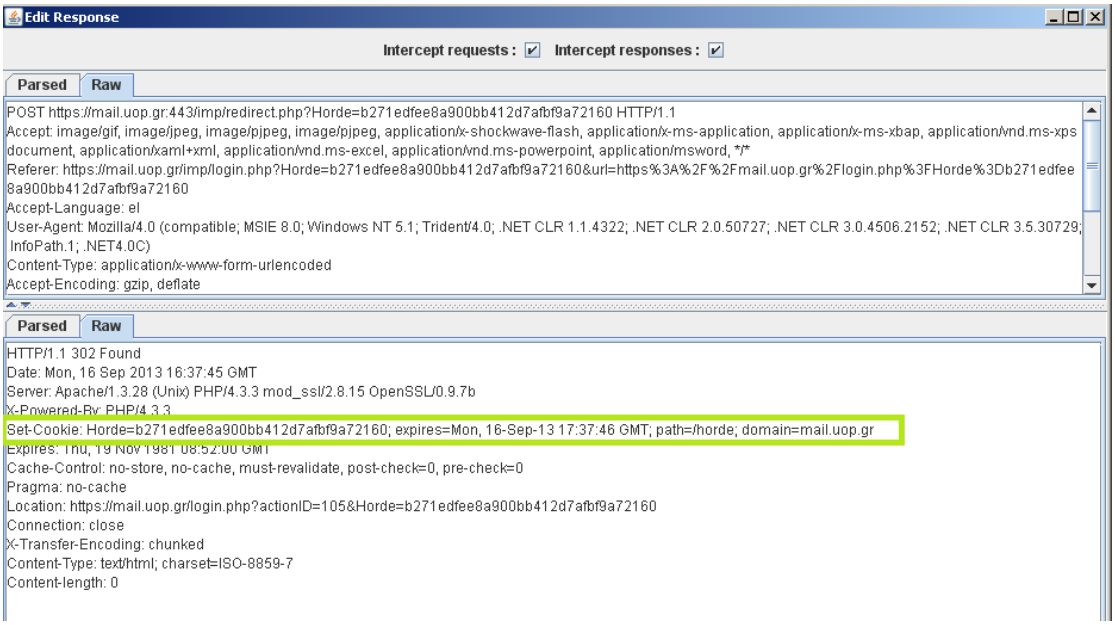


Εικόνα 28. Δυνατότητα αποθήκευσης του ονόματος χρήστη, στην εφαρμογή ιστού του χρήστη

6.8.4. Έλεγχος για αποσύνδεση χρήστη και διαχείριση προσωρινής μνήμης, Αρ. Αναφοράς OWASP-AT-007

Στον έλεγχο αυτό, θα δούμε αν γίνεται σωστή έξοδος από την εφαρμογή Web Mail και θα παρατηρήσουμε τη συμπεριφορά των cookies σχετικά με την λήξη τους και την δυνατότητα επαναχρησιμοποίησης τους.

Κατά τον έλεγχο, πραγματοποιήσαμε είσοδο στην εφαρμογή και σημειώσαμε το cookie της συνόδου (Εικόνα 29) μέσα από το webscarab.



Εικόνα 29. Έλεγχος αποσύνδεσης χρήστη με επαναχρησιμοποίηση cokkie.

Κάναμε επιτυχή έξοδο και επαναλάβαμε νέα είσοδο στέλνοντας στον εξυπηρέτη την τιμή από το cookie της προηγούμενης συνόδου, πραγματοποιώντας επιτυχή είσοδο. Διερευνώντας τα αίτια, παρατηρήσαμε πως το cookie μου μας έστειλε ο εξυπηρέτης έχει ισχύ μιας ώρας. Επαναλάβαμε την προσπάθεια μετά από την προβλεπόμενη ώρα λήξης του cookie και είδαμε πως η είσοδος πραγματοποιήθηκε πάλι.

