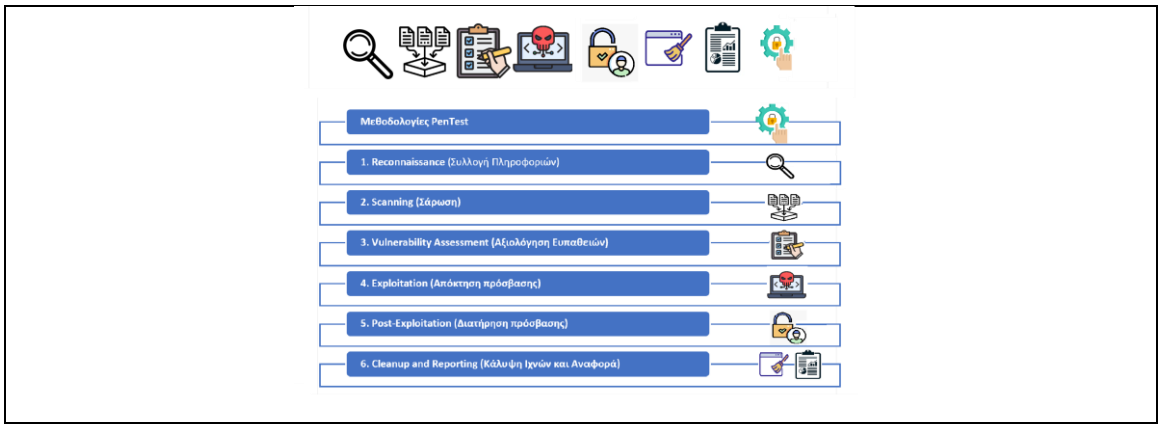
ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Επιθέσεις Παρείσδυσης (Penetration testing) σε Δικτυακές Βάσεις Δεδομένων»



Του φοιτητή
Κελεμενίδη Θεόδωρου
Αρ. Μητρώου: 2019068

Επιβλέπων
Δρ. Ηλιούδης Χρήστος
Καθηγητής

9 Δεκεμβρίου 2024

Τίτλος Δ.Ε. Επιθέσεις Παρείσδυσης (Penetration testing) σε Δικτυακές Βάσεις δεδομένων
Κωδικός Δ.Ε. 24200
Κελεμενίδης Θεόδωρος
Ηλιούδης Χρήστος
Ημερομηνία ανάληψης Δ.Ε. 10-04-2024
Ημερομηνία περάτωσης Δ.Ε. 09-12-2024

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραιτήτως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

«Σε όσους αγωνίζονται για την εκπλήρωση των ονείρων τους»

Πρόλογος

Η ασφάλεια των συστημάτων αποτελεί σημαντική προτεραιότητα στον κυβερνοχώρο, και για αυτό οι επιχειρήσεις αναζητούν ανθρώπους να πραγματοποιήσουν επιθέσεις παρείσδυσης για να χρησιμοποιήσουν τις δεξιότητές τους στο hacking ώστε να αποκαλύψουν κενά ασφαλείας που θα μπορούσαν να εκμεταλλευτούν εισβολείς και να προστατεύσουν έτσι ευαίσθητα ψηφιακά δεδομένα. Οι επιθέσεις παρείσδυσης αποκαλύπτουν αδυναμίες που οδηγούν σε πολιτικές αντιμετώπισης τους ώστε να ενισχυθεί η ασφάλεια των συστημάτων. Ταυτόχρονα οι απειλές στον κυβερνοχώρο εξελίσσονται συνεχώς παρακάμπτοντας τα βελτιωμένα συστήματα ασφαλείας. Έτσι η πραγματοποίηση κακόβουλων επιθέσεων με στόχο να εκβιαστούν οι εταιρείες θα παραμένει μια διαρκής ανησυχία. Η πραγματοποίηση επιθέσεων παρείσδυσης είναι χρήσιμες προκειμένου να εξεταστεί η αποτελεσματικότητα των μέτρων ασφαλείας ενός συστήματος, όπως και για τον μετριασμό του κινδύνου, την ευαισθητοποίηση σχετικά με την ασφάλεια γενικότερα αλλά και αποτελεί βοήθεια για το προσωπικό ώστε να μάθει πως να ενεργεί έναντι κακόβουλων ενεργειών και να διασφαλίσει τη συμμόρφωση για την αντιμετώπιση επιθέσεων. Για την ανάδειξη της χρησιμότητας των επιθέσεων παρείσδυσης έγινε καταγραφή και αξιολόγηση των προβλημάτων ασφάλειας των Συστημάτων Βάσεων Δεδομένων, η συγκριτική αξιολόγηση των καταγεγραμμένων ευπαθειών για τα πιο διαδεδομένα RDBMS και η περιγραφή των μεθοδολογιών και των εργαλείων επιθέσεων παρείσδυσης για RDBMS.

Περίληψη

Η ανάπτυξη του ηλεκτρονικού εμπορίου μέσω της ανάπτυξης του Παγκόσμιου Ιστού είναι μια πραγματικότητα και η ανάπτυξη αυτή συνεχίζεται αμείωτη. Πολλές και διαφορετικές τεχνολογίες και πλατφόρμες υποστηρίζουν διαδικτυακές εφαρμογές ηλεκτρονικού εμπορίου. Η τεράστια ανάπτυξη και η πολυπλοκότητα που παρουσιάζουν τέτοιες εφαρμογές με την ποικιλία των τεχνολογιών που τις υποστηρίζουν κάνουν ιδιαίτερα κρίσιμη την ασφάλεια τους. Ευαίσθητα προσωπικά δεδομένα επιβάλλεται να προστατευτούν, και οι πολιτικές αντιμετώπισης επιθέσεων προβάλλονται αναδεικνύοντας τη σημασία της ασφάλειας στο σημερινό ψηφιακό κόσμο. Η παρούσα εργασία επιχειρεί μια βιβλιογραφική ανασκόπηση σχετικά με τις επιθέσεις παρείσδυσης. Οι επιθέσεις παρείσδυσης θεωρούνται μια καλά οργανωμένη μέθοδος αξιολόγησης των υπολογιστικών συστημάτων ενός οργανισμού, και καθώς ένα υπολογιστικό σύστημα είναι ένας συνδυασμός λογισμικού και υλικού, αλλά και των ανθρώπων που τα χρησιμοποιούν οι αδυναμίες και τα κενά ασφαλείας αναζητούνται παντού. Προκειμένου να εντοπιστούν τρωτά σημεία ασφαλείας, απειλές και κίνδυνοι που μπορεί ένας εισβολέας να εκμεταλλευτεί σε εφαρμογές λογισμικού, δίκτυα ή εφαρμογές του Ιστού, αυτή η τεχνική περιλαμβάνει την ανάλυση του υπολογιστικού συστήματος της εταιρείας για την αναζήτηση ελαττωμάτων. Η αναζήτηση είναι πολύ-επίπεδη και κατά τη διάρκεια των επιθέσεων παρείσδυσης γίνεται ενεργή εξέταση του συστήματος για τυχόν ευπάθειες. Εξετάζονται σφάλματα σχεδίασης, λανθασμένη ή κατώτερη του επιθυμητού διαμόρφωση του συστήματος, ελαττώματα στο υλικό και το λογισμικό αλλά και αδυναμίες στις διαδικασίες λειτουργίας. Υπάρχουν διάφοροι τύποι επιθέσεων/δοκιμών παρείσδυσης που περιγράφονται στην εργασία. Επιπρόσθετα, έγινε καταγραφή και αξιολόγηση των προβλημάτων ασφαλείας των Συστημάτων Βάσεων Δεδομένων και συγκριτική αξιολόγηση των καταγεγραμμένων ευπαθειών των πιο διαδεδομένων RDBMS. Στη συνέχεια, αναλύθηκε η σημασία των δοκιμών παρείσδυσης για την διασφάλιση της ασφάλειας και κατά συνέπεια και της ποιότητας των συστημάτων. Στο τέλος, δίνονται τα συμπεράσματα της έρευνας και τα πλεονεκτήματα που οι επιθέσεις παρείσδυσης προσδίδουν.

«Penetration testing in Network Databases»

«Kelemenidis Theodoros»

Abstract

The growth of e-commerce through the development of the World Wide Web is a reality and this growth continues unabated. Many different technologies and platforms support online e-commerce applications. The enormous growth and complexity of such applications with the variety of technologies that support them make their security particularly critical. Sensitive personal data must be protected, and policies to deal with attacks are highlighted, highlighting the importance of security in today's digital world. This paper attempts a literature review on penetration tests. Penetration tests are considered a well-organized method of evaluating an organization's computer systems, and as a computer system is a combination of software and hardware, as well as the people who use them, weaknesses and security gaps are being searched for everywhere. In order to identify security vulnerabilities, threats and risks that an attacker can exploit in software applications, networks or web applications, this technique involves analyzing the company's computer system to search for defects. The search is multi-layered and during penetration tests, the system is actively examined for any vulnerabilities. Design errors, incorrect or suboptimal system configuration, hardware and software defects, as well as weaknesses in operating procedures are examined. There are various types of attacks/penetration tests described in the work. In addition, the security problems of Database Systems were recorded and evaluated and the recorded vulnerabilities of the most widespread RDBMS were compared. Then, the importance of penetration tests for ensuring security and consequently the quality of systems was analyzed. Finally, the conclusions of the research and the advantages that intrusion attacks provide are given.

Ευχαριστίες

Θέλω να ευχαριστήσω την οικογένεια μου που με στήριξε σε αυτό το κυριότερο ταξίδι της ζωής μου. Θα ήθελα να ευχαριστήσω από καρδίας για την καθοδήγηση και την υποστήριξη του, τον καθηγητή Κύριο Χρήστο Ηλιούδη που με την κατάλληλη κατεύθυνση του και την άρτια γνώση του στο αντικείμενο μπόρεσα να ανταπεξέλθω στης προκλήσεις της διπλωματικής μου εργασίας. Επιπρόσθετα θα ήθελα να ευχαριστήσω όσους και όσες με στήριξαν που πίστεψαν σε εμένα και με ενθάρρυναν να συνεχίσω τον αγώνα μου για την εκπλήρωση των στόχων μου ακόμα και όταν μου φαίνονταν αδύνατο να τα πραγματοποιήσω. Τέλος δεν θα μπορούσα να μην ευχαριστήσω την γιαγιά μου αλλά και τον παππού μου που με της συμβουλές τους και την υποστήριξη τους με εμψύχωσαν και με οδήγησαν στον ορθό δρόμο για την πραγμάτωση των ονείρων μου.

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία
DBMS	Database Management Systems
DCL	Data Control Language
DDL	Data Definition Language
DDoS	Distributed Denial-of-Service
DML	Data Manipulation Language
DNS	Domain Name System
DoS	Denial of Service
DQL	Data Query Language
MFA	Multifactor Authentication
MITM	Man-in-the-Middle
PenTest	Penetration Testing
RDBMS	Relational Database Management Systems
SDLC	Software Development Life Cycle
SET	Social Engineering Toolkit
SQLi	SQL Injection

Κεφάλαιο 1ο: Εισαγωγή

1.1 Περιοχή έρευνας και αντικείμενο της διπλωματικής εργασίας

Οι συνεχείς εξελίξεις στις τεχνολογίες πληροφοριακών συστημάτων συντέλεσαν ώστε πολλές εφαρμογές σε διάφορους εμπορικούς τομείς να έχουν μηχανογραφηθεί. Κι αυτό με τη σειρά του συνέβαλε στην αξία που έχουν για πολλές επιχειρήσεις τα δεδομένα, τα οποία έχουν γίνει ένας απαραίτητος πόρος. Είναι απαραίτητο να υπάρχει η δυνατότητα πρόσβασης για χρήση και επεξεργασία των δεδομένων, επιτρέποντας την εξαγωγή πληροφοριών που υπάρχουν στις βάσεις δεδομένων με τη μορφή προτύπων και τάσεων, επιπλέον της ενσωμάτωσης των πολυάριθμων πηγών δεδομένων που βρίσκονται διάσπαρτες σε πολλούς ιστοτόπους. Αυτές οι πηγές δεδομένων προέρχονται από βάσεις δεδομένων που επιβλέπουν τα Συστήματα Διαχείρισης Βάσεων Δεδομένων (Database Management Systems - DBMS) ή προέρχονται από μια ποικιλία πηγών δεδομένων που συνδυάζονται και αποθηκεύονται σε ένα χώρο αποθήκευσης.

Η εισαγωγή του Παγκόσμιου Ιστού (WWW) στα μέσα της δεκαετίας του 1990 αύξησε τη ζήτηση για αποτελεσματική διαχείριση δεδομένων, πληροφοριών και γνώσης. Σήμερα, υπάρχουν τόσα πολλά δεδομένα στο Διαδίκτυο που καθιστά δύσκολη τη διαχείριση τους με παραδοσιακές τεχνικές. Ως αποτέλεσμα, δημιουργείται ένας αριθμός τεχνολογιών που επιτρέπουν την εξαγωγή πληροφοριών από βάσεις δεδομένων και αποθήκες στο διαδίκτυο, καθώς και για την παροχή διαλειτουργικότητας και αποθήκευσης μεταξύ πολλών πηγών δεδομένων και συστημάτων.

Η διατήρηση της ασφάλειας τόσο των βάσεων δεδομένων, όσο και των εφαρμογών αλλά και των διαφόρων συστημάτων πληροφοριών είναι ζωτικής σημασίας καθώς αυξάνεται η ζήτηση για διαχείριση δεδομένων και πληροφοριών. Οι πληροφορίες και τα δεδομένα πρέπει να προστατεύονται από κακόβουλη χειραγώγηση και ανεπιθύμητη πρόσβαση. Από τότε που υπάρχει το Διαδίκτυο, η προστασία των δεδομένων και των πληροφοριών έχει γίνει ακόμη πιο σημαντική, λόγω της καθολικής πρόσβασης σε αυτό σε παγκόσμιο επίπεδο. Για το λόγο αυτό, απαιτούνται ισχυρά μέτρα ασφαλείας των δεδομένων αλλά και των εφαρμογών.

Επιπρόσθετα ο σημασιολογικός ιστός αναπτύσσεται αυτήν τη στιγμή ως η εξέλιξη του Παγκόσμιου Ιστού αποτελώντας ουσιαστικά έναν ιστό διασυνδεδεμένων δεδομένων. Η διασφάλιση ότι οι μηχανές μπορούν να διαβάζουν και να κατανοούν ιστοσελίδες είναι ο στόχος του σημασιολογικού ιστού. Οι υποδομές Ιστού, οι βάσεις δεδομένων Ιστού, οι υπηρεσίες Ιστού, η διαχείριση οντολογίας και η ενοποίηση πληροφοριών είναι τα κύρια στοιχεία του σημασιολογικού ιστού. Καθένα από αυτά τα τρία θέματα έχει αποτελέσει αντικείμενο εκτενούς εργασίας. Από την άλλη πλευρά, η ασφάλεια έχει λάβει πολύ λίγη προσοχή. Είναι σημαντικό να διασφαλιστεί πως τα δεδομένα στον ιστό προστατεύονται από κακόβουλες αλλαγές και παράνομη πρόσβαση, εάν θέλουμε να έχουμε έναν επιτυχημένο σημασιολογικό ιστό. Επιπλέον, πρέπει να διασφαλίσουμε ότι τηρείται το απόρρητο των ανθρώπων. Για το λόγο αυτό πρέπει αν εστιάζουμε σε θέματα ασφάλειας και απορρήτου με βάσεις δεδομένων καθώς οι βάσεις δεδομένων και οι υπηρεσίες Ιστού αποτελούν συστατικό και του σημασιολογικού ιστού [1].

1.2 Συστήματα Βάσεων Δεδομένων και η ασφάλεια τους

Καθώς τα συστήματα βάσεων δεδομένων χρησιμοποιούνται ευρύτερα από τους οργανισμούς ως το κύριο εργαλείο διαχείρισης δεδομένων για τις καθημερινές λειτουργίες και τη λήψη αποφάσεων, η ασφάλεια των δεδομένων που διαχειρίζονται αυτά τα συστήματα γίνεται όλο και πιο σημαντική. Η απώλεια δεδομένων και η κακή χρήση μπορεί να έχουν καταστροφικές επιπτώσεις σε ολόκληρο τον

οργανισμό, εκτός από το να επηρεάσουν ένα μεμονωμένο άτομο ή εφαρμογή. Η προστασία δεδομένων είναι τώρα πιο σημαντική από ποτέ, λόγω της πρόσφατης και ταχείας επέκτασης των εφαρμογών και των συστημάτων πληροφοριών που βασίζονται στο Web, τα οποία έχουν αυξήσει την έκθεση στον κίνδυνο των βάσεων δεδομένων. Η αναγνώριση ότι οι εσωτερικοί κίνδυνοι πρέπει να αντιμετωπίζονται εκτός από τους εξωτερικούς είναι ζωτικής σημασίας για την προστασία των δεδομένων.

Η μη διαθεσιμότητα δεδομένων όταν αυτά χρειάζονται, η αλλοίωση δεδομένων και η μη εξουσιοδοτημένη πρόσβαση σε δεδομένα είναι οι τρεις κύριες κατηγορίες που περιγράφουν τις παραβιάσεις ασφαλείας. Πληροφορίες στις οποίες δεν επιτρέπεται νόμιμα η πρόσβαση αποκαλύπτονται στους χρήστες μέσω μη εξουσιοδοτημένης πρόσβασης σε δεδομένα και μπορεί να οδηγήσει σε σημαντικές απώλειες από οικονομική και ανθρώπινη άποψη για όλους τους τύπους οργανισμών, από κοινωφελείς έως εμπορικούς, που δραστηριοποιούνται σε διάφορους τομείς όπως στο χώρο της υγείας. Τα προβλήματα στις βάσεις δεδομένων είναι συνέπεια σκόπιμων ή ακούσιων αλλαγών δεδομένων. Οποιαδήποτε χρήση δεδομένων σε χρήστη που δεν έχει εξουσιοδότηση θα μπορούσε να προκαλέσει την μη ομαλή λειτουργία του οργανισμού και ο οργανισμός να υποστεί σημαντικές ζημίες.

Για να αντιμετωπιστεί πλήρως η ασφάλεια των δεδομένων, πρέπει να πληρούνται οι ακόλουθες τρεις προϋποθέσεις, η τριάδα CIA – εμπιστευτικότητα (confidentiality), ακεραιότητα (integrity) και διαθεσιμότητα (availability). Αναλυτικότερα [2]:

1. Η εχεμύθεια ή εμπιστευτικότητα που αναφέρεται στη διασφάλιση των δεδομένων από ακατάλληλη ή μη εξουσιοδοτημένη αποκάλυψη,
2. Η ακεραιότητα που αναφέρεται στην αποφυγή ακατάλληλης ή μη εξουσιοδοτημένης τροποποίησης δεδομένων και
3. Η διαθεσιμότητα που αναφέρεται στην ανάκτηση και την πρόληψη αρνήσεων πρόσβασης κακόβουλων δεδομένων και σφαλμάτων υλικού και λογισμικού που καθιστούν το σύστημα βάσης δεδομένων ανενεργό.

Κάθε εφαρμογή πρέπει να τρέχει σε ένα περιβάλλον που πληροί και τις τρεις προϋποθέσεις.

Αν για παράδειγμα εξεταστεί μια βάση δεδομένων στην οποία διατηρούνται τα δεδομένα μισθοδοσίας. Οι διαταγές πληρωμής πρέπει να εκτυπώνονται εγκαίρως στο τέλος της περιόδου αποδοχών, ενώ οι μισθοί των υπαλλήλων θα πρέπει να αποκαλύπτονται μόνο στους ίδιους κι όχι σε χρήστες μη εξουσιοδοτημένους. Επιπρόσθετα, μόνο οι εξουσιοδοτημένοι χρήστες μπορούν να αλλάξουν τους μισθούς.