6.9. Έλεγχος επικύρωσης δεδομένων

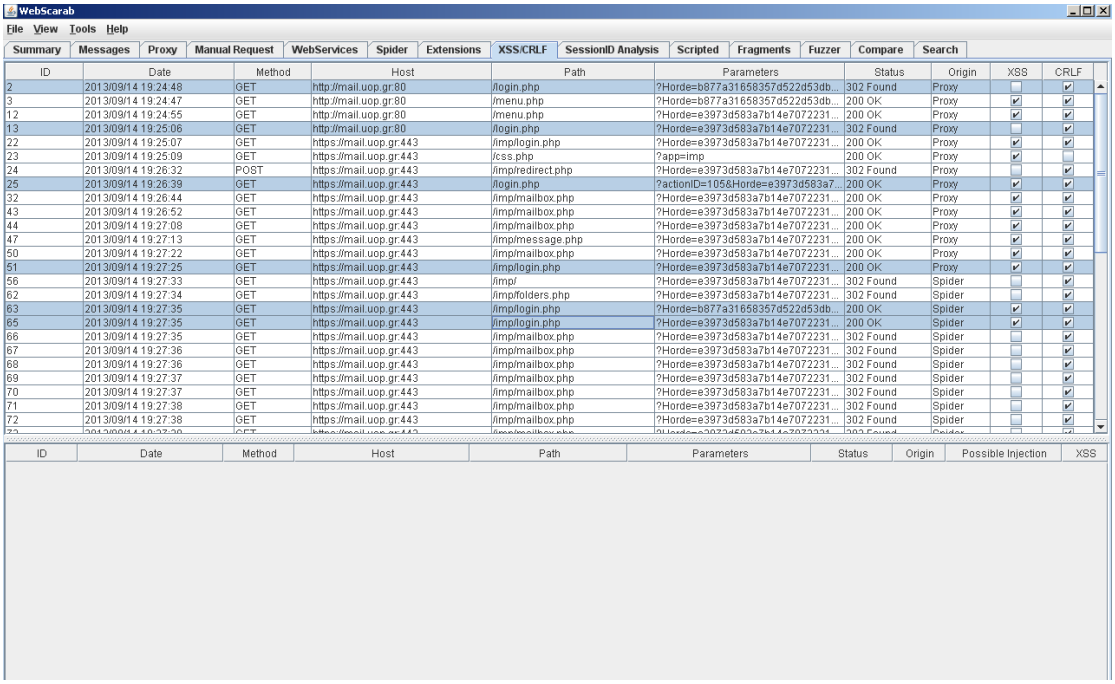
Η πιο κοινή ευπάθεια των εφαρμογών ιστού είναι η αδυναμία για την ορθή επικύρωση των εισερχομένων δεδομένων από το χρήση ή από κάποια εξωτερική οντότητα, πριν αυτά χρησιμοποιηθούν. Η ευπάθεια αυτή οδηγεί σχεδόν σε όλες τις μεγάλες ευπάθειες εφαρμογών ιστού, όπως η χρήση scripts μεταξύ πολλαπλών ιστοχώρων *(cross-site scripting)*, έκχυση SQL *(SQL injection),* έκχυση στο διερμηνευτή *(interpreter injection)*, επιθέσεις στο σύστημα αρχείων, και υπερχειλίσεις ενδιάμεσης μνήμης *(buffer overflows).* Δεδομένα από μια εξωτερική οντότητα ή από τον χρήστη ποτέ δεν πρέπει να θεωρηθούν αξιόπιστα, δεδομένου ότι μπορεί να έχουν παραποιηθεί αυθαίρετα από ένα εισβολέα.

Στη συνέχεια παρουσιάζονται μερικές δοκιμές ελέγχου στην εφαρμογή ηλεκτρονικού ταχυδρομείου του Πανεπιστημίου Πελοποννήσου για να δούμε αν επικυρώνει επαρκώς τα δεδομένα εισόδου πριν τα χρησιμοποιήσει.

6.9.1. Έλεγχος για Cross Site Scripting, Αρ. Αναφοράς OWASP-DV-001

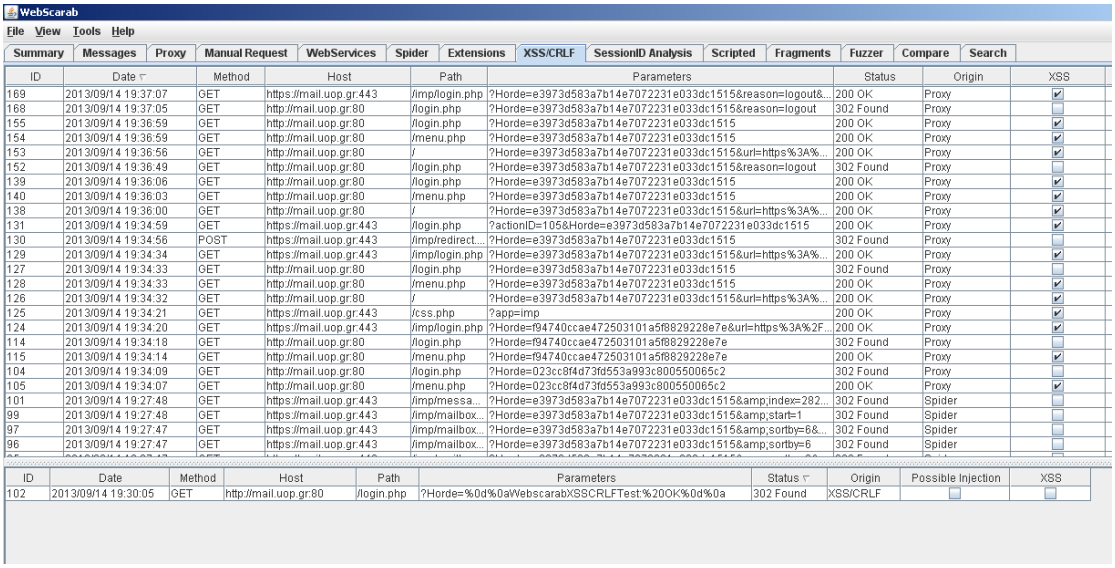
Η εφαρμογή που χρησιμοποιούμε στις δοκιμές μας, το Webscarab, σε συνδυασμό με το πρόσθετο που διαθέτει το ”XSS/CRLF” είναι σε θέση να αναγνωρίσει ποια αιτήματα μας προς τον εξυπηρέτη είναι ευπαθή στην εν λόγω ευπάθεια. Όπως αναφέραμε στην περιγραφή του προσθέτου, είναι σε θέση εξετάζει ύποπτα HTTP αιτήματα για XSS και CRLF και να αναλύει παθητικά, όλες τις συνομιλίες που περνούν από αυτό.

Στείλαμε αρκετά αιτήματα στον εξυπηρέτη του Web Mail του Πανεπιστημίου και διαπιστώσαμε (Εικόνα 30) πως υπάρχουν μερικά αιτήματα τα οποία τα χαρακτηρίζει ως ύποπτα για την ευπάθεια.



Εικόνα 30. Έλεγχος για XSS ευπάθεια με το Webscarab

Στη συνέχεια επιλέξαμε μια από τις συνομιλίες που συλλέξαμε και πατήσαμε την επιλογή “check” προκειμένου να κάνουμε ενδελεχή έλεγχο της συνομιλίας. Το αποτέλεσμα που λάβαμε ήταν αρνητικό, που σημαίνει πως η εφαρμογή δεν είναι ευπαθής στην ευπάθεια XSS όπως φαίνεται και στην (Εικόνα 31).



Εικόνα 31. Αναλυτικός έλεγχος συγκεκριμένης συνομιλίας για την ευπάθεια XSS

Ο έλεγχος επαναλήφθηκε και στις άλλες συνομιλίες που εμφανίστηκαν ως ύποπτες και καμία από αυτές δεν κατέδειξε ευπάθεια XSS.

7. Επισκόπηση - Συμπεράσματα

Η μέτρηση ασφαλείας είναι μιας κρίσιμης σημασίας μέθοδος ελέγχου και μπορεί να προστατεύσει σε μεγάλο βαθμό τον αμυνόμενο. Η διαδικασία δεν είναι απλή. Απαιτεί κατάλληλο γνωστικό υπόβαθρο και αφορά τόσο τους υπεύθυνους ανάπτυξης των εφαρμογών ιστού, όσο και τους υπεύθυνους ασφαλείας. Η μέτρηση ασφάλειας έχει να κάνει με τον έλεγχο διαβλητότητας ενός πληροφοριακού συστήματος. Όμως, επειδή αυτή μπορεί να αποτελέσει κίνδυνο για την αποκάλυψη ευαίσθητων πληροφοριών, δεν είθισται να δημοσιεύονται τα αποτελέσματα αυτά σε μη εξουσιοδοτημένους χρήστες.