Ανάλογα αν μελετηθεί η περίπτωση ενός ιστοτόπου μιας αεροπορικής εταιρείας. Πρέπει οι κρατήσεις πελατών σε αυτόν τον ιστότοπο, να παραμένουν προσβάσιμες στους συγκεκριμένους και μόνο πελάτες με τους οποίους σχετίζονται. Επίσης αυτές οι κρατήσεις να μην μπορούν να αλλάξουν κατά βούληση και να είναι συνεχώς διαθέσιμες οι πληροφορίες πτήσεων και κρατήσεων.

Εκτός από αυτές τις προϋποθέσεις, τα ζητήματα απορρήτου είναι αρκετά σημαντικά στον σημερινό κόσμο. Αν και το απόρρητο και η εμπιστευτικότητα μερικές φορές χρησιμοποιούνται ως συνώνυμες, δεν είναι το ίδιο πράγμα. Οι τεχνικές εμπιστευτικότητας των πληροφοριών μπορούν να εφαρμοστούν για την εφαρμογή του απορρήτου. Ωστόσο, η εγγύηση του απορρήτου απαιτεί άλλες προσεγγίσεις, συμπεριλαμβανομένων των διαδικασιών συναίνεσης των χρηστών. Επιπλέον, η εμπιστευτικότητα μπορεί να επιτευχθεί αποτρέποντας την πρόσβαση σε συγκεκριμένα δεδομένα, ενώ το απόρρητο είναι απαραίτητο ακόμη και μετά την αποκάλυψη των δεδομένων. Αυτό σημαίνει πως οι πληροφορίες πρέπει να χρησιμοποιούνται μόνο για το σκοπό που εξουσιοδότησε ο χρήστης και όχι για άλλους σκοπούς.

Πολλά στοιχεία ενός συστήματος διαχείρισης βάσης δεδομένων (DBMS) εγγυώνται την προστασία των δεδομένων. Ειδικότερα, οι μηχανισμοί ελέγχου πρόσβασης, εγγυώνται την εμπιστευτικότητα των δεδομένων. Κάθε φορά που ένα άτομο επιχειρεί να αποκτήσει πρόσβαση σε ένα αντικείμενο δεδομένων, ο μηχανισμός ελέγχου πρόσβασης επαληθεύει τα δικαιώματά του έναντι μιας λίστας εξουσιοδοτήσεων, οι οποίες συνήθως παρέχονται από έναν διαχειριστή ασφαλείας.

Μια εξουσιοδότηση καθορίζει εάν επιτρέπεται ή όχι σε ένα άτομο να χρησιμοποιεί ένα αντικείμενο με συγκεκριμένο τρόπο. Οι πολιτικές ελέγχου πρόσβασης του οργανισμού ακολουθούνται κατά τη δήλωση εξουσιοδοτήσεων. Όταν τα δεδομένα μεταφέρονται μέσω δικτύων ή διατηρούνται σε μέσο για δευτερεύουσα αποθήκευση, χρησιμοποιούνται τεχνικές κρυπτογράφησης για περαιτέρω βελτίωση της εμπιστευτικότητας των δεδομένων.

Στο πλαίσιο της εξωτερικής ανάθεσης διαχείρισης δεδομένων, η χρήση τεχνικών κρυπτογράφησης έχει προσελκύσει πρόσφατα μεγάλη προσοχή. Σε αυτές τις περιπτώσεις, το κύριο μέλημα είναι πως να υλοποιηθούν λειτουργίες όπως οι αναζητήσεις για παράδειγμα, σε κρυπτογραφημένα δεδομένα.

Οι περιορισμοί που προκύπτουν προκειμένου να διατηρηθεί η σημασιολογική ακεραιότητα των δεδομένων καθώς και η ύπαρξη μηχανισμών ελέγχου πρόσβασης σε αυτά συμβάλουν στην ακεραιότητα των δεδομένων. Το υποσύστημα σημασιολογικής ακεραιότητας επιβεβαιώνει ότι τα τροποποιημένα δεδομένα είναι σημασιολογικά έγκυρα και ο μηχανισμός ελέγχου πρόσβασης επιβεβαιώνει ότι ο χρήστης έχει την εξουσία να επεξεργάζεται τα δεδομένα κάθε φορά που ένα υποκείμενο προσπαθεί να το κάνει. Ένα σύνολο συνθηκών που πρέπει να ελεγχθούν σε σχέση με την κατάσταση της βάσης δεδομένων προκειμένου να διασφαλιστεί η σημασιολογική ακρίβεια. Η ψηφιακή υπογραφή δεδομένων μπορεί να χρησιμοποιηθεί για τον εντοπισμό χειραγώγησης. Τελικά, η διαθεσιμότητα και η ακρίβεια των δεδομένων είναι εγγυημένες από τον μηχανισμό ελέγχου ταυτόχρονης χρήσης και το υποσύστημα ανάκτησης, ακόμη και σε περίπτωση δυσλειτουργιών λογισμικού και υλικού, καθώς και πρόσβασης από εφαρμογές ταυτόχρονων εφαρμογών. Η διαθεσιμότητα δεδομένων μπορεί να αυξηθεί περαιτέρω, ιδιαίτερα για δεδομένα για τα οποία υπάρχει πρόσβαση μέσω του Διαδικτύου, με τη χρήση μεθόδων για την αποτροπή επιθέσεων άρνησης υπηρεσίας (DoS), όπως για παράδειγμα μεθόδους στις οποίες γίνεται χρήση της μηχανικής μάθησης [3].

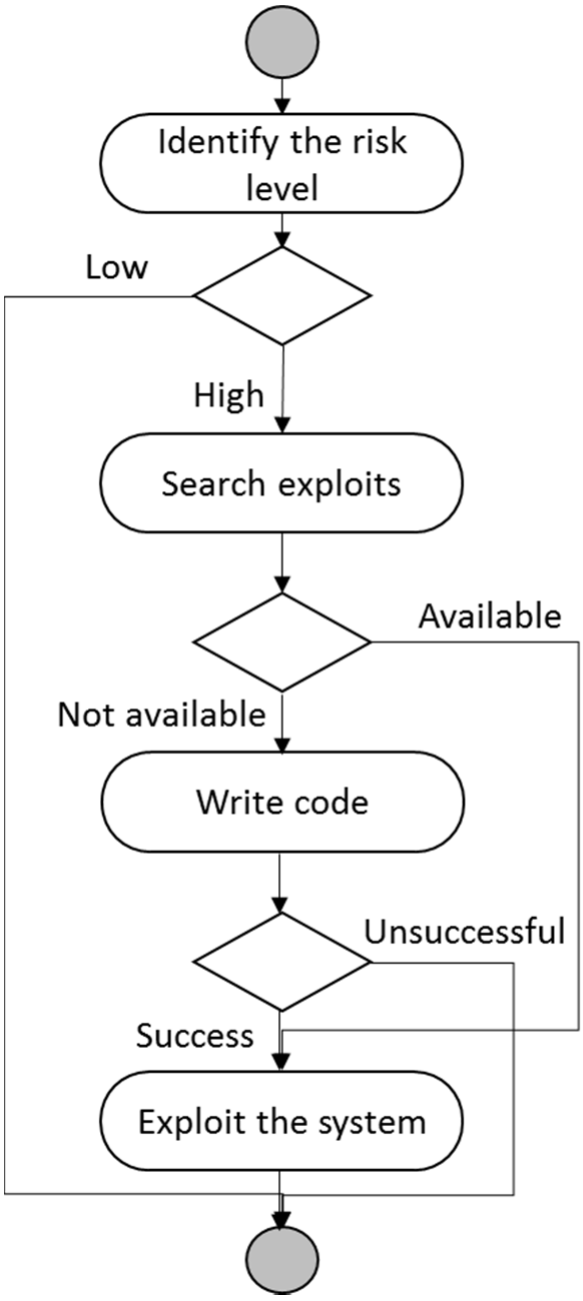
Ο επικεφαλής Κυβερνοασφάλειας και Επιχειρήσεων (COO) του Οργανισμού της ΕΕ για την Κυβερνοασφάλεια (ENISA), Χανς Ντε Βρίες, ανέφερε την ύπαρξη διαφόρων τύπων κυβερνοεπιθέσεων, όπως επιθέσεις DDoS και το κακόβουλο λογισμικό ransomware, λέγοντας πως ενώ τώρα αυτές προκαλούν μικρά προβλήματα, για παράδειγμα μπορεί μια επίθεση να βγάλει προσωρινά έναν ιστότοπο εκτός λειτουργίας, ενδεχομένως μία τέτοιου είδους κυβερνοεπίθεση να μπορέσει να επιφέρει δυσάρεστες επιπτώσεις στους επιτιθέμενους οργανισμούς. Και ενώ όπως σημειώνει η τεχνικές μηχανικής μάθησης αποτελούν εργαλεία που μπορούν να χρησιμοποιηθούν για την αποτροπή κυβερνοεπιθέσεων, βρίσκουν εφαρμογή από χάκερς που επιδιώκουν να εξαπατήσουν για να αποσπάσουν μεγάλα χρηματικά ποσά. Ήδη έχουν αναφερθεί οι πρώτες επιθέσεις με χρήση Τεχνητής Νοημοσύνης, με Deepfake συγκεκριμένα που αποτελεί τη σύγχρονη παραπληροφόρηση (fake news). Ενδεχόμενος το πιο επικίνδυνο είδος παραπληροφόρησης, όπου αντιγράφονται πρόσωπα και φωνές δημιουργώντας πολυμεσικά αρχεία όπου φαίνεται ή και ακούγεται κάποιος-α να κάνει ή να λέει κάτι αναληθές [4] [5].

Η ασφάλεια των Συστημάτων Βάσεων Δεδομένων είναι ζωτικής σημασίας όσον αφορά στην προστασία των δεδομένων από μη εξουσιοδοτημένη πρόσβαση και κακόβουλες ενέργειες. Οι επιθέσεις παρείσδυσης, γνωστές ως Penetration Testing (PenTest), ανήκουν στις μεθόδους που χρησιμοποιούνται για τον εντοπισμό και την αξιολόγηση των ευπαθειών στα συστήματα αυτά.

1.3 Στόχοι της διπλωματικής εργασίας

Οι στόχοι της διπλωματικής εργασίας είναι:

- η καταγραφή και αξιολόγηση των προβλημάτων ασφάλειας των Συστημάτων Βάσεων Δεδομένων
- η συγκριτική αξιολόγηση των καταγεγραμμένων ευπαθειών για τα πιο διαδεδομένα RDBMS (Σχεσιακές/Relational DBMS)
- η περιγραφή των μεθοδολογιών και των εργαλείων penTest για RDBMS
- η παρουσίαση μιας μελέτης περίπτωσης εφαρμογής.



Σχήμα 1.1: Λογικό μοντέλο διαδικασίας επίθεσης παρείσδυσης [6]

1.4 Επιτεύγματα της διπλωματικής εργασίας

Η εργασία επιχειρεί να παρουσιάσει τη σημασία της ασφάλειας στις βάσεις δεδομένων και τις πρακτικές που μπορούν να εφαρμοστούν για τη καλύτερη δυνατή ασφάλεια μέσα από:

1. Την καταγραφή και ανάλυση των προβλημάτων ασφάλειας των Συστημάτων Βάσεων Δεδομένων.

- Ανάλυση Επιθέσεων SQL Injection: Εμβάθυνση στη φύση των επιθέσεων SQL Injection και πώς οι επιτιθέμενοι τις εκμεταλλεύονται, με έμφαση στη χρήση εργαλείων όπως το SQLMap για αυτοματοποιημένη ανίχνευση και εκμετάλλευση.
- Ανάλυση Τεχνικών Υποκλοπής: Αναγνώριση και ανάλυση των μεθόδων που χρησιμοποιούνται για την υποκλοπή μη κρυπτογραφημένων διαπιστευτηρίων, όπως οι επιθέσεις μέσω network sniffing, Man-in-the-Middle (MITM) επιθέσεις, και session hijacking.
- Ανάλυση Επιθέσεων Buffer Overflows: Διερεύνηση του τρόπου που οι επιτιθέμενοι εκμεταλλεύονται ευπάθειες στη διαχείριση της μνήμης, επιτρέποντας την εκτέλεση κακόβουλου κώδικα ή την καταστροφή δεδομένων.

2. Την απαρίθμηση και τη συγκριτική αξιολόγηση των καταγεγραμμένων ευπαθειών στα πιο διαδεδομένα RDBMS.

- Επισκόπηση Κινδύνων: Παρουσίαση των κινδύνων που υπάρχουν από γνωστές ευπάθειες ανάλογα με το RDBMS.
- Συγκριτική Μελέτη: Σύγκριση των πιο κοινών ευπαθειών σε διαφορετικά RDBMS (π.χ., MySQL, Oracle, SQL Server), εστιάζοντας σε SQL Injection, buffer overflows, και άλλα θέματα ασφάλειας.

3. Την περιγραφή μεθοδολογιών και εργαλείων Penetration Testing για RDBMS.

- Μεθοδολογίες Penetration Testing: Περιγραφή των βέλτιστων πρακτικών για τη διεξαγωγή Penetration Testing σε βάσεις δεδομένων, συμπεριλαμβανομένων των τεχνικών για την ανακάλυψη και εκμετάλλευση ευπαθειών.
- Εργαλεία PenTest: Παρουσίαση των εργαλείων (για SQL Injection και άλλα είδη επιθέσεων) που χρησιμοποιούνται για τον έλεγχο των τρωτών σημείων των RDBMS.

4. Την παρουσίαση μιας Μελέτης Περίπτωσης Εφαρμογής.

- Εφαρμογή Τεχνικών PenTest σε Πραγματικό Σενάριο: Παρουσίαση μιας μελέτης περίπτωσης όπου εφαρμόζονται οι παραπάνω μέθοδοι και εργαλεία σε ένα συγκεκριμένο RDBMS για την ανίχνευση και εκμετάλλευση ευπαθειών.
- Αποτελέσματα και Προτάσεις Βελτίωσης: Επισήμανση των αποτελεσμάτων του Penetration Test και προτάσεις για βελτίωση της ασφάλειας της βάσης δεδομένων.

5. Προτάσεις για Μέτρα Αντιμετώπισης και Βελτίωσης της Ασφάλειας.

- Ανάπτυξη Πολιτικών Ασφάλειας: Προτάσεις για την εφαρμογή ισχυρών πολιτικών ασφαλείας, όπως η χρήση κρυπτογράφησης, ισχυρών κωδικών πρόσβασης, και πολυπαραγοντικής αυθεντικοποίησης.
- Εκπαίδευση Χρηστών: Προτάσεις για την εκπαίδευση των χρηστών στην αναγνώριση επιθέσεων όπως phishing και MITM, καθώς και στην υιοθέτηση ασφαλών πρακτικών.

1.5 Η δομή της εργασίας

Στην παρούσα εργασία γίνεται βιβλιογραφική ανασκόπηση σχετικά με τις επιθέσεις παρείσδυσης. Γίνεται ανάλυση της αξίας χρήσης δοκιμών παρείσδυσης για την διασφάλιση της ασφάλειας και κατά συνέπεια και της ποιότητας ενός συστήματος. Επιπρόσθετα, δίνονται τα συμπεράσματα της έρευνας και τα πλεονεκτήματα που οι επιθέσεις παρείσδυσης προσδίδουν. Ανά κεφάλαιο η δομή της εργασίας είναι:

Κεφάλαιο 1: Γίνεται μια εισαγωγική αναφορά στη σημασία των δεδομένων και των Βάσεων Δεδομένων στο σύγχρονο κόσμο. Ακολουθεί αναφορά στην ασφάλεια των συστημάτων Βάσεων Δεδομένων, και στη συμβολή των δοκιμών παρείσδυσης για τον εντοπισμό και την αξιολόγηση των ευπαθειών που εμφανίζουν. Επίσης παρουσιάζεται η δομή της διπλωματικής εργασίας.

Κεφάλαιο 2: Γίνεται καταγραφή και αξιολόγηση των προβλημάτων ασφαλείας των Συστημάτων Βάσεων Δεδομένων. Αρχικά, γίνεται μια γενική εισαγωγή στα Προβλήματα Ασφάλειας των RDBMS και στους τρόπους προστασίας για την αποτροπή προβλημάτων Ασφάλειας των RDBMS. Στη συνέχεια αναλύονται προβλήματα ασφάλειας, όπως επιθέσεις SQL Injection, Privilege Escalation, Data Breaches, Denial of Service (DoS), Weak Authentication Mechanisms, Buffer Overflows, κρυπτογράφηση και προστασία δεδομένων καθώς και Auditing και Monitoring.

Κεφάλαιο 3: Μια επισκόπηση, των δημοφιλέστερων RDBMS. Στη συνέχεια του κεφαλαίου αναφέρονται μηχανισμοί προστασίας έναντι των ευπαθειών. Τέλος ακολουθεί συγκριτική αξιολόγηση των καταγεγραμμένων ευπαθειών ασφαλείας για τα πιο διαδεδομένα RDBMS γίνεται στο κεφάλαιο αυτό.

Κεφάλαιο 4: Αναλύονται οι επιθέσεις παρείσδυσης – Penetration Tests (PenTests). Πως λειτουργούν, ποια είναι η διαδικασία εκτέλεσης μιας επίθεσης παρείσδυσης δικτύου και ποια τα οφέλη που αποκομίζουν οι εταιρείες από τις δοκιμές αυτές. Επίσης γίνεται αναφορά στη διαδικασία ελέγχου του κώδικα (Code Audit) του οποίου μια από τις συνηθέστερες τεχνικές είναι οι επιθέσεις παρείσδυσης εφαρμογών (penetration testing).

Κεφάλαιο 5: Περιγράφονται οι Μεθοδολογίες και τα Εργαλεία PenTest για RDBMS. Αναλύονται τα στάδια μιας επίθεσης Παρείσδυσης και αναφέρονται διάφορα παραδείγματα εργαλείων όπως Αυτοματοποιημένα Εργαλεία Εύρεσης Ευπαθειών (Automated Vulnerability Scanners), Εργαλεία Χειροκίνητων Δοκιμών (Manual Testing Tools), Εργαλεία που είναι Πλατφόρμες Εκμετάλλευσης (Exploitation Frameworks) και Εργαλεία Παρακολούθησης και Καταγραφής (Monitoring και Logging Tools).

Κεφάλαιο 6: Γίνεται η παρουσίαση μιας μελέτης Περίπτωσης Εφαρμογής.

Κεφάλαιο 7: Η επισκόπηση της έρευνας των Επιθέσεων Παρείσδυσης (Penetration testing) σε Δικτυακές Βάσεις Δεδομένων. Ακολούθως, δίνονται τα συμπεράσματα της έρευνας και τα πλεονεκτήματα που οι δοκιμές λογισμικού προσδίδουν.

Στο τέλος της εργασίας υπάρχει η βιβλιογραφία της εργασίας.

Κεφάλαιο 2ο: Καταγραφή και Αξιολόγηση των Προβλημάτων Ασφάλειας των Συστημάτων Βάσεων Δεδομένων

2.1 Εισαγωγή στα Προβλήματα Ασφάλειας των RDBMS

Για την επίτευξη της ασφάλειας των βάσεων δεδομένων στο σύγχρονο ψηφιακό περιβάλλον είναι σημαντική η συμβολή της πρόληψης. Θα πρέπει να λαμβάνονται υπόψη τα εξής κύρια σημεία [7]:

- **Η αύξηση απειλών**: Λόγω της αύξησης των συσκευών και των συστημάτων που παράγουν δεδομένα και οδηγεί σε μεγαλύτερο κίνδυνο παραβιάσεων ασφαλείας αλλά και οι επιπτώσεις του Covid-19, μια που η πανδημία είχε επιτείνει την ψηφιακή αλλαγή αυξάνοντας τις απειλές για την κυβερνοασφάλεια.
- **Οι αδυναμίες προστασίας**: Οι βάσεις δεδομένων, ειδικά οι NoSQL, αλλά και οι Σχεσιακές, είναι ευάλωτες σε επιθέσεις λόγω ανεπαρκούς προστασίας.
- **Τα εργαλεία αναζήτησης IoT**: Τα εργαλεία αναζήτησης IoT μπορούν να αποκαλύψουν ευπάθειες και να εκθέσουν δεδομένα.
- **Η προτεραιότητα στην Ασφάλεια**: Είναι αναγκαία η προτεραιοποίηση της ασφάλειας των δεδομένων, με την εφαρμογή βασικών μέτρων ασφαλείας και συνεχή επιθεώρηση για προστασία από διαρροές και άλλες απειλές για τη διατήρηση της αξίας τους.