Υπάρχουν διάφοροι μέθοδοι για να πραγματοποιήσει ο διαχειριστής έναν έλεγχο ασφαλείας. Ένας από αυτούς και ίσως ο πιο αποτελεσματικός, είναι η επιθεώρηση του κώδικα της εφαρμογής. Κάτι τέτοιο όμως χρειάζεται λεπτομερή και μεθοδικό έλεγχο, ο οποίος απαιτεί μεγάλη χειρωνακτική προσπάθεια, σημαντικό υπόβαθρο γνώσεων και άφθονο χρόνο. Για αυτούς τους λόγους οι συγκεκριμένοι έλεγχοι αναλαμβάνονται από εταιρείες εξειδικευμένες στο αντικείμενο, ενέργεια που ανεβάζει το κόστος. Κατά συνέπεια, οι περισσότεροι οργανισμοί – επιχειρήσεις, καταφεύγουν στη λύση του ελέγχου ασφαλείας με τη μέθοδο της δοκιμής διείσδυσης.

Στην εργασία αυτή επιλέξαμε την μέθοδο black box για να κάνουμε μια δοκιμή διείσδυσης στους εξυπηρέτες του Τμήματος Κοινωνικής & Εκπαιδευτικής Πολιτικής. Όπως αναφέρθηκε και στο θεωρητικό υπόβαθρο της παρούσης, η μέθοδος αυτή αφορά στον έλεγχο ασφαλείας της εφαρμογής μέσω εξωτερικής διεπαφής, δηλαδή από χρήστες που δεν γνωρίζουν την δομή της υπό εξέταση εφαρμογής. Επίσης, λόγω της μεγάλης εξειδίκευσης που απαιτείται για χειροκίνητο έλεγχο, επιλέχτηκε ο έλεγχος με αυτοματοποιημένα εργαλεία. Για την επιλογή των εργαλείων για τον έλεγχο διείσδυσης, συνεκτιμήθηκε ότι πολλά εργαλεία είναι εμπορικά, κάποια άλλα είναι ανοιχτού κώδικα ή παρέχονται δωρεάν χωρίς όμως να έχουν καλή τεκμηρίωση των λειτουργιών τους, χωρίς επαρκείς οδηγίες για τη χρήση τους και με αμφισβητήσιμη αποτελεσματικότητα. Το Webscarab επιλέχτηκε διότι συνοδεύεται από καλή τεκμηρίωση, οδηγίες για τον τρόπο διεξαγωγής μερικών βασικών ελέγχων και υποστήριξη από τους υπεύθυνους ανάπτυξης, μέσω e-mail. Είναι ευέλικτο και αποτελείται από διάφορα πρόσθετα τα οποία πραγματοποιούν πλήρεις ελέγχους στις εφαρμογές ιστού.

Σύμφωνα με τις μετρήσεις του ελέγχου ασφαλείας που πραγματοποιήσαμε στον ιστοχώρο του Τμήματος Κοινωνικής & Εκπαιδευτικής Πολιτικής και στην εφαρμογή Web Mail του Πανεπιστημίου Πελοποννήσου, δεν παρουσιάστηκαν σημαντικές ευπάθειες, ικανές να βλάψουν τον ιστοχώρο και τις εφαρμογές που αυτός υποστηρίζει. Βέβαια, σε ένα ολοκληρωμένο έλεγχο πρέπει να πραγματοποιηθεί το σύνολο των δοκιμών, αλλά κάτι τέτοιο ήταν αδύνατο να πραγματοποιηθεί, διότι πρόκειται για εφαρμογές που χρησιμοποιούνται διαρκώς από τους φοιτητές και το προσωπικό του Τμήματος και σε καμία περίπτωση δεν θέλαμε να περιορίσουνε την απρόσκοπτη λειτουργία του ιστοχώρου και των λειτουργιών του. Τα αποτελέσματα συνοψίζονται στον (Πίνακας 4) και μπορούμε να παραθέσουμε ορισμένα βασικά συμπεράσματα που αποκομίσαμε. Πιο συγκεκριμένα:

Λειτουργικό Σύστημα εξυπηρέτη:	**Ubuntu**
Εξυπηρέτης ιστού	Apache 2.2.14
Διεύθυνση IP	195.251.46.12
PHP	5.4.17RC1
Δυναμική ιστοσελίδα	Ναι
Θύρες ανοιχτές εξυπηρέτη	80
Πραγματοποίηση χαρτογράφησης	Ναι
Αναγνώριση συνδέσμων ιστοχώρου	Ναι
Μηχανισμός Αυθεντικοποίησης	HTML form-based
Χρησιμοποίηση cookies	Ναι
Χρησιμοποίηση μόνιμων cookies	Όχι
Απομνημόνευση ονόματος χρήστη	Ναι
Απομνημόνευση κωδικού πρόσβασης	Ναι (κατόπιν ερώτησης αποδοχής από την εφαρμογή ιστού)
Έλεγχος για παράκαμψη αυθεντικοποίησης	Όχι (ανεπιτυχής προσπάθεια)
Ευπάθεια XSS	Όχι

Πίνακας 4. Βασικά αποτελέσματα του ελέγχου ασφάλειας

Έχοντας υπόψη όσα αναφέρθηκαν στην παρούσα εργασία, κάθε διαχειριστής συστήματος και μηχανικός πληροφοριακών συστημάτων μπορεί να ελέγχει κατά πόσο είναι ασφαλή τα δεδομένα που βρίσκονται αποθηκευμένα σε πληροφοριακά δίκτυα εταιρειών, Πανεπιστημίων κ.ά., που έχουν διεπαφή με τον παγκόσμιο ιστό. Η επικοινωνία με το διαδίκτυο, από καταβολής του, είναι αμφίδρομη. Αυτό πρέπει να υπάρχει στο μυαλό κάθε υπεύθυνου και δεν πρέπει ποτέ να παρασύρεται κανείς από υποσχέσεις εταιρειών, για προμήθεια προϊόντων με παροχή 100% ασφάλειας. Αυτό το ποσοστό είναι αδύνατο να υπάρξει. Τα εργαλεία ελέγχου από μόνα τους δεν μπορούν να εγγυηθούν καμία ασφάλεια. Ο διαχειριστής και εν τέλει ο αμυνόμενος, πρέπει να προσπαθήσει να γνωρίσει τα τρωτά σημεία του συστήματος του, να τα ελέγξει και να τα περιορίσει τείνοντας τα προς το μηδέν, έχοντας παράλληλα ως δεδομένο ότι το κόστος εφαρμογής κάποιων μεθόδων προστασίας μπορεί να υπερβαίνει τη ζημία που θα υποστεί ο οργανισμός ή το όφελος που θα αποκομίσει ο επιτιθέμενος σε κάποια πιθανή παραβίαση, συνεπώς σε αυτές τις περιπτώσεις η εφαρμογή των συγκεκριμένων μέτρων δεν είναι σκόπιμη. Ο διαχειριστής πρέπει να είναι υπεύθυνος, έμπειρος, διορατικός, να ασχολείται με τη φροντίδα του συστήματος σε τακτική βάση και να έτοιμος να αποτρέψει κάθε πιθανή απειλή. Όσο πιο συχνά πραγματοποιείται έλεγχος για «τρύπες» ασφαλείας σε ένα πληροφοριακό σύστημα και όσο καλύτερα αυτές κλείνουν, τόσο ελαχιστοποιείται ο κίνδυνος.