Η μελέτη [7], αναφέρει ότι τα συστήματα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS) είναι επίσης ευάλωτα σε διάφορους τύπους ευπαθειών, παρά την ευρεία συζήτηση για το υψηλότερο κίνδυνο ευπαθειών για τα NoSQL συστήματα σε σύγκριση με τα SQL. Η πιο κοινή ευπάθεια που αναφέρεται συχνά είναι η εκτέλεση κώδικα, η οποία βρίσκεται στις τρεις πρώτες θέσεις στις περισσότερες βάσεις δεδομένων, ακολουθούμενη από την υπερχείλιση (overflow) και την άρνηση υπηρεσίας (Denial of Service - DoS). Η ασφάλεια των δεδομένων και η κατάλληλη διαμόρφωση της βάσης δεδομένων δεν αφορά μόνο τα NoSQL, τα οποία συνήθως θεωρούνται πολύ λιγότερο ασφαλή, αλλά και τα RDBMS. Η μελέτη τονίζει ότι η ασφάλεια των δεδομένων πρέπει να είναι κορυφαία προτεραιότητα κάθε στρατηγικής πληροφορικής ασφάλειας. Η αποτυχία συμμόρφωσης με τις απαιτήσεις για ασφάλεια και προστασία των δεδομένων μπορεί να οδηγήσει σε σημαντικές ζημιές και να βλάψει τη φήμη της εταιρείας.

Πλειάδα εφαρμογών χρησιμοποιεί σχεσιακές βάσεις δεδομένων για την αποθήκευση και την ανάκτηση δεδομένων. Καθώς οι βάσεις δεδομένων αποθηκεύουν κρίσιμες πληροφορίες αποτελούν στόχο για διάφορες κακόβουλες ενέργειες. Αλλά όπως και κάθε άλλη τεχνολογία, έχουν αδυναμίες που οι χάκερ μπορεί να εκμεταλλευτούν. Μερικά τυπικά τρωτά σημεία της σχεσιακής βάσης δεδομένων καθώς και τα προβλήματα ασφάλειας που μπορεί να προκύψουν και χρειάζεται να αντιμετωπιστούν περιλαμβάνουν τα παρακάτω [8] [9]:

- **SQL Injection**: Η πιο κοινή επίθεση, όπου οι επιτιθέμενοι εισάγουν κακόβουλο κώδικα SQL στις φόρμες εισαγωγής δεδομένων. Επιθέσεις που εκμεταλλεύονται την ανεπαρκή επαλήθευση των εισαγωγών στις SQL δηλώσεις. Ως αποτέλεσμα, ενδέχεται να εξαχθούν ευαίσθητα δεδομένα ή να εκτελεστούν απροσδόκητες εντολές.
- **Privilege Escalation**: Η κλιμάκωση επιπέδου πρόσβασης ή ανύψωση προνομίων είναι μια τεχνική όπου οι επιτιθέμενοι εκμεταλλεύονται ευπάθειες στον σχεδιασμό του συστήματος, λάθη διαμόρφωσης, κοινόχρηστα αρχεία και την κοινωνική μηχανική με παραπλάνηση χρηστών για να αποκτήσουν πρόσβαση σε συστήματα έχοντας υψηλότερα επίπεδα δικαιωμάτων από αυτά που αρχικά είχαν. Πρόκειται δηλαδή για κακόβουλες ενέργειες που προσπαθούν να αποκτήσουν υψηλότερα δικαιώματα πρόσβασης.

- **Data Breaches**: Οι διαρροές δεδομένων (ευαίσθητων κυρίως προσωπικών, οικονομικών, επιχειρηματικών κλπ.) σε μη εξουσιοδοτημένα άτομα λόγω ανεπαρκών μηχανισμών ασφάλειας. Προκαλείται συνεπώς λόγω ανεπαρκών πολιτικών ασφάλειας, όπως η αποθήκευση δεδομένων χωρίς κρυπτογράφηση ή άλλες αδυναμίες στα συστήματα ασφαλείας.
- **Denial of Service (DoS)**: Εκμετάλλευση ευπαθειών που προκαλούν υπερφόρτωση του συστήματος, καθιστώντας το μη διαθέσιμο. Επίθεση που στοχεύει στη μη διαθεσιμότητα της υπηρεσίας.
- **Weak Authentication Mechanisms**: Χρήση απλών ή προβλέψιμων κωδικών πρόσβασης και μηχανισμών αυθεντικοποίησης. Ανεπαρκείς μηχανισμοί αυθεντικοποίησης που επιτρέπουν τη μη εξουσιοδοτημένη πρόσβαση σε βάσεις δεδομένων, επιτρέποντας στους εισβολείς να τροποποιήσουν, να διαγράψουν ή να εξαγάγουν δεδομένα.
- **Buffer Overflows**: Μια επίθεση υπερχείλισης ενδιάμεσης μνήμης (buffer) συμβαίνει όταν ένας εισβολέας στέλνει περισσότερα δεδομένα από αυτά που μπορεί να χειριστεί η ενδιάμεση μνήμη. Οι επιθέσεις αυτές δηλαδή εκμεταλλεύονται ευπάθειες στον τρόπο που οι βάσεις δεδομένων και οι εφαρμογές διαχειρίζονται τη μνήμη, και επιτρέπουν την εκτέλεση αυθαίρετου κώδικα από τους επιτιθέμενους ή ακόμα και κατάρρευση της βάσης δεδομένων με συνέπεια τη διακοπή της λειτουργίας του συστήματος.
- **Encryption και Data Protection**: Η κρυπτογράφηση δεδομένων αποτελεί κρίσιμη τεχνική για την προστασία ευαίσθητων πληροφοριών από την αποκάλυψη και την ανεπιθύμητη πρόσβαση από μη εξουσιοδοτημένους χρήστες. Οι σχεσιακές βάσεις δεδομένων ενδέχεται να αποθηκεύουν ευαίσθητα δεδομένα, όπως οικονομικά, ιατρικά ή προσωπικά στοιχεία. Η ανεπαρκής ή αδύναμη κρυπτογράφηση μπορεί να οδηγήσει σε κλοπή δεδομένων ή παραβίαση του απορρήτου. Συνεπώς η σημασία της κρυπτογράφησης και προστασίας δεδομένων αποτελεί κρίσιμης σημασίας πτυχή της ασφάλειας βάσεων δεδομένων.
- **Auditing και Monitoring**: Η ανάλυση (Auditing) και η παρακολούθηση (Monitoring) αποτελούν κρίσιμες διαδικασίες προκειμένου να διασφαλιστεί η ασφάλεια των βάσεων δεδομένων. Αυτές οι διαδικασίες επιτρέπουν στους διαχειριστές να παρακολουθούν τις δραστηριότητες που συμβαίνουν στις βάσεις δεδομένων και να εντοπίζουν με τον τρόπο αυτό ενδεχόμενες απειλές ώστε να επιληφθούν των αναγκαίων ενεργειών για να τις αντιμετωπίσουν.

2.2 Τρόποι προστασίας για την αποτροπή προβλημάτων Ασφάλειας των RDBMS

Για την προστασία των σχεσιακών βάσεων δεδομένων από αυτά τα τρωτά σημεία, οι βέλτιστες πρακτικές περιλαμβάνουν τακτικές ενημερώσεις λογισμικού, ισχυρές πολιτικές κωδικών πρόσβασης, στοιχεία ελέγχου πρόσβασης και τεχνικές κρυπτογράφησης. Είναι επίσης σημαντικό να ακολουθούνται ασφαλείς πρακτικές κωδικοποίησης για να ελαχιστοποιηθούν οι κίνδυνοι επιθέσεων SQL injection. Οι τακτικές αξιολογήσεις ασφαλείας και οι επιθέσεις παρείσδυσης μπορούν να βοηθήσουν στον εντοπισμό τρωτών σημείων και στον μετριασμό των κινδύνων [9].

Οι πιο συνηθισμένες ευπάθειες περιλαμβάνουν DoS, εκτέλεση κώδικα, υπερχείλιση (overflow), παράκαμψη ασφαλείας, απόκτηση πληροφοριών και διαρροή δεδομένων. Είναι σημαντικό να εφαρμόζονται βασικά μέτρα ασφαλείας όπως η ταυτοποίηση, ο έλεγχος πρόσβασης, η κρυπτογράφηση δεδομένων και η ασφάλεια δικτύου για την προστασία των βάσεων δεδομένων. Η εκπαίδευση και η ενημέρωση των χρηστών σχετικά με την ασφάλεια πληροφοριών είναι επίσης κρίσιμη [7].

Παρακάτω παρατίθενται μερικά βασικά μέτρα ασφαλείας που προτείνονται [7]:

- **Αυθεντικοποίηση**: Είναι απαραίτητο να εξασφαλιστεί ότι όλοι οι χρήστες θα περάσουν από διαδικασία επαλήθευσης πριν αποκτήσουν πρόσβαση στα δεδομένα.
- **Έλεγχος Πρόσβασης**: Πρέπει να καθοριστούν αυστηρά δικαιώματα πρόσβασης για τον περιορισμό της πρόσβασης σε δεδομένα είτε απλά για να τα δει μόνο ή αν θα μπορεί και τα τροποποιήσει.
- **Κρυπτογράφηση Δεδομένων**: Χρήση κρυπτογράφησης για την προστασία των δεδομένων κατά τη μετάδοση και αποθήκευση.
- **Εκπαίδευση Χρηστών**: Ενημέρωση των χρηστών για καλές πρακτικές ασφαλείας και τη σημασία της διατήρησης της ασφάλειας των δεδομένων.

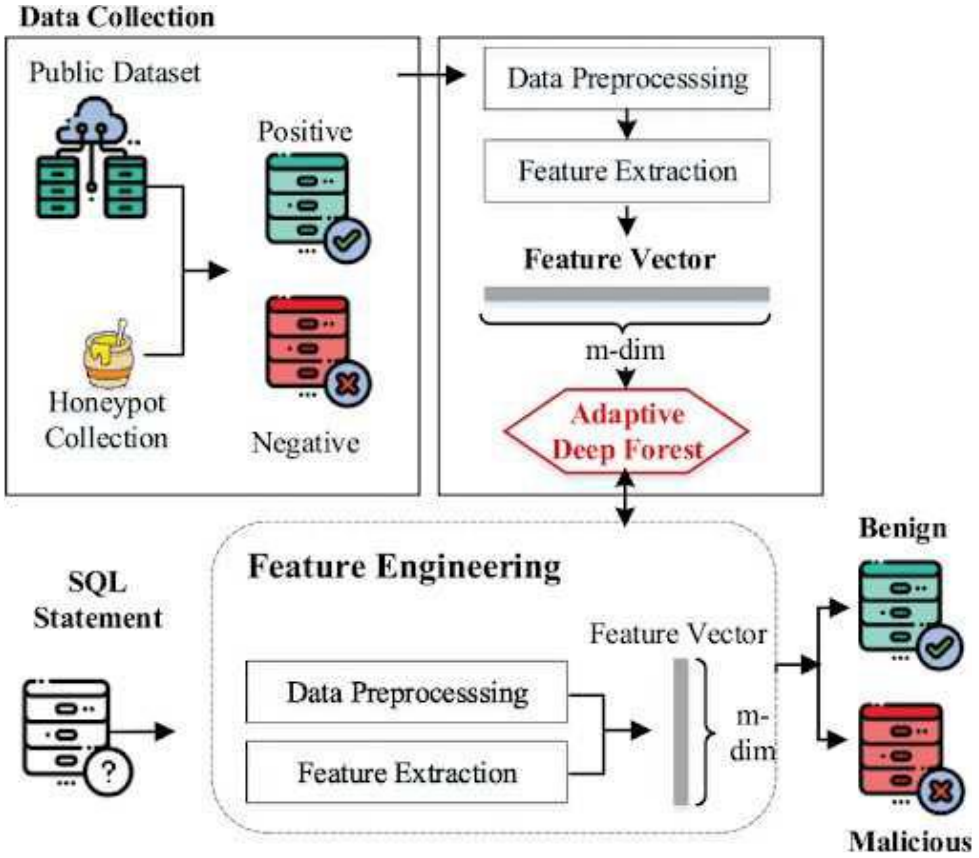
Αυτά τα μέτρα μπορούν να βοηθήσουν στην προστασία των συστημάτων DBMS από απειλές και να διασφαλίσουν την ασφάλεια των δεδομένων. Είναι σημαντικό να εφαρμόζονται συνεχώς και να ενημερώνονται με βάση τις τρέχουσες απειλές και τις καλύτερες πρακτικές [7].

2.3 Ανάλυση των Προβλημάτων Ασφάλειας

2.4 Επιθέσεις SQL Injection

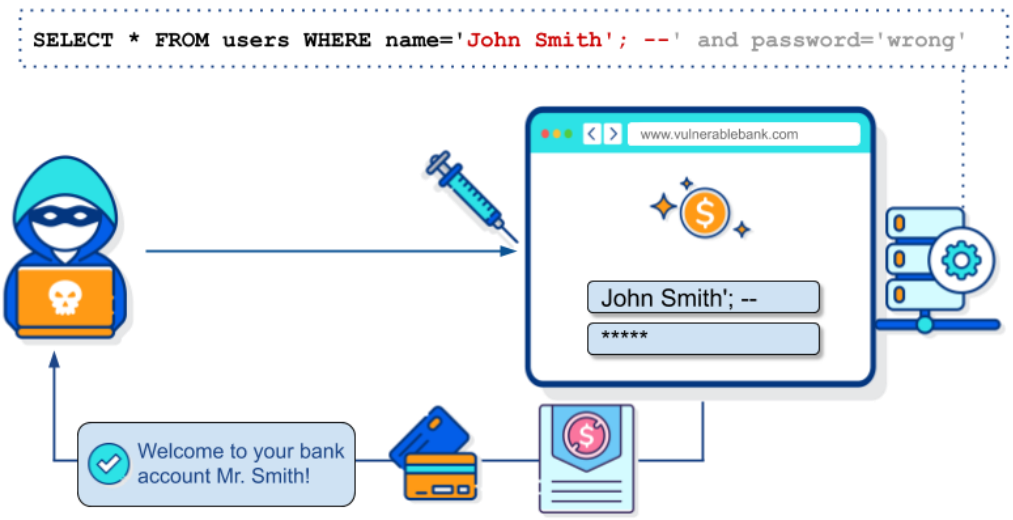
Η SQL αποτελεί μια δομημένη γλώσσα ερωτημάτων σχεσιακών βάσεων δεδομένων, την πιο διαδεδομένη, τόσο για την διαχείριση τους όσο και για την εκτέλεση διαφόρων λειτουργιών πάνω στα δεδομένα. Η SQL αποτελεί την γλώσσα που χρησιμοποιούν και οι αναλυτές και οι επιστήμονες των δεδομένων για την εύρεση δεδομένων και στις μεγάλες βάσεις δεδομένων, σε εφαρμογές όπως το Spotify, η Revolut, τραπεζικές εφαρμογές και σε ιστοτόπους κοινωνικής δικτύωσης όπως το Twitter, το Facebook και το Instagram. Χρησιμοποιεί δηλώσεις, οι οποίες είναι ομάδες λέξεων-κλειδιών που χρησιμοποιούνται για τη λήψη δεδομένων από βάσεις δεδομένων. Την SQL συνθέτουν τέσσερις διαφορετικοί τύποι εντολών: η DDL (Data Definition Language) που είναι η γλώσσα ορισμού δεδομένων, η DML (Data Manipulation Language) που είναι η γλώσσα χειρισμού δεδομένων, η DCL (Data Control Language) που είναι η γλώσσα ελέγχου δεδομένων, και τέλος η DQL (Data Query Language) που είναι η γλώσσα ελέγχου δεδομένων.

Ένας χάκερ θα χρησιμοποιήσει την SQLi (SQL Injection) για να προσπαθήσει να εισάγει ειδικά σχεδιασμένα ερωτήματα SQL σε ένα πεδίο φόρμας αντί για τα επιθυμητά δεδομένα με σκοπό να κατανοήσει τον τρόπο κατασκευής της βάσης δεδομένων, όπως για παράδειγμα να ανακαλύψει ονόματα πινάκων της βάσης. Το Σχ. 2.1 απεικονίζει πώς το βήμα εκπαίδευσης εκτός σύνδεσης και το στάδιο διαδικτυακής δοκιμής είναι κρίσιμα συστατικά της προσέγγισης ανίχνευσης SQLi [10].



Σχήμα 2.1: Εκτέλεση και εφαρμογή μοντέλου SQL [11]

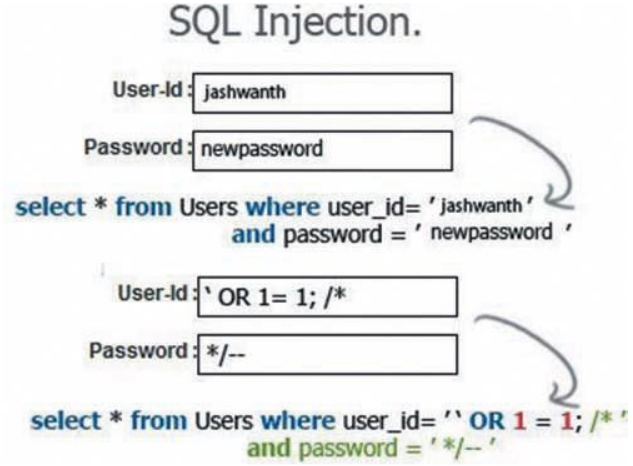
Οι επιθέσεις SQLi είναι από τις πιο διαδεδομένες και επικίνδυνες μορφές επιθέσεων που γίνονται σε βάσεις δεδομένων. Επιτρέπουν στους επιτιθέμενους να εκτελούν αυθαίρετες εντολές SQL μέσω των εφαρμογών που αλληλοεπιδρούν με τη βάση δεδομένων. Αποτελούν μια μέθοδο επίθεσης όπου ο επιτιθέμενος εκμεταλλεύεται κενά ασφαλείας σε μια εφαρμογή για να εισάγει κακόβουλες εντολές SQL, με σκοπό αυτές στη συνέχεια να εκτελεστούν στη βάση δεδομένων [8]. Ο τρόπος που επιτυγχάνεται αυτός το τύπος κυβερνοεπίθεσης είναι με την έγχυση κακόβουλου κώδικα βάσης δεδομένων στα πεδία εισόδου ενός ιστοτόπου [12]. Είναι ο πιο αποτελεσματικός τρόπος κλοπής δεδομένων από το backend των διαδικτυακών εφαρμογών. Επιτρέπουν στους εισβολείς – χάκερς, να έχουν πρόσβαση σε βάσεις δεδομένων και να επωφελούνται από ιδιωτικά δεδομένα [13]. Τα ευαίσθητα δεδομένα που ενδέχεται να υποκλαπούν, συμπεριλαμβάνουν τα αναγνωριστικά των χρηστών και τους κωδικούς πρόσβασης τους, προσωπικές πληροφορίες και οικονομικά δεδομένα, μπορούν να κλαπούν [12]. Αυτές οι επιθέσεις αναφέρονται ως επιθέσεις επικύρωσης εισόδου, καθώς συνήθως δημιουργούνται από την είσοδο Ιστού. Η πλειονότητα των εφαρμογών Ιστού έχουν πλέον παραβιαστεί από επιθέσεις SQLi. Περιλαμβάνεται μεταξύ των δέκα κορυφαίων απειλών ασφαλείας εφαρμογών ιστού [13].



Σχήμα 2.2: Έλεγχος για ευπάθειες σε SQLi [14]

Τα ακόλουθα βήματα εμπλέκονται στη διαδικασία SQLi [9]:

1. **Αρχικά προσδιορίζεται ποιο πεδίο εισαγωγής είναι αδύναμο**. Ένα πεδίο εισαγωγής σε μια εφαρμογή Ιστού που είναι επιρρεπές σε ένεση SQL, όπως μια φόρμα σύνδεσης, ένα πεδίο αναζήτησης ή μια φόρμα σχολίων, αναγνωρίζεται πρώτα από τον εισβολέα. Για να γίνει αυτό, μπορούν να χρησιμοποιηθούν είτε αυτόματα εργαλεία είτε ανασκόπηση του πηγαίου κώδικα HTML.
2. **Συγγραφή κακόβουλου ερωτήματος SQL**. Ο εισβολέας δημιουργεί ένα κακόβουλο ερώτημα SQL για να εκμεταλλευτεί την ευπάθεια του πεδίου εισαγωγής. Μπορείτε να το επιτύχετε εισάγοντας κώδικα SQL, όπως ένα μεμονωμένο εισαγωγικό (') ή ένα σχόλιο SQL (--), στο πεδίο εισαγωγής.
3. **Υποβολή του κακόβουλου ερωτήματος**. Μέσω του αδύναμου πεδίου εισαγωγής, ο εισβολέας στέλνει το κακόβουλο ερώτημα SQL στην εφαρμογή Ιστού. Μετά την επεξεργασία του ερωτήματος, η εφαρμογή Ιστού επιστρέφει στον εισβολέα τα αποτελέσματα.
4. **Ανάκτηση δεδομένων**. Ο εισβολέας μπορεί να λάβει ευαίσθητες πληροφορίες από τη βάση δεδομένων, όπως αριθμούς πιστωτικών καρτών, ονομάτων χρήστη και κωδικών πρόσβασης, εάν το SQLi είναι επιτυχής. Ανάλογα με τα δικαιώματα που δίνονται στον χρήστη της βάσης δεδομένων, ο εισβολέας μπορεί επίσης να προσθέσει, να αφαιρέσει ή να αλλάξει δεδομένα της βάσης δεδομένων.
5. **Κάλυψη του ίχνος του**. Για να καλύψει τα ίχνη του, ο εισβολέας μπορεί να επιχειρήσει να διαγράψει τις δραστηριότητές του από τα αρχεία καταγραφής ή να τροποποιήσει δεδομένα στα οποία του δόθηκε πρόσβαση.



Σχήμα 2.3: Πως προκαλείται η επίθεση SQL Injection [15]

Υπάρχουν διάφοροι τύποι απειλών SQL Injection. Οι επιθέσεις εναντίον βάσεων δεδομένων που χρησιμοποιούν SQL Injection χωρίζονται σε τέσσερις κύριες ομάδες [13]:

1. **SQL manipulation:** είναι η επέμβαση που γίνεται στην εντολή SQL ώστε να αλλάξει χρησιμοποιώντας εντολές όπως η UNION. Ένας τρόπος που αυτό επιτυγχάνεται είναι με την αλλαγή του τμήματος της πρότασης SQL που περιέχει τον όρο where για την αλλαγή των κριτηρίων της εντολής και τη λήψη διαφορετικών αποτελεσμάτων.
2. **Code Injection:** αυτό επιτυγχάνεται μέσω της προσθήκης μιας νέας πρότασης SQL. Μια τέτοια μέθοδος είναι μέσω της προσθήκης της εντολής SQL Server EXECUTE στην πρόταση SQL. Μια τέτοια επίθεση είναι δυνατόν να συμβεί στην περίπτωση και μόνο που επιτρέπονται πολλές δηλώσεις SQL ανά αίτημα βάσης δεδομένων.
3. **Function Call Injection:** είναι η μέθοδος όπου εισάγονται κλήσεις διάφορων συναρτήσεων βάσεων δεδομένων σε ένα αδύναμο ερώτημα SQL. Αυτές οι κλήσεις συναρτήσεων έχουν τη δυνατότητα της τροποποίησης δεδομένων της βάσης δεδομένων ή της εκκίνησης κλήσεων του λειτουργικού συστήματος.
4. **Υπερχείλιση buffer:** Η χρήση της παραπάνω μεθόδου function call injection μπορεί να οδηγήσει σε υπερχείλιση buffer. Για την αποτροπή εμφάνισης μιας τέτοιας επίθεσης επιβάλλονται οι ενημερώσεις του διακομιστή με τις πρόσφατες διαθέσιμες ενημερώσεις. Οι ενημερώσεις κώδικα είναι διαθέσιμες για την πλειονότητα των βάσεων δεδομένων τόσο του ανοιχτού κώδικα όσο και των εμπορικών βάσεων.

Εκτός από το απλό γέμισμα των πεδίων με κακόβουλο κώδικα σε εντελώς απροστάτευτες εφαρμογές υπάρχουν και διάφορα εργαλεία που χρησιμοποιούν οι επιτιθέμενοι για την εκμετάλλευση SQL Injection ευπαθειών, όπως το SQLMap. Οι μηχανισμοί και οι μέθοδοι εκμετάλλευσης περιλαμβάνουν διάφορες τεχνικές SQL Injection [8] [12] [16]:

- **Classic SQL Injection**: Εκμετάλλευση μέσω απλών εντολών εισαγωγής SQL σε φόρμες ή URL.
- **Blind SQL Injection**: Εκμετάλλευση όπου ο επιτιθέμενος δεν λαμβάνει άμεσα ορατά αποτελέσματα αλλά βασίζεται σε αναδρομικές τεχνικές για να εξάγει δεδομένα. Η Βάση Δεδομένων δεν εμφανίζει δεδομένα αλλά γενικά μηνύματα σφάλματος στην ιστοσελίδα. Έτσι ένας εισβολέας θέτει μια σειρά από ερωτήματα για να ανακαλύψει δεδομένα της Βάσης.
- **Time-Based Blind Injection**: Χρησιμοποιεί χρονικές καθυστερήσεις για να εξάγει πληροφορίες βάσει του χρόνου απόκρισης της βάσης δεδομένων.
- **Boolean-Based Blind Injection**: Συνάγει πληροφορίες με βάση τις απαντήσεις της εφαρμογής σε διαφορετικές εισόδους, ακόμη και χωρίς άμεση ανάδραση από τη βάση δεδομένων.

- **Error-Based SQL Injection**: Εκμετάλλευση μέσω των μηνυμάτων από σφάλματα που επιστρέφει η βάση δεδομένων για τη συλλογή πληροφοριών σχετικές με το σχήμα της βάσης δεδομένων.
- **Union-Based SQL Injection**: Συνδυάζει αποτελέσματα από πολλαπλές δηλώσεις SELECT για άντληση δεδομένων από διαφορετικούς πίνακες με τη χρήση της εντολής UNION.
- **Stacked queries SQL Injection:** Χρήση από στοιβαγμένα ερωτημάτων μέσω αυτής της τεχνικής, που αντλεί δεδομένα από τη βάση δεδομένων διαιρώντας πολλές εντολές SQL με ερωτηματικά.

Οι επιπτώσεις των SQL Injection επιθέσεων περιλαμβάνουν [8]:

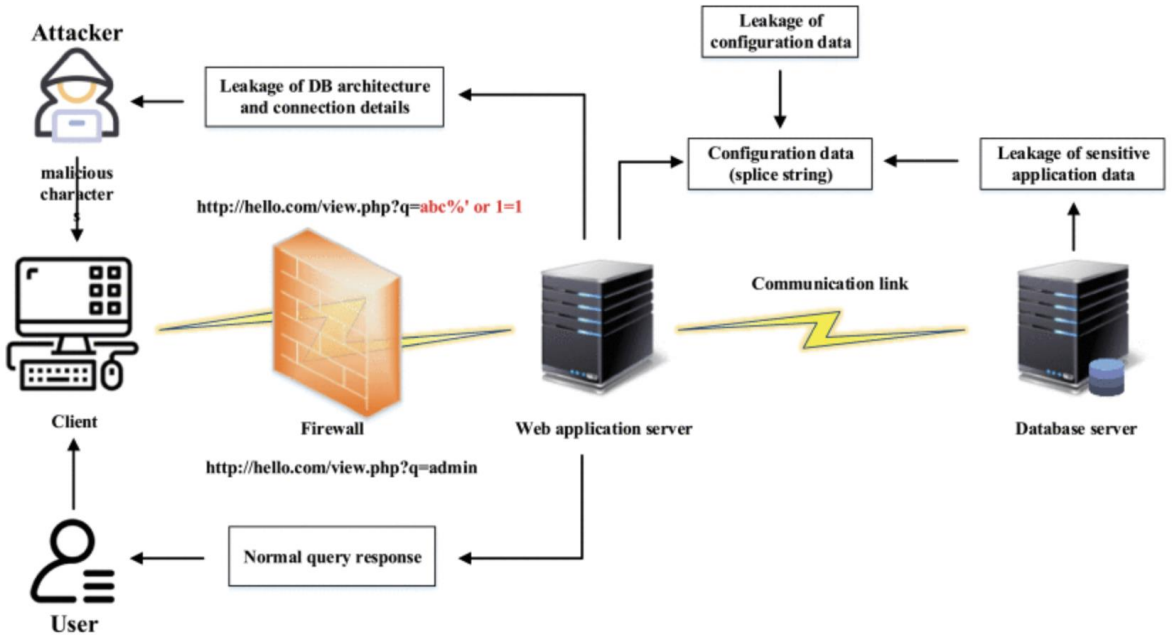
- **Ανάγνωση δεδομένων**: Οι επιτιθέμενοι μπορούν να αποκτήσουν πρόσβαση σε ευαίσθητα δεδομένα.
- **Αλλαγή ή διαγραφή δεδομένων**: Οι επιτιθέμενοι μπορούν να αλλάξουν ή να διαγράψουν δεδομένα, καταστρέφοντας την ακεραιότητα της βάσης δεδομένων.
- **Κλιμάκωση προνομίων**: Χρήση SQLi για την απόκτηση υψηλότερων προνομίων.
- **Άρνηση υπηρεσιών (Denial of Service)**: Εκτέλεση εντολών που προκαλούν διακοπή της λειτουργίας της βάσης δεδομένων.

Οι μέθοδοι πρόληψης και αντιμετώπισης SQLi επιθέσεων περιλαμβάνουν ενέργειες όπως οι παρακάτω [8]:

- **Επαλήθευση εισόδου**: Χρήση επαλήθευσης εισόδου (input validation) για να αποτραπεί η εισαγωγή κακόβουλων δεδομένων.
    - **Χρήση παραμέτρων**: Χρήση προετοιμασμένων δηλώσεων ή παραμετροποιημένων ερωτημάτων (prepared statements) και αποθηκευμένων διαδικασιών (stored procedures) για την αποφυγή SQL Injection.
    - **Escaping characters**: Χρήση ειδικών χαρακτήρων διαφυγής για την εξουδετέρωση κακόβουλων εντολών SQL.
- **Εργαλεία και Τεχνικές Άμυνας**: Περιγραφή εργαλείων που βοηθούν στον εντοπισμό και την αποτροπή SQLi επιθέσεων, όπως web application firewalls (WAFs).
- **Ασφάλεια εφαρμογών**: Βέλτιστες πρακτικές για την ανάπτυξη ασφαλών εφαρμογών που αλληλοεπιδρούν με βάσεις δεδομένων.
- **Τακτικοί έλεγχοι ασφαλείας**: Εκτέλεση τακτικών ελέγχων ασφαλείας και penetration testing για τον εντοπισμό και την αντιμετώπιση SQLi ευπαθειών.

Υπάρχουν προβλήματα με τον εντοπισμό επιθέσεων SQLi [12]:

- Υπάρχει δυσκολία εύρεσης αποτελεσματικής επικύρωσης εισόδου, καθώς οι εισβολείς μπορούν να παρακάμψουν τα φίλτρα κωδικοποιώντας τα δεδομένα εισόδου, εκμεταλλευόμενοι σχόλια και χρησιμοποιώντας άλλες τακτικές συσκότισης. Συνεπώς, είναι δύσκολη η ανάπτυξη και η εφαρμογή ολοκληρωμένων κανόνων επικύρωσης δεδομένων εισόδου.
- Οι τυφλές επιθέσεις SQLi είναι πιο δύσκολο να εντοπιστούν και να προληφθούν από τις συμβατικές επιθέσεις ένεσης SQL. Ο στόχος ενός εισβολέα σε μια τυφλή επίθεση ένεσης SQL είναι να καταστρέψει τα δεδομένα μιας εφαρμογής ιστού.
- Πολλές διαδικτυακές εφαρμογές είναι χτισμένες σε ξεπερασμένες βάσεις κωδικών που μπορεί να μην είχαν επίγνωση της ασφάλειας όταν αναπτύχθηκαν. Τα τρωτά σημεία SQLi σε αυτά τα συστήματα δεν είναι εύκολο να βρεθούν και επιπρόσθετα ίσως να χρειαστεί αρκετός χρόνος για να διορθωθούν.



Σχήμα 2.4: Διάγραμμα επίθεσης SQL injection [11]

2.5 Privilege Escalation

Το Privilege Escalation αναφέρεται στη κλιμάκωση προνομίων, και είναι η διαδικασία κατά την οποία ένας επιτιθέμενος αποκτά υψηλότερα δικαιώματα ή προνόμια αναφορικά με αυτά που κανονικά του αποδίδονται. Η κλιμάκωση προνομίων είναι η διαδικασία κατά την οποία ένας χρήστης ή μια εφαρμογή αποκτά υψηλότερα δικαιώματα ή προνόμια αναφορικά με αυτά που αρχικά τους έχουν δοθεί. Αυτό μπορεί να γίνει μέσω της εκμετάλλευσης ευπαθειών σε λογισμικό ή κακών πρακτικών διαχείρισης δικαιωμάτων [8].

Οι αιτίες και οι τεχνικές εκμετάλλευσης ευπαθειών περιλαμβάνουν [8]:

- **Λάθη στη διαχείριση δικαιωμάτων**: Κακή διαμόρφωση δικαιωμάτων πρόσβασης που επιτρέπει σε χρήστες να αποκτούν περισσότερα δικαιώματα αναφορικά με αυτά που θα έπρεπε.
- **Ευπάθειες λογισμικού**: Εκμετάλλευση σφαλμάτων στον κώδικα της βάσης δεδομένων ή στις εφαρμογές που την διαχειρίζονται.
- **Αδύναμοι κωδικοί πρόσβασης**: Χρήση αδύναμων ή κοινών κωδικών πρόσβασης που επιτρέπουν την εύκολη πρόβλεψη ή διάρρηξή τους.
- **SQL Injection**: Επιθέσεις SQLi που επιτρέπουν την εκτέλεση αυθαίρετων εντολών SQL με υψηλότερα προνόμια.

Οι μέθοδοι πρόληψης καθώς και οι τρόποι αντιμετώπισης τους αφορούν [8]:

- **Αυστηρή διαχείριση δικαιωμάτων**: Κατάλληλη ρύθμιση των δικαιωμάτων πρόσβασης για κάθε χρήστη και ρόλο, περιορίζοντας την πρόσβαση μόνο σε ό,τι είναι απαραίτητο.
- **Τακτικοί έλεγχοι και αναθεωρήσεις**: Τακτική αναθεώρηση και έλεγχος των δικαιωμάτων πρόσβασης για να διασφαλιστεί ότι είναι κατάλληλα ρυθμισμένα.
- **Ενημερώσεις και διορθώσεις**: Τακτική ενημέρωση του λογισμικού και εφαρμογή διορθωτικών ενημερώσεων για την εξάλειψη γνωστών ευπαθειών.

- **Ενίσχυση της ασφάλειας των κωδικών πρόσβασης**: Χρήση ισχυρών κωδικών πρόσβασης και εφαρμογή πολιτικών αλλαγής κωδικών πρόσβασης σε τακτά χρονικά διαστήματα.
- **Εκπαίδευση προσωπικού**: Εκπαίδευση των χρηστών και των διαχειριστών στην αναγνώριση και αντιμετώπιση των απειλών κλιμάκωσης προνομίων.

2.6 Data Breaches

Το θέμα των παραβιάσεων δεδομένων (Data Breaches) καλύπτεται στο πλαίσιο της γενικής ασφάλειας των βάσεων δεδομένων. Οι παραβιάσεις δεδομένων περιλαμβάνουν οποιοδήποτε περιστατικό όπου ευαίσθητα, προστατευμένα ή εμπιστευτικά δεδομένα εκτίθενται ή αποκτώνται από μη εξουσιοδοτημένα άτομα. Αυτό μπορεί να περιλαμβάνει προσωπικά δεδομένα, οικονομικές πληροφορίες, διαπιστευτήρια πρόσβασης και άλλα. Αποτελεί σημαντικό πρόβλημα ασφάλειας, και παρακάτω εξετάζονται οι αιτίες και οι μέθοδοι πρόληψης [8].

Στις αιτίες των παραβιάσεων δεδομένων περιλαμβάνονται [8]:

- **Ευπάθειες στις εφαρμογές**: Εκμετάλλευση αδυναμιών στις εφαρμογές που αλληλοεπιδρούν με τις βάσεις δεδομένων.
- **SQL Injection**: Πρόσβαση και εξαγωγή δεδομένων από τη βάση δεδομένων.
- **Εσφαλμένη διαμόρφωση**: Λανθασμένη ρύθμιση παραμέτρων που αφήνει τις βάσεις δεδομένων εκτεθειμένες σε επιθέσεις.
- **Ανεπαρκής αυθεντικοποίηση και εξουσιοδότηση**: Αδυναμίες στους μηχανισμούς ελέγχου ταυτότητας και δικαιωμάτων πρόσβασης.

Οι επιπτώσεις των παραβιάσεων δεδομένων αφορούν [8]:

- **Οικονομικές συνέπειες**: Κόστη για την επιδιόρθωση της παραβίασης, απώλεια πελατών, νομικές κυρώσεις και πρόστιμα.
- **Επιπτώσεις στην φήμη**: Απώλεια εμπιστοσύνης από πελάτες και συνεργάτες.
- **Νομικές συνέπειες**: Ρυθμιστικές κυρώσεις και νομικές ευθύνες.

Οι μέθοδοι πρόληψης και αντιμετώπισης τους περιλαμβάνουν [8]:

- **Τη χρήση κρυπτογράφησης δεδομένων**: Χρήση κρυπτογράφησης για την προστασία των δεδομένων τόσο κατά την αποθήκευση όσο και κατά τη μεταφορά.
- **Τη διαχείριση πρόσβασης**: Ενίσχυση των μηχανισμών αυθεντικοποίησης και εξουσιοδότησης, περιορισμός της πρόσβασης στα δεδομένα.
- **Τακτικούς ελέγχους και παρακολούθηση**: Τακτικοί έλεγχοι ασφαλείας και συνεχής παρακολούθηση των συστημάτων για την ανίχνευση ανωμαλιών.
- **Την εκπαίδευση του προσωπικού**: Εκπαίδευση των χρηστών και των διαχειριστών βάσεων δεδομένων σε θέματα ασφάλειας.

2.7 Denial of Service (DoS)

Η μορφή αυτή επιθέσεων Άρνησης Υπηρεσιών (Denial of Service - DoS), έχουν ως στόχο να καταστήσουν μια υπηρεσία ή έναν πόρο μη διαθέσιμο στους νόμιμους χρήστες. Αυτό μπορεί να επιτευχθεί προκαλώντας υπερφόρτωση των πόρων του συστήματος ή εκμεταλλευόμενες ευπάθειες που προκαλούν σφάλματα. Η γνώση των στρατηγικών και οι τεχνικές για την ανίχνευση και την αποτροπή αυτών των επιθέσεων, είναι χρήσιμες για όσους επιθυμούν να προστατεύσουν τις βάσεις δεδομένων και τις εφαρμογές τους από DoS επιθέσεις [8].