Πηγές - Βιβλιογραφία

1. Πολιτικές ασφάλειας Πληροφοριακών Συστημάτων, – HTTPS://www8.cs.ucy.ac.cy/courses/EPL674/lectures/Politikes-Asfaleias-Pliroforiakwn-Systimatwn.pdf
2. Πολιτικές ασφάλειας Πληροφοριακών Συστημάτων, – HTTP://www.icsd.aegean.gr/website_files/metaptyxiako/315624416.ppt
3. Ασφάλεια Πληροφοριακών Συστημάτων, - HTTP://el.wikipedia.org/wiki/Ασφάλεια_πληροφοριακών_συστημάτων
4. Krause M., Tipton H., (1993) *Handbook of Information Security Management,* CRC Press LLC, - HTTPS://www.cccure.org/Documents/HISM/ewtoc.html
5. Caravan J., (2001), *Fundamentals of Network Security,* Artech House, INC., - HTTP://www.securnet.biz/Ebooks/Network_Security.pdf
6. Swanson M., (2001), *Security Self-Assessment Guide for Information Technology Systems*, NIST SP 800-26, - HTTP://infohost.nmt.edu/~sfs/Regs/sp800-26.pdf
7. Santos O., (2008), *End-to-End Network Security. Defense-in-Depth,* Cisco Press
8. OWASP Testing Guide V.3, (2008), - HTTPS://www.google.gr/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&ved=0CDQQFjAB&url=https%3A%2F%2Fwww.owasp.org%2Fimages%2F5%2F56%2FOWASP_Testing_Guide_v3.pdf&ei=oNVWUrzFKImXtQbJvoDwDA&usg=AFQjCNFL2d3b66xLQA66GarlswyumhXXSQ&sig2=UN6Fh6hNnCG8efEUUg7pcg
9. OWASP: Top Ten 2010 Project, - HTTP://owasptop10.googlecode.com/files/OWASP%20Top%2010%20-%202010.pdf
10. Web Fuzzing, - HTTP://en.wikipedia.org/wiki/Fuzz_testing
11. Exploit, - HTTP://en.wikipedia.org/wiki/Exploit_(computer_security)
12. Papapanagiotou K., (2008), Detecting Web Application Vulnerabilities Using Open Source Means, - HTTP://repository.ellak.gr/ellak/bitstream/11087/1517/1/3-papapanagioutou-owasp-gr.pdf
13. Scarfone K., Souppaya M., Cody A., Orebaugh A., (2008), *Technical Guide to Information Security Testing and Assessment,* NIST SP 800-115, - HTTP://dl.acm.org/ft_gateway.cfm?id=2206199&ftid=1222157&dwn=1&CFID=369480346&CFTOKEN=41011152
14. Scarfone K., Mell M., (2007), *Guide to Intrusion Detection and Prevention Systems (IDPS),* NIST SP 800-94, - HTTP://dl.acm.org/ft_gateway.cfm?id=2206304&ftid=1222172&dwn=1&CFID=369480346&CFTOKEN=41011152
15. Protecting Your Core: Infrastructure Protection Access Control Lists, - HTTP://www.cisco.com/image/gif/paws/43920/iacl.pdf