Στους μηχανισμούς και τις μεθόδους εκμετάλλευσης περιλαμβάνονται [8]:

- **Τύποι DoS επιθέσεων**:
    - **Resource Exhaustion**: Επίθεση μέσω της κατανάλωσης όλων των διαθέσιμων πόρων, όπως CPU, μνήμη ή bandwidth.
    - **Vulnerability Exploitation**: Επίθεση που εκμεταλλεύεται συγκεκριμένες ευπάθειες σε εφαρμογές ή βάσεις δεδομένων για να προκαλέσει κατάρρευση.
    - **Flood Attacks**: Αποστολή μεγάλου όγκου δεδομένων ή αιτημάτων για την υπερφόρτωση του συστήματος.
- **Μέθοδοι Εκμετάλλευσης:** Οι τεχνικές που εφαρμόζονται για την εκμετάλλευση των ευπαθειών που προκαλούν DoS, είναι ειδικά διαμορφωμένα αιτήματα που προκαλούν υπερχείλιση μνήμης ή σφάλματα εφαρμογών.
- **Εργαλεία και τεχνικές:** Στα εργαλεία που χρησιμοποιούνται για την εκτέλεση DoS επιθέσεων, είναι τα stress testing tools και τα DoS frameworks.

Στις επιπτώσεις των DoS Attacks περιλαμβάνονται [8]:

- **Διακοπή υπηρεσιών**: Οι επιθέσεις DoS μπορούν να κάνουν μια βάση δεδομένων ή εφαρμογή μη διαθέσιμη στους χρήστες.
- **Απώλεια εσόδων**: Για επιχειρήσεις που βασίζονται σε online υπηρεσίες, οι διακοπές μπορούν να οδηγήσουν σε σημαντικές οικονομικές απώλειες.
- **Απώλεια εμπιστοσύνης**: Οι επαναλαμβανόμενες διακοπές μπορούν να βλάψουν τη φήμη της εταιρείας και να μειώσουν την εμπιστοσύνη των πελατών.

Οι μέθοδοι πρόληψης καθώς και οι τρόποι αντιμετώπισης τους αφορούν [8]:

- **Ανίχνευση και παρακολούθηση**: Χρήση εργαλείων ανίχνευσης ανωμαλιών και παρακολούθησης της δικτυακής κυκλοφορίας για τον εντοπισμό επιθέσεων.
    - **Rate Limiting**: Εφαρμογή περιορισμών στο ρυθμό των αιτημάτων που μπορούν να γίνουν δεκτά από το σύστημα.
    - **Resource Management**: Βελτιστοποίηση της διαχείρισης των πόρων του συστήματος για να αντέχει σε υψηλά φορτία.
- **Εργαλεία και Τεχνικές Άμυνας**: Στα εργαλεία και τις μεθόδους για την αποτροπή και την αντιμετώπιση DoS επιθέσεων, είναι τα web application firewalls (WAFs) και τα συστήματα κατανομής φορτίου (load balancers).
- **Ανθεκτικότητα συστήματος**: Σχεδίαση και υλοποίηση ανθεκτικών συστημάτων που μπορούν να ανακάμπτουν γρήγορα από DoS επιθέσεις.
- **Εκπαίδευση και ασκήσεις**: Εκπαίδευση του προσωπικού για την αναγνώριση και την άμεση ανταπόκριση σε DoS επιθέσεις, καθώς και εκτέλεση τακτικών ασκήσεων για τη δοκιμή της ανθεκτικότητας του συστήματος.

2.8 Weak Authentication Mechanisms

Οι αδύναμοι μηχανισμοί αυθεντικοποίησης (Weak Authentication Mechanisms) αποτελούν ένα από τα πιο σημαντικά προβλήματα ασφάλειας, καθώς επιτρέπουν στους επιτιθέμενους να αποκτούν μη εξουσιοδοτημένη πρόσβαση σε βάσεις δεδομένων και άλλα ευαίσθητα συστήματα. Αδύναμοι μηχανισμοί αυθεντικοποίησης είναι αυτοί που δεν παρέχουν επαρκή ασφάλεια για να προστατεύσουν την πρόσβαση σε συστήματα και δεδομένα. Αυτό μπορεί να περιλαμβάνει τη χρήση αδύναμων κωδικών πρόσβασης, έλλειψη κρυπτογράφησης ή ανεπαρκείς μεθόδους ταυτοποίησης.

Στους μηχανισμούς και τις μεθόδους εκμετάλλευσης περιλαμβάνονται [8]:

- **Τύποι αδυναμιών**:
    - **Αδύναμοι κωδικοί πρόσβασης**: Χρήση εύκολων προβλέψιμων ή κοινοί κωδικοί πρόσβασης που μπορούν εύκολα να παραβιαστούν μέσω brute force (εξαντλητικών) επιθέσεων ή dictionary επιθέσεων.
    - **Μη κρυπτογραφημένα credentials (διαπιστευτήρια)**: Μεταφορά κωδικών πρόσβασης και άλλων credentials χωρίς κρυπτογράφηση, επιτρέποντας την υποκλοπή τους.
    - **Έλλειψη multifactor authentication (MFA)**: Απουσία πρόσθετων επιπέδων ασφάλειας, όπως η χρήση δύο παραγόντων ταυτοποίησης.
    - **Ευπάθειες στα πρωτόκολλα αυθεντικοποίησης**: Χρήση παρωχημένων ή μη ασφαλών πρωτοκόλλων αυθεντικοποίησης που μπορούν να εκμεταλλευτούν οι επιτιθέμενοι.

Οι επιθέσεις brute force είναι μία από τις βασικές τεχνικές που χρησιμοποιούνται για την παραβίαση κωδικών πρόσβασης και την πρόσβαση σε συστήματα βάσεων δεδομένων. Οι επιθέσεις brute force περιλαμβάνουν τη συστηματική δοκιμή όλων των πιθανών συνδυασμών κωδικών πρόσβασης μέχρι να βρεθεί ο σωστός. Οι επιθέσεις brute force απαιτούν σημαντική υπολογιστική ισχύ και χρόνο, ιδιαίτερα όταν οι κωδικοί πρόσβασης είναι μεγάλοι και σύνθετοι.

Υπάρχουν διάφορες τεχνικές Brute Force, όπως [8]:

- **Επιθέσεις Λεξικού (Dictionary Attacks)**: Χρήση προκαθορισμένων λιστών με κοινά χρησιμοποιούμενους κωδικούς πρόσβασης και φράσεις για την επιτάχυνση της διαδικασίας.
- **Υβριδικές Επιθέσεις (Hybrid Attacks)**: Συνδυασμός επιθέσεων λεξικού και brute force, όπου οι προκαθορισμένοι κωδικοί συνδυάζονται με παραλλαγές και πρόσθετους χαρακτήρες.
- **Rainbow Tables**: Χρήση προϋπολογισμένων καταλόγων κρυπτογραφικών hash για την ταχύτερη αντιστοίχιση των hash των κωδικών πρόσβασης με τις πραγματικές τιμές.

Οι επιθέσεις λεξικού αναφέρονται ως μια κοινή μέθοδος για την παραβίαση κωδικών πρόσβασης. Οι επιθέσεις αυτές χρησιμοποιούν προκαθορισμένες λίστες λέξεων και φράσεων, οι οποίες είναι συνήθως γνωστοί ή κοινώς χρησιμοποιούμενοι κωδικοί πρόσβασης, για να δοκιμάσουν και να παραβιάσουν λογαριασμούς. Χρήση λιστών που περιέχουν κοινούς κωδικούς πρόσβασης, λέξεις από λεξικά, και άλλες εύκολα προβλέψιμες φράσεις. Προσαρμογή των λιστών λέξεων για να περιλαμβάνουν παραλλαγές και συνηθισμένες τροποποιήσεις που χρησιμοποιούν οι χρήστες, όπως αντικατάσταση γραμμάτων με αριθμούς (π.χ. "password" γίνεται "p@ssw0rd"). Επειδή οι επιθέσεις dictionary δεν δοκιμάζουν όλους τους πιθανούς συνδυασμούς, αλλά μόνο τους πιο πιθανούς, μπορεί να είναι ταχύτερες και πιο αποτελεσματικές από τις επιθέσεις brute force.

Κάποια παραδείγματα εργαλείων για Brute Force και Επιθέσεις Λεξικού είναι τα [8]:

- **Hydra**: Ένα ευέλικτο εργαλείο για brute force και dictionary επιθέσεις που μπορεί να χρησιμοποιηθεί για την πραγματοποίηση επιθέσεων σε διάφορες υπηρεσίες και πρωτόκολλα δικτύου (όπως FTP, HTTP κλπ.).
- **John the Ripper**: Ένα εργαλείο γνωστό για το σπάσιμο κωδικών πρόσβασης που υποστηρίζει διάφορους τύπους κρυπτογραφικών hash και είναι ιδιαίτερα παραμετροποιήσιμο με διάφορες λίστες λέξεων.
- **Hashcat:** Ένα ισχυρό εργαλείο για την εκτέλεση brute force attacks χρησιμοποιώντας την ισχύ της GPU, καθιστώντας τη διαδικασία ταχύτερη και πιο αποτελεσματική.

- **Cain & Abel**: Εκτός από τις δυνατότητες παρακολούθησης δικτύου και ανάκτησης κωδικών πρόσβασης, το εργαλείο αυτό υποστηρίζει και dictionary επιθέσεις για την παραβίαση λογαριασμών.

Για την αντιμετώπισης τέτοιων επιθέσεων συμπεριλαμβάνονται τα εξής μέτρα [8]:

- **Αποφυγή Χρήσης Κοινών Κωδικών Πρόσβασης:** Εκπαίδευση των χρηστών να αποφεύγουν τη χρήση κοινών και εύκολα προβλέψιμων κωδικών πρόσβασης.
- **Ισχυρές Πολιτικές Κωδικών Πρόσβασης (Complex Password Policies)**: Εφαρμογή πολιτικών για τη δημιουργία σύνθετων και μεγάλων κωδικών πρόσβασης που δυσκολεύουν τις επιθέσεις brute force και τις επιθέσεις dictionary.
- **Μηχανισμοί κλειδώματος λογαριασμών** (**Account Lockout Mechanisms)**: Χρήση μηχανισμών κλειδώματος λογαριασμών μετά από έναν συγκεκριμένο αριθμό αποτυχημένων προσπαθειών σύνδεσης για την αποτροπή συνεχιζόμενων επιθέσεων.
- **Αυθεντικοποίηση Πολλαπλών Παραγόντων (Multi-Factor Authentication - MFA)**: Ενίσχυση της ασφάλειας μέσω της απαίτησης του ελέγχου της ταυτότητας πολλών παραγόντων, πέρα από τον κωδικό πρόσβασης.

Η υποκλοπή μη κρυπτογραφημένων credentials (διαπιστευτηρίων όπως ονόματα χρηστών και κωδικοί πρόσβασης) αποτελεί μία από τις πιο κοινές και επικίνδυνες επιθέσεις ασφαλείας των συστημάτων. Η κατανόηση των μεθόδων που χρησιμοποιούνται για την υποκλοπή αυτών των διαπιστευτηρίων και η εφαρμογή των κατάλληλων μέτρων προστασίας είναι κρίσιμη για την ασφάλεια των βάσεων δεδομένων και των συστημάτων γενικότερα. Οι κύριες στρατηγικές αντιμετώπισης περιλαμβάνουν τη χρήση κρυπτογράφησης για την προστασία των δεδομένων κατά τη μετάδοση και την εκπαίδευση των χρηστών για την αναγνώριση και αποφυγή επιθέσεων.

Υπάρχουν τα εργαλεία που χρησιμοποιούνται για την εκτέλεση επιθέσεων σε αδύναμους μηχανισμούς αυθεντικοποίησης, όπως password crackers και network sniffers. Αναλυτικότερα, οι μέθοδοι και τα εργαλεία:

1. **Network Sniffing:** Επιτρέπουν την παρακολούθηση της δικτυακής κυκλοφορίας για την ανίχνευση και καταγραφή δεδομένων που μεταδίδονται μη κρυπτογραφημένα, για να συμβάλλει στον εντοπισμό ευπαθειών, κακόβουλων δραστηριοτήτων αλλά και στον εντοπισμό ευαίσθητων δεδομένων.

   Τα εργαλεία που μπορούν να χρησιμοποιηθούν [8]:

   - **Wireshark**: Ισχυρό εργαλείο για την ανάλυση δικτυακής κυκλοφορίας που μπορεί να καταγράψει και να εμφανίσει μη κρυπτογραφημένα δεδομένα. Παρέχει λεπτομερείς πληροφορίες για τα πακέτα δεδομένων που μεταδίδονται μέσω δικτύου και υποστηρίζει διάφορα πρωτόκολλα.
   - **tcpdump**: Εργαλείο γραμμής εντολών που επιτρέπει την καταγραφή και ανάλυση της δικτυακής κυκλοφορίας. Είναι ελαφρύ και ισχυρό, κατάλληλο για χρήση σε διάφορα λειτουργικά συστήματα. Ως εργαλείο γραμμής εντολών, είναι ιδανικό να εκτελείται σε απομακρυσμένους διακομιστές ή συσκευές για τις οποίες δεν υπάρχει διαθέσιμο GUI, για τη συλλογή δεδομένων. Περιλαμβάνει πολλές επιλογές και φίλτρα δίνοντας για παράδειγμα τη δυνατότητα αποθήκευσης συγκεκριμένων πακέτων δεδομένων.
   - **Ettercap**: Ένα εργαλείο που υποστηρίζει επιθέσεις man-in-the-middle και ανάλυση δικτυακής κυκλοφορίας σε πραγματικό χρόνο. Είναι ιδανικό για την εκτέλεση επιθέσεων παρακολούθησης και υποκλοπής δεδομένων.

- **Cain & Abel**: Εκτός από την ανάκτηση κωδικών πρόσβασης (μπορεί να σπάσει και κρυπτογραφημένους κωδικούς πρόσβασης), αυτό το εργαλείο περιλαμβάνει ποικιλία επιλογών χρήσης όπως παρακολούθηση δικτυακής κυκλοφορίας. Τρέχει μόνο σε Windows περιβάλλον.

Για την αποφυγή, επιβάλλεται χρήση των εξής μέτρων:

- Χρήση κρυπτογράφησης (TLS/SSL) για την ασφαλή μετάδοση δεδομένων.
- Χρήση VPN για την κρυπτογράφηση της συνολικής δικτυακής κυκλοφορίας.

2. **Επιθέσεις Man-in-the-Middle (MITM):** Ο επιτιθέμενος παρεμβάλλεται στην επικοινωνία μεταξύ δύο μερών και υποκλέπτει δεδομένα που μεταδίδονται μη κρυπτογραφημένα.

   Τα εργαλεία που μπορούν να χρησιμοποιηθούν [8]:

   - **Ettercap**: Ένα εργαλείο που επιτρέπει την εκτέλεση επιθέσεων MITM και την καταγραφή μη κρυπτογραφημένων δεδομένων.
   - **Cain & Abel**: Ένα εργαλείο που μπορεί να εκτελέσει MITM επιθέσεις και να υποκλέψει κωδικούς πρόσβασης.

   Για την αποφυγή, επιβάλλεται χρήση των εξής μέτρων:

   - Χρήση κρυπτογράφησης για την ασφαλή μετάδοση δεδομένων.
   - Επαλήθευση των πιστοποιητικών SSL/TLS για την αποτροπή MITM επιθέσεων.

3. **Session Hijacking:** Η υποκλοπή ενός ενεργού session μεταξύ ενός χρήστη και ενός server για την απόκτηση μη κρυπτογραφημένων διαπιστευτηρίων.

   Τα εργαλεία που μπορούν να χρησιμοποιηθούν [8]:

   - **Wireshark και Ettercap** μπορούν να χρησιμοποιηθούν για την ανίχνευση και καταγραφή sessions των χρηστών.

   Για την αποφυγή, επιβάλλεται χρήση των εξής μέτρων:

   - Χρήση κρυπτογράφησης για την ασφαλή μετάδοση δεδομένων.
   - Χρήση secure cookies και άλλων τεχνικών για την προστασία των sessions.

4. **Phishing:** Η χρήση απατηλών emails ή ιστοσελίδων για την απόσπαση μη κρυπτογραφημένων διαπιστευτηρίων από τους χρήστες.

   Το εργαλείο που μπορεί να χρησιμοποιηθεί [8]:

   - **Social Engineering Toolkit (SET)**: Ένα εργαλείο για τη δημιουργία και εκτέλεση phishing επιθέσεων. Είναι ένα εργαλείο ανοιχτού κώδικα σχεδιασμένο για κοινωνική μηχανική. Το SET διαθέτει έναν αριθμό προσαρμοσμένων διανυσμάτων επίθεσης που επιτρέπουν τη δημιουργία δοκιμών αξιόπιστων επιθέσεων γρήγορα. [17]

   Για την αποφυγή, επιβάλλεται να εφαρμοστούν τα εξής μέτρα:

   - Εκπαίδευση των χρηστών για την αναγνώριση phishing επιθέσεων.
   - Χρήση τεχνικών ελέγχου ταυτότητας πολλαπλών παραγόντων (MFA).

Οι επιπτώσεις των αδύναμων μηχανισμών αυθεντικοποίησης περιλαμβάνουν [8]:

- **Μη εξουσιοδοτημένη πρόσβαση**: Οι επιτιθέμενοι μπορούν να έχουν πρόσβαση σε ευαίσθητα δεδομένα και συστήματα.
- **Αλλαγή ή διαγραφή δεδομένων**: Οι επιτιθέμενοι μπορούν να αλλοιώσουν ή να διαγράψουν δεδομένα, καταστρέφοντας την ακεραιότητα της βάσης δεδομένων.
- **Κλιμάκωση προνομίων**: Οι επιτιθέμενοι μπορούν να χρησιμοποιήσουν κλεμμένα credentials για να αποκτήσουν υψηλότερα προνόμια στο σύστημα.
- **Απώλεια δεδομένων**: Οι επιτιθέμενοι μπορούν να εξάγουν ή να καταστρέψουν ευαίσθητα δεδομένα, προκαλώντας απώλεια για την επιχείρηση.

Οι μέθοδοι πρόληψης και αντιμετώπισης τους περιλαμβάνει [8]:

- **Ισχυροί κωδικοί πρόσβασης**: Χρήση ισχυρών, πολυπλοκότητας κωδικών πρόσβασης που είναι δύσκολο να προβλεφθούν ή να παραβιαστούν.
    - **Multifactor authentication (MFA)**: Εφαρμογή ελέγχου ταυτότητας δύο ή περισσότερων παραγόντων ταυτοποίησης για αποτροπή πρόσβασης σε λογαριασμό ακόμα κι αν γνωρίζουν τον κωδικό, από άλλα μη εξουσιοδοτημένα άτομα, για την ενίσχυση της ασφάλειας.
    - **Κρυπτογράφηση credentials**: Χρήση κρυπτογράφησης για την προστασία των credentials κατά τη μεταφορά και την αποθήκευση.
    - **Τακτική αλλαγή κωδικών πρόσβασης**: Επιβολή πολιτικής τακτικής αλλαγής κωδικών πρόσβασης για να μειωθεί ο κίνδυνος παραβίασης.
- **Εργαλεία και Τεχνικές Άμυνας**: Χρήση εργαλείων και μεθόδων κατάλληλων ώστε να αποτραπούν επιθέσεις σε αδύναμους μηχανισμούς αυθεντικοποίησης, όπως password managers και intrusion detection systems (IDS).
- **Εκπαίδευση χρηστών**: Εκπαίδευση των χρηστών και των διαχειριστών για την αναγνώριση και την αποφυγή των αδύναμων μηχανισμών αυθεντικοποίησης και την εφαρμογή των βέλτιστων πρακτικών ασφάλειας.
- **Τακτικοί έλεγχοι ασφαλείας**: Διεξαγωγή τακτικών ελέγχων ασφαλείας και penetration testing για την αξιολόγηση και βελτίωση των μηχανισμών αυθεντικοποίησης [8].