16. Configuring SPAN, - HTTP://www.cisco.com/en/US/docs/switches/datacenter/sw/5_x/nx-os/system_management/configuration/guide/sm_14span.pdf
17. Teare D., (2008), *Designing for Cisco Internetwork Solutions (DESGN)*, Cisco Press, pp. 47, - HTTP://www.itsolutions.pro/images/stories/docs/cisco.press.designing.for.cisco.internetwork.solutions.desgn.pdf
18. The Web Application Security Consortium (WASC) Threat Classification V.2.0, (2010), - HTTP://projects.webappsec.org/f/WASC-TC-v2_0.pdf
19. Σαμψών Δ., (2003), *Η Γλώσσα Σήμανσης XML,* Σημειώσεις Παν. Πειραιώς, - HTTP://www.fme.aegean.gr/sites/default/files/dsampson_xml_lectures-notes-dec2003.pdf
20. Symbolic link, - HTTP://en.wikipedia.org/wiki/Symbolic_link
21. JavaServer Pages, - HTTP://en.wikipedia.org/wiki/JavaServer_Pages
22. JSON, - HTTP://en.wikipedia.org/wiki/Json
23. Stateless Protocol, - HTTP://en.wikipedia.org/wiki/Stateless_protocol
24. Tiller J., (2005), *The Ethical Hack. A Framework fot Business Value Penetration Testing*, Auerbach Publications, pp. 189
25. Kyriakou N., Kintis P., Penetration Testing: A guide to network Vulnerabilities and Fortification, - HTTP://panagiotious.files.wordpress.com/2009/11/ptt1.pdf
26. Vulnerability scanner, - HTTP://en.wikipedia.org/wiki/Vulnerability_scanner
27. SecTools.org: Top 125 network Security Tools, - HTTP://sectools.org/tool
28. Nessus, - HTTP://www.tenable.com/products/nessus
29. Core Impact, - HTTP://www.coresecurity.com/what-core-impact-tests
30. OpenVAS, - HTTP://en.wikipedia.org/wiki/OpenVAS
31. Metasploit, - HTTP://www.metasploit.com/
32. Social Engineer Toolkit, - HTTP://www.social-engineer.org/framework/Computer_Based_Social_Engineering_Tools:_Social_Engineer_Toolkit_(SET)
33. WebGoat, - HTTPS://www.owasp.org/index.php/Category:OWASP_WebGoat_Project
34. Backtrack, - HTTP://www.backtrack-linux.org/about/
35. Wireshark, - HTTP://www.wireshark.org/
36. Cain and Abel, - HTTP://www.oxid.it/cain.html
37. Voice over IP, - HTTP://en.wikipedia.org/wiki/Voip
38. TcpDump, - HTTP://www.tcpdump.org/
39. Ettercap, - HTTP://ettercap.github.io/ettercap/
40. Packet crafting, - HTTP://en.wikipedia.org/wiki/Packet_crafting

41. Netcat, - HTTP://en.wikipedia.org/wiki/Netcat
42. Hping, - HTTP://www.hping.org/
43. Yersinia, - HTTP://www.yersinia.net/
44. Understanding and Configuring Spanning Tree Protocol (STP) in Catalyst Switches, - HTTP://www.cisco.com/image/gif/paws/5234/5.pdf
45. Cisco Discovery Protocol, - HTTP://www.cisco.com/en/US/tech/tk648/tk362/tk100/tsd_technology_support_sub-protocol_home.html
46. Hot Stanby Router Protocol Features abd Fuctionality, - HTTP://www.cisco.com/image/gif/paws/9234/hsrpguidetoc.pdf
47. Password cracking, - HTTP://en.wikipedia.org/wiki/Password_cracking
48. Aircrack-ng, - HTTP://www.aircrack-ng.org/
49. THC Hydra, - HTTP://www.thc.org/thc-hydra/
50. Medusa, - HTTP://foofus.net/goons/jmk/medusa/medusa.html
51. RainbowCrack, - HTTP://project-rainbowcrack.com/
52. Web application security scanner, - HTTP://en.wikipedia.org/wiki/Web_application_security_scanner
53. Challenges faced by automated web application security assessment tools, - HTTP://www.cgisecurity.com/scannerchallenges.html
54. W3af, - HTTP://w3af.org/
55. Webscarab, - HTTPS://www.owasp.org/index.php/Category:OWASP_WebScarab_Project
56. Skipfish, - HTTPS://code.google.com/p/skipfish/
57. Netsparker, - HTTPS://www.mavitunasecurity.com/netsparker/
58. Firebug, - HTTP://getfirebug.com/
59. Rootkit Detectors, - Windows Rootkit Overview, Symantc, - HTTP://www.symantec.com/avcenter/reference/windows.rootkit.overview.pdf
60. Rootkit, - HTTP://en.wikipedia.org/wiki/Rootkit
61. Sysinternals, - HTTP://technet.microsoft.com/en-us/sysinternals/default.aspx
62. HijackThis, - HTTP://www.hijackthis.de/en
63. WebScarab Getting Started, - HTTPS://www.owasp.org/index.php/WebScarab_Getting_Started
64. Webscarab documentation, - HTTP://dawes.za.net/rogan/webscarab/docs/
65. BeanShell, - HTTP://en.wikipedia.org/wiki/BeanShell
66. Παπαπαναγιώτου Κ., *OWASP: Ασφάλεια στις Web εφαρμογές*, HTTP://vlm1.uta.edu/~akotsif/ELF22-OWASP.pdf