2.9 Buffer Overflows

Οι επιθέσεις Buffer Overflows εκμεταλλεύονται ευπάθειες στον τρόπο που οι βάσεις δεδομένων και οι εφαρμογές διαχειρίζονται τη μνήμη, δίνοντας τη δυνατότητα στους επιτιθέμενους να εκτελούν αυθαίρετο κώδικα ή να προκαλέσουν διακοπή της λειτουργίας του συστήματος. Οι επιθέσεις Buffer Overflow συμβαίνουν όταν ένας επιτιθέμενος εισάγει περισσότερα δεδομένα σε ένα buffer (προσωρινή μνήμη) από ό,τι αυτός μπορεί να χειριστεί. Αυτό μπορεί να προκαλέσει υπερχείλιση των δεδομένων σε γειτονικές περιοχές μνήμης, επιτρέποντας την εκτέλεση κακόβουλου κώδικα ή την καταστροφή δεδομένων.

Οι τύποι Buffer Overflow επιθέσεων είναι [8]**:**

- **Stack-based Buffer Overflow**: Εκμετάλλευση υπερχείλισης στο stack, συνήθως για την εκτέλεση αυθαίρετου κώδικα.
- **Heap-based Buffer Overflow**: Εκμετάλλευση υπερχείλισης στο heap, που μπορεί να οδηγήσει σε αυθαίρετη τροποποίηση της μνήμης.

Τα εργαλεία που χρησιμοποιούνται για τον εντοπισμό και την εκμετάλλευση των Buffer Overflow ευπαθειών περιλαμβάνουν τους debuggers και fuzzers και υπάρχουν διάφορες τεχνικές όπως το Fuzzing και το Spiking. Το fuzzing (ή fuzz) είναι μια αυτοματοποιημένη τεχνική δοκιμής λογισμικού που δίνονται ως είσοδοι σε ένα πρόγραμμα διάφορες παραλλαγές εισόδου όπως μη έγκυρες τιμές, απροσδόκητες τιμές ή τυχαία δεδομένα. Ο σκοπός της είναι η ανακάλυψη τρωτών σημείων, σφαλμάτων ή αδυναμιών ασφάλειας σε προγράμματα όπως ελλιπείς έλεγχοι εισόδου, ειδικά εκείνων που επεξεργάζονται εισόδους από χρήστες ή από εξωτερικές πηγές. Στο spiking, η εστίαση είναι στον εντοπισμό τρωτών σημείων σε συγκεκριμένες εντολές [18].

Οι επιπτώσεις των Buffer Overflow επιθέσεων αφορούν [8]:

- **Εκτέλεση αυθαίρετου κώδικα**: Οι επιτιθέμενοι μπορούν να εκτελέσουν κακόβουλο κώδικα με τα δικαιώματα της εφαρμογής που επηρεάζεται.
- **Αλλαγή ή διαγραφή δεδομένων**: Οι επιτιθέμενοι μπορούν να αλλοιώσουν δεδομένα στη μνήμη ή στο σύστημα αρχείων.
- **Κατάρρευση εφαρμογών**: Οι επιτιθέμενοι μπορούν να προκαλέσουν διακοπή της λειτουργίας εφαρμογών ή υπηρεσιών.
- **Κλιμάκωση προνομίων**: Οι επιτιθέμενοι μπορούν να αποκτήσουν υψηλότερα προνόμια στο σύστημα εκμεταλλευόμενοι την υπερχείλιση.

Οι μέθοδοι πρόληψης και αντιμετώπισης περιλαμβάνουν [8]:

- **Χρήση ασφαλών συναρτήσεων**: Αντικατάσταση των μη ασφαλών συναρτήσεων διαχείρισης μνήμης με ασφαλέστερες εκδόσεις που ελέγχουν τα όρια.
    - **Εφαρμογή τεχνικών αποφυγής**: Χρήση τεχνικών όπως stack canaries, ASLR (Address Space Layout Randomization) και DEP (Data Execution Prevention) για την αποτροπή των Buffer Overflow επιθέσεων.
        - **Stack canaries**. Το όνομα δόθηκε ως αναλογία με το καναρίνι σε ανθρακωρυχείο. Όπως οι ανθρακωρύχοι χρησιμοποιούσαν κάποτε καναρίνια για να προειδοποιήσουν για επιβλαβή αέρια daedal, τα Stack Canaries προειδοποιούν τους προγραμματιστές για μια πιθανή επίθεση υπερχείλισης buffer. Για την αποφυγή τέτοιων εκμεταλλεύσεων, χρησιμοποιούνται τα Stack Canaries που είναι μικρές, τυχαιοποιημένες τιμές που τοποθετούνται στη στοίβα, ακριβώς πριν από τα δεδομένα ελέγχου. Οποιαδήποτε προσπάθεια αντικατάστασης αυτών των δεδομένων και επιστροφής διευθύνσεων (ένα κρίσιμο πρώτο βήμα σε μια επίθεση υπερχείλισης buffer) θα μπορούσε να σταματήσει στα ίχνη της.

          Η λειτουργία του Stack Canaries είναι απλή: κάθε φορά που καλείται μια συνάρτηση, δημιουργείται μια τιμή Stack Canary και τοποθετείται στη μνήμη, αμέσως μετά τη μνήμη που έχει δεσμευτεί για τοπικές μεταβλητές. Πριν τελειώσει η συνάρτηση, το σύστημα ελέγχει την τιμή Canary. Αν παραμείνει ίδια, η λειτουργία τελειώνει κανονικά, εάν έχει αλλάξει, το σύστημα υποθέτει ότι ξεκίνησε μια επίθεση υπερχείλισης buffer και ειδοποιεί συναγερμό, τερματίζοντας τη λειτουργία αμέσως [19].

        - **ASLR:** Η τυχαιοποίηση της διάταξης του χώρου διευθύνσεων (Address space layout randomization - ASLR) είναι μια μέθοδος που εφαρμόζεται για να αυξήσει τη δυσκολία εκτέλεσης μιας επίθεσης υπερχείλισης buffer που απαιτεί από τον εισβολέα να γνωρίζει τη θέση ενός εκτελέσιμου κώδικα στη μνήμη. Όταν ένας εισβολέας επιχειρήσει να εκμεταλλευτεί μια εσφαλμένη τοποθεσία

του χώρου διευθύνσεων, αυτομάτως η εφαρμογή στόχος θα διακοπεί, αποκρούοντας την επίθεση και ειδοποιώντας το σύστημα.

Το ASLR δημιουργήθηκε αρχικά ως ενημερωμένη έκδοση κώδικα Linux το 2001 και αργότερα ενσωματώθηκε στο λειτουργικό σύστημα Windows ξεκινώντας το 2007 στην έκδοση Vista. Να σημειωθεί πως πριν την χρήση του ASLR, οι θέσεις μνήμης των εφαρμογών είτε ήταν γνωστές είτε προσδιορίζονταν εύκολα [20] [21].

- **DEP:** Η πρόληψη εκτέλεσης δεδομένων (Data Execution Prevention - DEP) είναι ένα χαρακτηριστικό ασφαλείας που εφαρμόζεται σε σύγχρονα λειτουργικά συστήματα και αρχιτεκτονικές επεξεργαστών. Ορίζοντας ορισμένες περιοχές της μνήμης ως μη εκτελέσιμες, το DEP παρέχει ένα πρόσθετο επίπεδο προστασίας έναντι αυτών των επιθέσεων. Όταν ένα πρόγραμμα επιχειρεί να εκτελέσει κώδικα από μια μη εκτελέσιμη περιοχή μνήμης, το DEP ενεργοποιεί μια εξαίρεση, διακόπτοντας την εκτέλεση και ειδοποιώντας το λειτουργικό σύστημα. Αυτή η προληπτική προσέγγιση βοηθά στην αποφυγή της εκμετάλλευσης των τρωτών σημείων υπερχείλισης buffer, καθιστώντας σημαντικά δυσκολότερο για τους εισβολείς να παραβιάσουν συστήματα [22].

- **Εργαλεία και Τεχνικές Άμυνας**: Είναι εργαλεία για τον εντοπισμό και την πρόληψη των Buffer Overflow, όπως static analysis tools και runtime protections.
- **Ανασκόπηση κώδικα**: Τακτική ανασκόπηση και ανάλυση του κώδικα για τον εντοπισμό πιθανών ευπαθειών.
- **Εκπαίδευση προγραμματιστών**: Εκπαίδευση των προγραμματιστών στη σωστή διαχείριση της μνήμης και στις βέλτιστες πρακτικές προγραμματισμού ώστε να αποφευχθούν οι ευπάθειες τύπου Buffer Overflow.

2.10 Κρυπτογράφηση και προστασία δεδομένων (Encryption και Data Protection)

Η σημασία και η ανάγκη κρυπτογράφησης των δεδομένων μπορεί να κατανοηθεί μέσα από τους ενδεχόμενους κίνδυνους και τα ζητήματα που μπορούν να προκύψουν όπως [8]:

- **Η αποκρυπτογράφηση δεδομένων**: Αρκεί να καταδειχτεί σε όλους τους εμπλεκόμενους η σημασία της κρυπτογράφησης για την προστασία των δεδομένων κατά την αποθήκευση και τη μεταφορά τους σε βάσεις δεδομένων.
- **Η προστασία ευαίσθητων πληροφοριών**: Η κρυπτογράφηση είναι κρίσιμη για την προστασία ευαίσθητων πληροφοριών όπως είναι τα προσωπικά δεδομένα, οι πιστωτικές κάρτες, αλλά και εμπιστευτικές επιχειρηματικές πληροφορίες.
- **Τα αποτελέσματα παραβίασης**: Είναι απαραίτητο να εξετάζονται και να αναλύονται οι σοβαρές συνέπειες που μπορεί να έχει η παραβίαση ασφαλείας δεδομένων λόγω έλλειψης κρυπτογράφησης. Θα πρέπει να υπάρχει σχέδιο για τις επιπτώσεις παραβίασης και τη διαδικασία ανάκτησης και αποκρυπτογράφησης δεδομένων σε περίπτωση απώλειας πρόσβασης σε κρυπτογραφημένα δεδομένα.

Αναγκαία η πραγματοποίηση προληπτικών δράσεων τόσο για την αποτροπή όσο και την αντιμετώπιση παραβιάσεων κρυπτογράφησης. Η εφαρμογή καλών πρακτικών μπορούν να περιλαμβάνουν την [8]:

- **Εκπαίδευση και εποπτεία**: Η σημασία της συνεχούς εκπαίδευσης του προσωπικού και της εποπτείας για την τήρηση καλών πρακτικών κρυπτογράφησης.
- **Διαχείριση κλειδιών**: Η διαχείριση και η προστασία των κρυπτογραφικών κλειδιών ως κρίσιμος παράγοντας για την ασφάλεια των δεδομένων.

2.11 Auditing και Monitoring

Η ανάλυση (Auditing) και η παρακολούθηση (Monitoring) αποτελούν κρίσιμες διαδικασίες για τη διασφάλιση της ασφάλειας των βάσεων δεδομένων. Αυτές οι διαδικασίες επιτρέπουν στους διαχειριστές να παρακολουθούν τις δραστηριότητες στη βάση δεδομένων, να εντοπίζουν ενδεχόμενες απειλές και να αναλαμβάνουν τα αναγκαία μέτρα για την αντιμετώπισή τους.

Είναι σημαίνουσα η ανάγκη για λεπτομερή καταγραφή και ανάλυση των δραστηριοτήτων των χρηστών και των εφαρμογών που έχουν πρόσβαση στη βάση δεδομένων για ανάλυση δραστηριοτήτων όπως σημαντική και η σημασία της παρακολούθησης για την ανίχνευση ανωμαλιών και απειλών στη βάση δεδομένων.

Στις τεχνικές ανίχνευσης αξιοσημείωτη είναι η χρήση ημερολογίων (Logging). Η καταγραφή λεπτομερών ημερολογίων συμβάλλει στην προσπάθεια αναγνώρισης παραβιάσεων και ανωμαλιών. Διενεργείται με παρακολούθηση των δικαιωμάτων πρόσβασης και των ενεργειών που διεξάγονται με αυτά.

Αναφορικά με την αντιμετώπιση συμβάντων πρέπει να αναλύονται οι διαδικασίες αντιμετώπισης και ανάκτησης από πιθανές επιθέσεις και ανωμαλίες. Βασικό εργαλείο αποτελεί η συνεχής εκπαίδευση και εποπτεία που πετυχαίνετε μέσα από [8]:

- **Την εκπαίδευση του προσωπικού**: Η ανάγκη για συνεχή εκπαίδευση του προσωπικού σχετικά με τις διαδικασίες παρακολούθησης και αντίδρασης.
- **Την Εποπτεία και τις Αυτοματοποιημένες Διαδικασίες**: Η αξιοποίηση αυτοματοποιημένων διαδικασιών συμβάλλει στην αύξηση της αποτελεσματικότητας της παρακολούθησης.

Επίσης σε σχέση με τα προληπτικά μέτρα θα πρέπει να εξετάζονται οι καλές πρακτικές για την πρόληψη προβλημάτων ασφάλειας μέσω της συστηματικής παρακολούθησης και ανάλυσης. Η εφαρμογή καλών πρακτικών αναφέρεται στη [8]:

- **Συστηματική Αναθεώρηση**: Η σημασία της συστηματικής αναθεώρησης των διαδικασιών αυτών για την εξασφάλιση της αποτελεσματικότητάς τους.
- **Συνεχής Βελτίωση**: Η ανάγκη για συνεχή βελτίωση των διαδικασιών παρακολούθησης και αντίδρασης με βάση την ανάλυση εμπειριών και την αντιμετώπιση νέων απειλών.

Κεφάλαιο 3ο: Συγκριτική Αξιολόγηση των Καταγεγραμμένων Ευπαθειών για τα πιο Διαδεδομένα RDBMS

3.1 Επισκόπηση των Δημοφιλέστερων RDBMS

Στη βιβλιογραφία είναι καταγεγραμμένες γνωστές ευπάθειες που παρουσιάζονται σε συστημάτων διαχείρισης βάσεων δεδομένων (DBMS). Τα πιο δημοφιλή RDBMS περιλαμβάνουν την Oracle, τη MySQL, την PostgreSQL, και τη Microsoft SQL Server καθώς επίσης και το MongoDB που όμως είναι NoSQL, με την Oracle να είναι το πιο δημοφιλές σύστημα. Σε σχέση με ευπάθειες που εμφανίζουν η MySQL έχει τον υψηλότερο αριθμό ανακαλυφθεισών ευπαθειών, αλλά η τελευταία καταγραφή ήταν το 2015. Τα NoSQL DBMS όπως το MongoDB και το Redis έχουν συχνά συζητηθεί για το χαμηλό επίπεδο ασφάλειας τους. Δεν υπάρχει DBMS σύστημα που να είναι ασφαλές κατά το σχεδιασμό, καθώς η απόλυτη ασφάλεια είναι αδύνατη, όμως η συνεχής βελτίωση τους είναι κρίσιμη [7].

3.2 Ανάλυση Ευπαθειών ανά RDBMS

Τα πιο διαδεδομένα Συστήματα Διαχείρισης Σχεσιακών Βάσεων Δεδομένων (Relational Database Management Systems - RDBMS) περιλαμβάνουν τα παρακάτω συστήματα [7]:

- **MySQL**
- **PostgreSQL**
- **Microsoft SQL Server**
- **Oracle Database**
- **IBM Db2**

Η αναλυτική αξιολόγηση των καταγεγραμμένων ευπαθειών για τις πλατφόρμες RDBMS δείχνει κάποια κοινά χαρακτηριστικά και κάποιες διαφοροποιήσεις ανάμεσα στις πιο διαδεδομένες πλατφόρμες [7] [16]:

1. **MySQL**: Έχει τον υψηλότερο αριθμό ανακαλυφθέντων ευπαθειών, αλλά δεν παρέχονται πλέον δεδομένα, καθώς η τελευταία αναφορά για τις ευπάθειες καταγράφηκε το 2015. Χρησιμοποιείται συχνά σε διαδικτυακές εφαρμογές ανοιχτού κώδικα, καθιστώντας το δημοφιλή στόχο. Δεν διαθέτει ορισμένα προηγμένα χαρακτηριστικά ασφαλείας που διαθέτουν άλλες RDBMS, αποτελώντας ευκολότερο στόχο. Πολλές εφαρμογές χρησιμοποιούν παλαιότερες εκδόσεις του MySQL που περιέχουν γνωστές ευπάθειες συνεπώς η τακτική ενημέρωση και αναβάθμιση είναι κρίσιμη. Συχνά στοχεύετε από SQLi επιθέσεις και ευπάθειες στην αυθεντικοποίηση, λόγω της ευρείας χρήσης του MySQL σε web εφαρμογές. Ενδείκνυται η χρήση προετοιμασμένων δηλώσεων ή παραμετροποιημένων ερωτημάτων (prepared statements) για την αποτροπή SQLi επιθέσεων. Οι ευπάθειες στην αυθεντικοποίηση είναι συχνή και οφείλεται στη χρήση αδύναμων κωδικών πρόσβασης και την έλλειψη πολύπλοκων πολιτικών κωδικών πρόσβασης. Για την αντιμετώπιση της προτείνεται η χρήση πολιτικών ισχυρών κωδικών πρόσβασης και τακτικές αλλαγές των κωδικών. Οι συνέπειες από SQLi επιθέσεις είναι οι εισβολείς να διαβάσουν ευαίσθητα δεδομένα, να αλλάξουν το περιεχόμενο της βάσης δεδομένων και ακόμη και να υπονομεύσουν τον υποκείμενο διακομιστή.
2. **PostgreSQL**: Έχει πολλές ευπάθειες, αλλά φαίνεται να έχει χαμηλό αριθμό περιπτώσεων που ήταν δυνατή η σύνδεση κατά τη διάρκεια των επιθέσεων παρείσδυσης. Αξιοσημείωτη για την ασφάλειά της, αλλά υπάρχουν περιπτώσεις privilege escalation. Η χρήση των ελάχιστων απαιτούμενων προνομίων είναι κρίσιμη για τις προσπάθειες απόκτησης περισσότερων

προνομίων. Ακολουθεί μαζί με την DB2 της IBM τη MySQL σε αριθμό ευπαθειών. Το SQLi λιγότερο συχνό λόγω των αυστηρών τύπων δεδομένων που χρησιμοποιεί, όμως δεν είναι εντελώς ανθεκτική. Η χρήση σωστής επικύρωσης εισόδου είναι απαραίτητη. Επίσης προτείνεται η τακτική αναθεώρηση των ρυθμίσεων ασφαλείας για να μην είναι ευάλωτες σε επιθέσεις. Οι διαδικαστικές γλωσσικές της ικανότητες (PL/pgSQL) μπορούν να αξιοποιηθούν για επιθέσεις εάν δεν διαχειρίζονται σωστά. Η εκμετάλλευση των SQLi επιθέσεων μπορεί να οδηγήσει σε διαρροές δεδομένων, μη εξουσιοδοτημένες αλλαγές και ενδεχομένως εκτέλεση κώδικα μέσω συναρτήσεων.

3. **Microsoft SQL Server**: Έχει το υψηλότερο επίπεδο κινδύνου από τις καταγεγραμμένες ευπάθειες. Η ευρεία χρήση του σε πλήθος εφαρμογών το καθιστά στόχο για εκμεταλλεύσεις (exploits). Είναι σημαντική η τακτική ενημέρωση με τα τελευταία patches ασφαλείας. Αποτελεί στόχο επιθέσεων DoS και ευπαθειών στην αυθεντικοποίηση. Για αντιμετώπιση Brute Force επιθέσεων πρέπει να επιδιώκονται αυστηρά μέτρα ασφαλείας (για παράδειγμα πολύπλοκοι κωδικοί πρόσβασης και εφαρμογή πολιτικών κλειδώματος λογαριασμού μετά από πολλαπλές αποτυχημένες προσπάθειες). Επιρρεπείς σε κλασικές επιθέσεις SQLi, όπως αυτές που εκμεταλλεύονται μη επικυρωμένη είσοδο σε εφαρμογές web. Η χρήση παραμετροποιημένων ερωτημάτων και αποθηκευμένων διαδικασιών βοηθά στην αποτροπή SQLi επιθέσεων. Τα stored procedures όμως εκτός από το να προστατεύσουν μπορεί να αποτελέσουν και το αδύναμο σημείο που θα γίνει επίθεση μέσω αυτών. Οι συνέπειες από SQLi επιθέσεις περιλαμβάνουν τη δυνατότητα εξαγωγής δεδομένων, τη μη εξουσιοδοτημένη τροποποίηση δεδομένων και την εκτέλεσης διαχειριστικών εντολών. Επίσης η δυνατότητα xp_cmdshell του SQL Server μπορεί να αξιοποιηθεί για την εκτέλεση εντολών σε επίπεδο συστήματος.

4. **Oracle Database**: Είναι η πιο δημοφιλής, πολύ ασφαλής, αλλά περιέχει σύνθετες ευπάθειες που σχετίζονται με τον χειρισμό των δεδομένων. Είχε πολλές αναφορές ευπαθειών, κυρίως σχετικά με DoS. Η Oracle εκδίδει τακτικά ενημερώσεις ασφαλείας που πρέπει να εφαρμόζονται άμεσα. Η διαχείριση των δικαιωμάτων χρήστη μπορεί να είναι περίπλοκη και συχνά δεν χρησιμοποιείται σωστά, οδηγώντας σε ενδεχόμενες παραβιάσεις ασφαλείας. Προτείνεται η τακτική αναθεώρηση και περιορισμός των δικαιωμάτων. Εκτιμάται ιδιαίτερα για χαρακτηριστικά ασφαλείας, αλλά εξακολουθεί να είναι ευάλωτο στις SQL Injection επιθέσεις μέσω κακογραμμένων εφαρμογών. Η σύνθετη αρχιτεκτονική παρέχει περισσότερες δυνατότητες για πιθανά σημεία επιθέσεων SQLi. Η χρήση δυναμικής SQL εντός PL/SQL μπορεί να εισάγει τρωτά σημεία. Οι επιπτώσεις των SQLi επιθέσεων περιλαμβάνουν την εκμετάλλευση αδυναμιών που μπορεί να οδηγήσει σε παραβιάσεις δεδομένων, κλιμάκωση προνομίων και μη εξουσιοδοτημένη πρόσβαση στη βάση δεδομένων.

5. **IBM Db2**: Ακολουθεί τη PostgreSQL και τη MySQL σε αριθμό ανακαλυφθέντων ευπαθειών. Γνωστή για ισχυρά χαρακτηριστικά ασφαλείας, αλλά εξακολουθεί να είναι ευάλωτη λόγω ελαττωμάτων στο επίπεδο εφαρμογής. Η χρήση δυναμικών SQL σε αποθηκευμένες διαδικασίες μπορεί να εισάγει τρωτά σημεία. Στις επιπτώσεις επιθέσεων SQLi περιλαμβάνονται οι επιθέσεις SQLi και μπορεί να οδηγήσει σε παραβίαση δεδομένων και μη εξουσιοδοτημένη χειραγώγηση δεδομένων. Επίσης είναι πιθανή η εκμετάλλευση για επιθέσεις χαρακτηριστικών και λειτουργιών που συναντώνται στη DB2.

Η παραπάνω αναλυτική αξιολόγηση των καταγεγραμμένων ευπαθειών δείχνει ότι όλες οι πλατφόρμες RDBMS αντιμετωπίζουν κοινές προκλήσεις ασφάλειας, όπως οι επιθέσεις SQLi, οι αδύναμοι κωδικοί πρόσβασης και οι κακές πρακτικές διαμόρφωσης. Διαφοροποιήσεις ανά πλατφόρμα ωστόσο υπάρχουν, καθώς η συχνότητα και η σοβαρότητα αυτών των προβλημάτων μπορεί να διαφέρουν ανάλογα με την πλατφόρμα και τη χρήση της. Απαραίτητη η τακτική ενημέρωση των συστημάτων, η χρήση ισχυρών μέτρων ασφάλειας και η συνεχής παρακολούθηση και βελτίωση των πρακτικών ασφάλειας, για την προστασία των βάσεων δεδομένων από επιθέσεις.

Για της NoSQL DBMS αντίστοιχα τα προβλήματα ασφαλείας [7]:

- **MongoDB**: Συζητείται ευρέως για το χαμηλό επίπεδο ασφάλειας και συχνά αποτελεί αντικείμενο "εκπαίδευσης" μηχανών αναζήτησης IoT. Είναι επιρρεπές σε DoS, εκτέλεση κώδικα και αποκάλυψη δεδομένων. Συχνά αντικείμενο επιθέσεων λόγω του χαμηλού επιπέδου ασφάλειας.
- **Redis**: Περίπου το 30% των βάσεων δεδομένων ήταν εκτεθειμένες σε επιθέσεις.

3.3 Μηχανισμοί προστασίας

Είναι σημαντικό σε κάθε εφαρμογή να αναγνωρίζονται ποια είναι τα τρωτά σημεία για τα οποία τρωτά σημεία είναι επιβεβλημένη η χρήση μηχανισμών προστασίας. Τα τρωτά σημεία στα οποία πρέπει να δοθεί η δέουσα προσοχή είναι [10]:

- **Η επικύρωση εισόδου**: Όλα τα συστήματα RDBMS είναι επιρρεπή σε SQLi εάν η είσοδος του χρήστη δεν έχει επικυρωθεί σωστά. Η επικύρωση εισόδου είναι η πρώτη γραμμή άμυνας.
- **Τα παραμετροποιημένα ερωτήματα**: Η χρήση παραμετροποιημένων ερωτημάτων ή προετοιμασμένων δηλώσεων (prepared statements) είναι ζωτικής σημασίας για την πρόληψη επιθέσεων με ένεση. Αυτή η μέθοδος αποδεικνύεται αποτελεσματική για όλα τα συστήματα RDBMS.
- **Οι αποθηκευμένες διαδικασίες**: Η σωστή χρήση των αποθηκευμένων διαδικασιών μπορεί να βοηθήσει στον μετριασμό των κινδύνων, αλλά η δυναμική SQL σε αυτές τις διαδικασίες μπορεί να εξακολουθεί να είναι ευάλωτη.
- **Τα ORM και οι βιβλιοθήκες ασφαλείας**: Η μέθοδος Object relational mapping (ORMs) είναι μία τεχνική που μπορεί να βοηθήσει στην αφαίρεση των άμεσων ερωτημάτων SQL, μειώνοντας τον κίνδυνο SQL Injections. Οι προγραμματιστές έχουν τη δυνατότητα να χειρίζονται αντικείμενα, που αντιστοιχούν σε πίνακες της βάσης δεδομένων, για να δουλεύουν σε ένα επίπεδο αφαίρεσης αντί να χρησιμοποιούν άμεσα ερωτήματα στη βάση δεδομένων. Ωστόσο, πρέπει να εφαρμοστούν σωστά [23].

Ενώ διαφορετικά συστήματα RDBMS έχουν μοναδικά χαρακτηριστικά και μηχανισμούς ασφαλείας, όλα μοιράζονται κοινές ευπάθειες στις επιθέσεις SQLi. Η αποτελεσματικότητα αυτών των επιθέσεων εξαρτάται σε μεγάλο βαθμό από τις εφαρμογές που αλληλοεπιδρούν με αυτές τις βάσεις δεδομένων και αν αυτές οι εφαρμογές είναι ασφαλείς έναντι των ευπαθειών. Η χρησιμοποίηση βέλτιστων καθολικά αναγνωρισμένων πρακτικών όπως η επικύρωση εισόδου, τα παραμετροποιημένα ερωτήματα και τα πρότυπα ασφαλούς κωδικοποίησης είναι κρίσιμη για την άμυνα έναντι των προβλημάτων από SQL injection σε κάθε πλατφόρμα βάσεων δεδομένων.

3.4 Συγκριτική Αξιολόγηση των Καταγεγραμμένων Ευπαθειών για τα πιο Διαδεδομένα RDBMS

Η συγκριτική αξιολόγηση των ευπαθειών για τα πιο διαδεδομένα RDBMS, όπως περιγράφεται στο [16], παρέχει μια ανάλυση των διαφορών και των κοινών σημείων μεταξύ των συστημάτων αυτών. Ακολουθεί μια περίληψη των βασικών ευπαθειών για κάθε σύστημα [16]:

1. **Microsoft SQL Server**
   - **Συνολική Ασφάλεια:**
     - Υψηλή ευπάθεια λόγω ευρείας χρήσης και γνωστών λειτουργιών που μπορούν να εκμεταλλευτούν επιτιθέμενοι.
   - **Ευπάθειες:**
     - xp_cmdshell: Δίνει τη δυνατότητα εκτέλεσης εντολών σε επίπεδο συστήματος.
     - Δυναμική SQL: Εύκολα ευάλωτη σε SQLi αν δεν επικυρωθεί κατάλληλα η είσοδος.
     - Λανθασμένη διαχείριση δικαιωμάτων: Μπορεί να οδηγήσει σε αυξημένη πρόσβαση και διοικητικές εντολές.
     - Παρακολούθηση της δικτυακής κυκλοφορίας (network sniffers) για τον εντοπισμό μη κρυπτογραφημένων SQL queries και δεδομένων που μεταδίδονται μέσω δικτύου.

2. **MySQL**
   - **Συνολική Ασφάλεια:**
     - Ευάλωτο κυρίως λόγω παλαιότερων εκδόσεων και της ευρείας χρήσης του σε ανοικτές πηγές εφαρμογές.
   - **Ευπάθειες:**
     - Παλαιότερες Εκδόσεις: Έχουν πολλές γνωστές ευπάθειες.
     - mysql_real_escape_string(): Μπορεί να μην προστατεύει επαρκώς έναντι όλων των τύπων SQLi.
     - Δυναμική SQL σε αποθηκευμένες διαδικασίες: Μπορεί να χρησιμοποιηθούν κακόβουλα αν δεν επικυρωθεί σωστά η είσοδος.
     - Χρήση network sniffers για την καταγραφή και ανάλυση της κυκλοφορίας μεταξύ των clients και του MySQL server για την ανίχνευση ευπαθειών.

3. **PostgreSQL**
   - **Συνολική Ασφάλεια:**
     - Ισχυρή έμφαση στη συμμόρφωση με τα πρότυπα και την επεκτασιμότητα, αλλά και ευπάθειες μέσω κακής διαχείρισης δυναμικής SQL.
   - **Ευπάθειες:**
     - Λειτουργίες SECURITY DEFINER: Μπορούν να εκτελεστούν με αυξημένα δικαιώματα. Από προεπιλογή, οι συναρτήσεις PostgreSQL ορίζονται ως SECURITY INVOKER. Αυτό σημαίνει ότι εκτελούνται με το User ID και το περιβάλλον ασφαλείας του χρήστη που τους καλεί. Οι SQL εντολές που εκτελούνται από μια τέτοια συνάρτηση εκτελούνται με το User ID και το περιβάλλον ασφαλείας του κατόχου της λειτουργίας, δηλαδή με τα ίδια δικαιώματα σαν να τις είχε εκτελέσει απευθείας ο χρήστης. Ο κίνδυνος που ελλοχεύει ακόμα κι αν ο κώδικας είναι καλογραμμένος είναι η πρόσβαση σε αντικείμενα της βάσης δεδομένων από τη συνάρτηση [24].
     - Δυναμική SQL σε PL/pgSQL: Ευάλωτο σε SQLi αν δεν επικυρωθεί σωστά.
     - Προσαρμοσμένες Λειτουργίες: Μπορεί να περιέχουν ευπάθειες αν δεν επικυρωθεί σωστά η είσοδος.

4. **Oracle Database**

- **Συνολική Ασφάλεια:**
  - Γνωστό για τα ισχυρά χαρακτηριστικά ασφαλείας του, αλλά ευάλωτο σε SQL injection μέσω κακογραμμένων εφαρμογών.
- **Ευπάθειες:**
  - DBMS_ASSERT: Μπορεί να παρακαμφθεί αν δεν χρησιμοποιηθεί σωστά.
  - Δυναμική SQL σε PL/SQL: Μπορεί να οδηγήσει σε SQL injection αν δεν επικυρωθεί σωστά.
  - Πολυπλοκότητα Συστήματος: Δημιουργεί περισσότερα σημεία επίθεσης αν δεν διαμορφωθεί σωστά.
  - Παρακολούθηση της κυκλοφορίας (network sniffers) για την ανίχνευση και την καταγραφή δεδομένων και εντολών που μεταδίδονται μη κρυπτογραφημένα.

5. **IBM DB2**

- **Συνολική Ασφάλεια:**
  - Αν και γνωστό για τα ισχυρά χαρακτηριστικά ασφαλείας του, εξακολουθεί να είναι ευάλωτο μέσω δυναμικού SQL και κακής διαχείρισης ρόλων.
- **Ευπάθειες:**
  - Δυναμική SQL: Ευάλωτη σε SQLi αν δεν επικυρωθεί σωστά.
  - Ρόλοι και Δικαιώματα: Λανθασμένη διαχείριση μπορεί να επιτρέψει αυξημένη πρόσβαση.
  - Περιορισμένη Χρήση Στατικού SQL: Αν και ασφαλέστερο, οι προγραμματιστές συχνά επιλέγουν δυναμικό SQL, αυξάνοντας τον κίνδυνο.

Πίνακας 3.1: Βασικές διαφορές και τρωτά σημεία σε δημοφιλή RDBMS [16]

Βασικές διαφορές και τρωτά σημεία στα συστήματα RDBMS		
MySQL		
Τρωτά σημεία	***Επιπτώσεις***	***Βέλτιστες πρακτικές***
Χρησιμοποιείται συνήθως σε web εφαρμογές ανοιχτού κώδικα, καθιστώντας το δημοφιλή στόχο.	Οι εισβολείς μπορούν να εκμεταλλευτούν την επίθεση SQL injection για να διαβάσουν ευαίσθητα δεδομένα, να αλλάξουν το περιεχόμενο της βάσης δεδομένων, και να θέσουν σε κίνδυνο ακόμα και τον υποκείμενο διακομιστή.	Χρήση του mysql_real_escape_string() η οποία κάνει escape (δηλαδή βάζει το \) σε ειδικούς χαρακτήρες σε αλφαριθμητικά όταν χρησιμοποιηθεί σε παλαιότερες εκδόσεις του MySQL. Να σημειωθεί όμως πως μπορεί να μην προστατεύει επαρκώς έναντι όλων των τύπων SQLi.
Δεν διαθέτει ορισμένα προηγμένα χαρακτηριστικά ασφαλείας που βρίσκονται σε άλλα RDBMS, επιφέροντας μεγαλύτερη ευαισθησία σε επιθέσεις.	Είναι δυνατόν να συμβούν τεχνικές επιθέσεων όπως injections που βασίζονται σε Union-based SQL Injections και Boolean-based Blind Injections.	Αλλαγή σε InnoDB. Είναι προτιμότερη η χρήση του μηχανισμού αποθήκευσης InnoDB για τη καλύτερη διαχείριση συναλλαγών και την ακεραιότητα των δεδομένων.
Οι παλαιότερες εκδόσεις είναι ιδιαίτερα επιρρεπείς σε διάφορους τύπους επιθέσεων SQL Injection λόγω ελλείψεων σε λειτουργίες ασφαλείας και επικύρωσης εισόδου.		Χρήση του Strict SQL Mode. Η ενεργοποίηση της λειτουργίας επιβάλλει αυστηρότερη επικύρωση εισόδου και διαχείριση σφαλμάτων.

PostgreSQL		
Τρωτά σημεία	***Επιπτώσεις***	***Βέλτιστες πρακτικές***
Γνωστό για την έντονη έμφαση που δίνει στη συμμόρφωση με τα πρότυπα και την επεκτασιμότητα.	Η εκμετάλλευση των SQL Injections μπορεί να οδηγήσει σε διαρροές δεδομένων, μη εξουσιοδοτημένες αλλαγές και ενδεχομένως εκτέλεση κώδικα μέσω συναρτήσεων που καθορίζονται από τον χρήστη.	Χρήση του pg_escape_string() ή παρόμοιων συναρτήσεων για να γίνουν escape ειδικοί χαρακτήρες στην είσοδο χρήστη.
Ευάλωτο σε επιθέσεις SQLi εάν οι εφαρμογές δεν προστατεύονται κατάλληλα έναντι αυτών με χρήση σωστής επικύρωσης εισόδου.		Προσοχή στις λειτουργίες SECURITY DEFINER, κατά τον Ορισμό Ασφαλείας, καθώς μπορούν να εκτελεστούν με αυξημένα δικαιώματα.
Η διαδικαστική γλώσσα που υποστηρίζει η PL/pgSQL (Procedural Language/PostgreSQL) μπορεί να χρησιμοποιηθεί κακόβουλα.		Έλεγχος Πρόσβασης Με Ρόλους, οι οποίοι θα πρέπει να αξιοποιηθούν για πιο ισχυρές δυνατότητες ελέγχου πρόσβασης με ρόλους του PostgreSQL για να περιοριστεί η πρόσβαση κατάλληλα.

Microsoft SQL Server		
Τρωτά σημεία	***Επιπτώσεις***	***Βέλτιστες πρακτικές***
Υψηλή ευπάθεια λόγω ευρείας χρήσης σε εταιρικά περιβάλλοντα και γνωστών λειτουργιών που μπορούν να εκμεταλλευτούν επιτιθέμενοι.	Δίνεται η δυνατότητα εξαγωγής δεδομένων, η μη εξουσιοδοτημένη τροποποίηση δεδομένων και η εκτέλεση διαχειριστικών εντολών.	Να αποφεύγεται η χρήση του xp_cmdshell και να απενεργοποιηθεί το κέλυφος xp_cmdshell αν δεν είναι απαραίτητο.
Επιρρεπείς σε κλασικές επιθέσεις SQLi, όπως αυτές που εκμεταλλεύονται μη επικυρωμένη είσοδο σε εφαρμογές Web.	Η δυνατότητα xp_cmdshell του SQL Server μπορεί να αξιοποιηθεί για την εκτέλεση εντολών σε επίπεδο συστήματος.	Να γίνεται χρήση της συνάρτησης QUOTENAME() για να τοποθετούνται με ασφάλεια τα αναγνωριστικά σε εισαγωγικά.
Οι προηγμένες λειτουργίες, όπως οι stored procedures, μπορούν να είναι τόσο ένας μέσο/μηχανισμός άμυνας όσο και ένα μέσο επίθεσης εάν δεν ασφαλιστούν σωστά.		Κρυπτογράφηση Δεδομένων χρησιμοποιώντας τις δυνατότητες κρυπτογράφησης του SQL Server για την προστασία ευαίσθητων δεδομένων.
Λανθασμένη διαχείριση δικαιωμάτων μπορεί να οδηγήσει σε πρόσβαση με αυξημένα δικαιώματα.		

Oracle DB		
Τρωτά σημεία	***Επιπτώσεις***	***Βέλτιστες πρακτικές***
Εκτιμάται ιδιαίτερα για χαρακτηριστικά ασφαλείας, αλλά εξακολουθεί να είναι ευάλωτο σε SQLi μέσω κακογραμμένων εφαρμογών.	Η εκμετάλλευση μπορεί να οδηγήσει σε παραβιάσεις δεδομένων, κλιμάκωση προνομίων και μη εξουσιοδοτημένη πρόσβαση στη βάση δεδομένων.	Χρήση του πακέτου DBMS_ASSERT για την επικύρωση και τον καθαρισμό εισόδου σε δυναμικό SQL.
Η σύνθετη αρχιτεκτονική παρέχει περισσότερες δυνατότητες για πιθανά ευάλωτα σημεία για SQLi. Η πολυπλοκότητα των Βάσεων Δεδομένων της Oracle μπορεί να δημιουργήσει περισσότερα σημεία επίθεσης, ειδικά αν δεν διαμορφωθεί σωστά.	Οι επιθέσεις μπορούν να χειραγωγήσουν συγκεκριμένες λειτουργίες και πακέτα της Oracle για να διευρύνουν την απήχησή τους.	Χρήση Εικονικής Ιδιωτικής Βάσης Δεδομένων (VPD) της Oracle για τη δημιουργία πολιτικών ασφαλείας που επιβάλλουν λεπτομερή έλεγχο πρόσβασης.
Η χρήση δυναμικής SQL εντός PL/SQL μπορεί να εισάγει τρωτά σημεία αν η είσοδος δεν επικυρωθεί επαρκώς.		Εφαρμογή Απόκρυψης Δεδομένων Oracle για την κάλυψη ευαίσθητων δεδομένων στα αποτελέσματα της εφαρμογής.

IBM DB2		
Τρωτά σημεία	***Επιπτώσεις***	***Βέλτιστες πρακτικές***
Γνωστό για ισχυρά χαρακτηριστικά ασφαλείας, αλλά εξακολουθεί να είναι ευάλωτο λόγω ελαττωμάτων στο επίπεδο εφαρμογής.	Το SQL Injection ενδέχεται να έχει ως αποτέλεσμα τη παραβίαση δεδομένων και τη μη εξουσιοδοτημένη χειραγώγηση δεδομένων.	Χρήση Στατικού SQL αντί για δυναμική SQL για τη μείωση του κίνδυνου από SQLi.
Η χρήση δυναμικών SQL ερωτημάτων σε αποθηκευμένες διαδικασίες μπορεί να εισάγει τρωτά σημεία αν δεν επικυρωθεί σωστά η είσοδος.	Πιθανή εκμετάλλευση χαρακτηριστικών και λειτουργιών που έχει ειδικά η DB2.	Χρήση αξιόπιστων πλαισίων για την παροχή πιο λεπτομερών ελέγχων των συνδέσεων και της αυθεντικοποίησης της βάσης δεδομένων.
Λανθασμένη διαχείριση ρόλων και δικαιωμάτων μπορεί να επιτρέψει την πρόσβαση σε δεδομένα ή τη δυνατότητα να εκτελέσουν εντολές μη εξουσιοδοτημένοι χρήστες.		Ενεργοποίηση και ρύθμιση του ελέγχου ασφαλείας για την παρακολούθηση και καταγραφή των δραστηριοτήτων της βάσης δεδομένων.
Αν και το στατικό SQL είναι ασφαλέστερο, πολλοί προγραμματιστές επιλέγουν χρήση δυναμικής SQL λόγω ευελιξίας, αυξάνοντας έτσι τον κίνδυνο SQLi.		

Από τη συγκριτική αξιολόγηση των παραπάνω προκύπτουν τα εξής συμπεράσματα:

- **Κοινά Σημεία Ευπάθειας:**
    - Όλα τα συστήματα είναι ευάλωτα σε SQLi κυρίως λόγω της υποστήριξης της δυναμικής SQL άρα και δυνατής χρήσης της από όλα τα συστήματα, η οποία ενδέχεται να εκτελείται και χωρίς κατάλληλη επικύρωση εισόδου αποτελώντας βασικό παράγοντα ευπάθειας. Όταν τα δεδομένα εισόδου ενός χρήστη ενσωματώνονται σε δυναμικά SQL ερωτήματα χωρίς να ελέγχονται ή να επικυρώνονται, οι εισβολείς μπορούν να εκμεταλλευτούν αυτήν την αδυναμία και να εκτελέσουν κακόβουλες εντολές SQL. Μέσω αυτών των εντολών μπορούν να διαβάσουν, να τροποποιήσουν ή να διαγράψουν δεδομένα της βάσης δεδομένων.
    - Η διαχείριση δικαιωμάτων και η σωστή διαμόρφωση ρόλων στους χρήστες είναι κρίσιμη για την αποφυγή πρόσβασης με αυξημένα δικαιώματα που δίνει τη δυνατότητα εκτέλεσης κακόβουλων εντολών. Όταν τα δικαιώματα δεν είναι σωστά διαμορφωμένα ή και οι ρόλοι αντίστοιχα, υπάρχει αυξημένος κίνδυνος παραβίασης ασφάλειας μέσω τεχνικών privilege escalation, μέσω των οποίων μπορεί ένας χρήστης με χαμηλά δικαιώματα να αποκτήσει πρόσβαση σε ευαίσθητες περιοχές του συστήματος. Η σωστή διαχείριση δικαιωμάτων αποτρέπει τέτοιες επιθέσεις, περιορίζοντας την πρόσβαση. Με την εφαρμογή τέτοιων πρακτικών, διασφαλίζεται ότι οι χρήστες έχουν την απαραίτητη

μόνο πρόσβαση για τις ανάγκες τους, μειώνοντας σημαντικά τις πιθανότητες κακόβουλης εκμετάλλευσης των συστημάτων και προστατεύοντας τα δεδομένα και τις υποδομές από μη εξουσιοδοτημένη πρόσβαση και επιθέσεις.

- **Ειδικά Σημεία Ευπάθειας:**
    - Όπως φαίνεται και στον πίνακα 3.1 κάθε σύστημα έχει μοναδικά χαρακτηριστικά και δυνατότητες τα οποία μπορούν να αποτελέσουν πηγή ευπάθειας αν δεν ακολουθηθούν οι βέλτιστες πρακτικές.
    - Οι αποθηκευμένες διαδικασίες και οι προσαρμοσμένες λειτουργίες σε κάθε RDBMS μπορούν να αποτελέσουν σημεία εκμετάλλευσης αν δεν εφαρμοστούν σωστές πρακτικές ασφαλείας.

Η εφαρμογή των βέλτιστων πρακτικών ασφαλείας, η τακτική επικύρωση και αναθεώρηση του κώδικα, η κατάλληλη διαμόρφωση και διαχείριση των δικαιωμάτων χρηστών καθώς και η επικύρωση εισόδου, είναι απαραίτητες για την ενίσχυση της ασφάλειας σε κάθε RDBMS.

Κεφάλαιο 4ο: Επιθέσεις παρείσδυσης (Penetration Testing - PenTests)

4.1 Εισαγωγή

Η μεγάλη και διαρκώς αυξανόμενη ζήτηση για εφαρμογές λογισμικού έχει οδηγήσει στη ανάγκη διασφάλισης της ποιότητας του λογισμικού που αναπτύσσεται. Για τη διασφάλιση της ποιότητας του λογισμικού κρίνεται αναγκαία η δοκιμή του λογισμικού. Ως δοκιμή λογισμικού ορίζεται η διαδικασία εκτέλεσης μιας εφαρμογής για την εύρεση σφαλμάτων του λογισμικού (όπως λάθη ή άλλα ελαττώματα) που μπορεί να περιέχει. Συνιστά ένα θεμελιώδες, αν όχι το πιο σημαντικό, στάδιο του κύκλου ζωής ανάπτυξης λογισμικού. Μια δοκιμή μπορεί να αναλύσει κάθε στοιχείο του λογισμικού για να εντοπίσει τη διαφορά μεταξύ των πραγματικών και των προβλεπόμενων τιμών που λαμβάνουν συνθήκες κατά την εκτέλεση του λογισμικού και να αξιολογήσει το λογισμικό. Η διαδικασία δοκιμής ενός λογισμικού οδηγεί στην ελαχιστοποίηση των σφαλμάτων του και συνεπακόλουθα στη μείωση του κόστους του. Για το σκοπό αυτό, ακολουθούνται διάφορες τεχνικές και στρατηγικές δοκιμών λογισμικού για σκοπούς καλύτερης διασφάλισης ποιότητας [25].

Η διαδικασία καταγραφής και αξιολόγησης των προβλημάτων ασφάλειας των Συστημάτων Βάσεων Δεδομένων γίνεται μέσω επιθέσεων παρείσδυσης δικτύων.

Μια επίθεση παρείσδυσης δικτύου είναι ένας τύπος δοκιμής που στοχεύει σε ολόκληρο το δίκτυο υπολογιστών μιας εταιρείας μέσω της πρακτικής της ηθικής εισβολής .

Ο στόχος μιας επίθεσης παρείσδυσης δικτύου είναι να αποκαλύψει και να εντοπίσει τυχόν τρωτά σημεία εντός του οργανισμού. Αυτό περιλαμβάνει τη διεξοδική αξιολόγηση των μέτρων ασφαλείας του δικτύου μέσω τόσο εσωτερικών δοκιμών όσο και εξωτερικών δοκιμών, όπως δοκιμές εφαρμογών ιστού και ψευδείς επιθέσεις phishing [26].

4.2 Έλεγχος του κώδικα (Code Audits)

Ο έλεγχος του κώδικα ενός λογισμικού είναι μια ολοκληρωμένη ανάλυση του πηγαίου κώδικα με σκοπό την ανακάλυψη σφαλμάτων και παραβιάσεων ασφαλείας. Επιχειρεί να μειώσει τα σφάλματα πριν κυκλοφορήσει το λογισμικό. Να σημειωθεί πως η διαδικασία αναθεώρησης του κώδικα (code review) είναι διαφορετική διαδικασία από τον έλεγχο του κώδικα (code audit). Η αναθεώρηση κώδικα αποτελεί έναν γρήγορο έλεγχο. Εστιάζει σε ένα μικρότερο μέρος του κώδικα και συμβαίνει πιο συχνά [27].

Ο έλεγχος κώδικα είναι μια πιο σύνθετη διαδικασία. Εξετάζει ένα μεγαλύτερο μέρος του κώδικα, και συμβαίνει λιγότερο συχνά, ίσως ετησίως ή πριν από μια μεγάλη κυκλοφορία. Οι έλεγχοι είναι πιο διεξοδικοί, ελέγχοντας για τρωτά σημεία ασφαλείας, συμμόρφωση με κανονισμούς, ζητήματα απόδοσης και πόσο εύκολο είναι να διατηρηθεί ο κώδικας μακροπρόθεσμα. Ο έλεγχος καταλήγει σε μια έκθεση με μια λίστα προβλημάτων και συστάσεων που πρέπει να αντιμετωπιστούν [28].

Οι επιθέσεις παρείσδυσης εφαρμογών περιλαμβάνονται στις πιο συνηθισμένες τεχνικές ελέγχου κώδικα. Προσπαθούν να εντοπίσουν τρωτά σημεία στο λογισμικό, ξεκινώντας όσο το δυνατόν περισσότερες γνωστές τεχνικές επίθεσης σε πιθανά σημεία πρόσβασης που επιχειρούν να παραβιάσουν την ασφάλεια της εφαρμογής [27].

Η διαδικασία ελέγχου κώδικα έχει ως εξής, πρώτα διαβάζεται ο πηγαίος κώδικας του συστήματος. Δεύτερον, ελέγχεται ο βαθμός διασφάλισης της ασφάλειας από τον πηγαίο κώδικα. Ο σκοπός του ελέγχου στον κώδικα είναι να διασφαλίσει ότι οι προγραμματιστές ακολουθούν αυστηρά τα πρότυπα

ασφαλείας. Γενικά επιδιώκεται, μετά τη διαδικασία ελέγχου και μετά από την πλήρη αναθεώρηση του πηγαίου κώδικα, αν μπορούν να αποφευχθούν τα περισσότερα κενά ασφαλείας στην εφαρμογή [27].

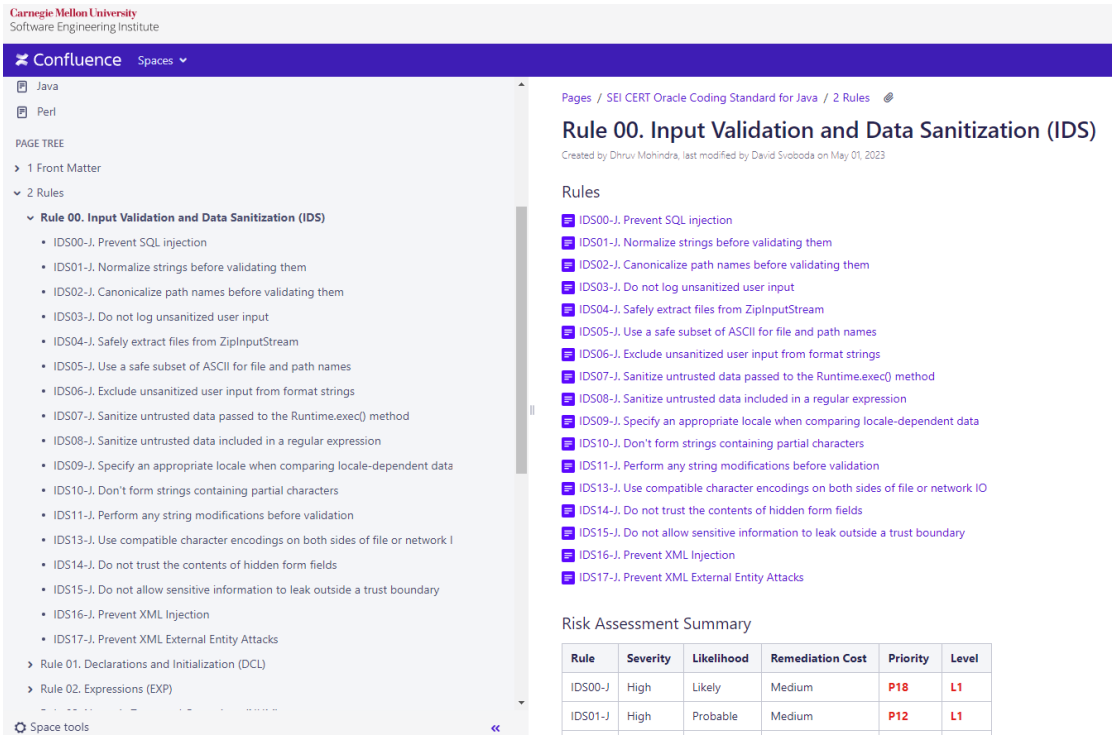
Α. Όχι εμπιστοσύνη στις εισόδους [27].

Σε κάθε πρόγραμμα συνήθως υπάρχουν πολλές είσοδοι. Κάθε είσοδος μπορεί να χρησιμοποιηθεί κακόβουλα και να προκαλέσει ευπάθειες στον κώδικα. Σύμφωνα με στατιστικά στοιχεία, το πρόβλημα ασφαλείας που προκαλείται από τα δεδομένα εισαγωγής σε κενά ασφαλείας είναι πολύ υψηλά (πλησιάζουν το 90%), επομένως η μορφή και το περιεχόμενο της ημερομηνίας εισαγωγής θα πρέπει να είναι αυστηρά περιορισμένη. Είναι επομένως απαραίτητο να γίνεται έλεγχος ασφαλείας στην είσοδο.

Β. Να ακολουθούνται τα πρότυπα ασφαλείας στη συγγραφή κώδικα [27].

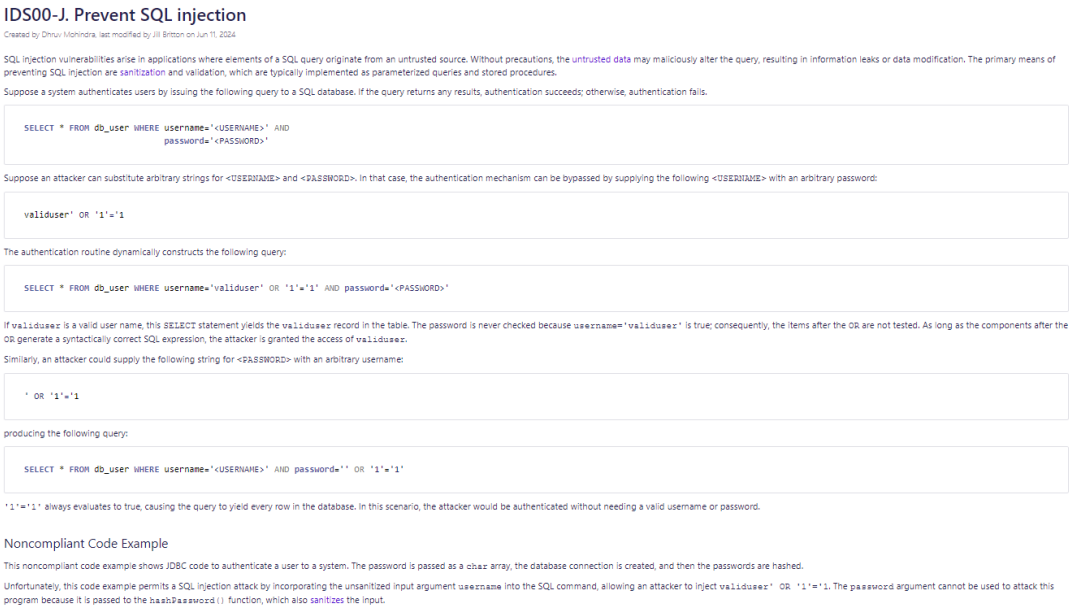
Το Πρόγραμμα CERT, αποτελεί μέρος του Ινστιτούτου Μηχανικής Λογισμικού του Πανεπιστημίου Carnegie Mellon, υιοθετεί μια συνολική στρατηγική για την αναγνώριση και την εξάλειψη των αδυναμιών που μπορεί να περιέχει ένα λογισμικό καθώς και άλλων ελαττωμάτων που μπορεί επίσης να περιέχει. Το CERT έχει συντάξει πολλούς κανόνες κωδικοποίησης ασφαλείας για πολλές γλώσσες υπολογιστών [29] [30].

Στο πρότυπο κωδικοποίησης CERT Oracle για Java, που δημοσιεύτηκε το 2011, που περιλαμβάνει κανόνες και συστάσεις, οι κανόνες παρέχουν κατευθυντήριες γραμμές για τον τρόπο συγγραφής του κώδικα για να αποφευχθούν τυχόν τρωτά σημεία, ενώ οι συστάσεις παρέχουν καθοδήγηση που, όταν τηρούνται, οδηγούν στη βελτίωση της ασφάλειας και της αξιοπιστίας των συστημάτων λογισμικού.



Σχήμα 4.1: SEI CERT Πρότυπο κωδικοποίησης της Oracle για Java - Κανόνας 00 [31]

Ο κανόνος 00 για παράδειγμα αναφέρεται στην Επικύρωση εισόδου και στην εξυγίανση των δεδομένων (Input Validation and Data Sanitization - IDS).



IDS00-J. Prevent SQL injection

Created by Dhruv Mohindra, last modified by Jill Britton on Jun 11, 2024

SQL injection vulnerabilities arise in applications where elements of a SQL query originate from an untrusted source. Without precautions, the untrusted data may maliciously alter the query, resulting in information leaks or data modification. The primary means of preventing SQL injection are sanitization and validation, which are typically implemented as parameterized queries and stored procedures.

Suppose a system authenticates users by issuing the following query to a SQL database. If the query returns any results, authentication succeeds; otherwise, authentication fails.

```
SELECT * FROM db_user WHERE username='<USERNAME>' AND
                            password='<PASSWORD>'
```

Suppose an attacker can substitute arbitrary strings for <USERNAME> and <PASSWORD>. In that case, the authentication mechanism can be bypassed by supplying the following <USERNAME> with an arbitrary password:

```
validuser' OR '1'='1
```

The authentication routine dynamically constructs the following query:

```
SELECT * FROM db_user WHERE username='validuser' OR '1'='1' AND password='<PASSWORD>'
```

If `validuser` is a valid user name, this `SELECT` statement yields the `validuser` record in the table. The password is never checked because `username='validuser'` is true; consequently, the items after the `OR` are not tested. As long as the components after the `OR` generate a syntactically correct SQL expression, the attacker is granted the access of `validuser`.

Similarly, an attacker could supply the following string for <PASSWORD> with an arbitrary username:

```
' OR '1'='1
```

producing the following query:

```
SELECT * FROM db_user WHERE username='<USERNAME>' AND password='' OR '1'='1'
```

`'1'='1'` always evaluates to true, causing the query to yield every row in the database. In this scenario, the attacker would be authenticated without needing a valid username or password.

Noncompliant Code Example

This noncompliant code example shows JDBC code to authenticate a user to a system. The password is passed as a `char` array, the database connection is created, and then the passwords are hashed.

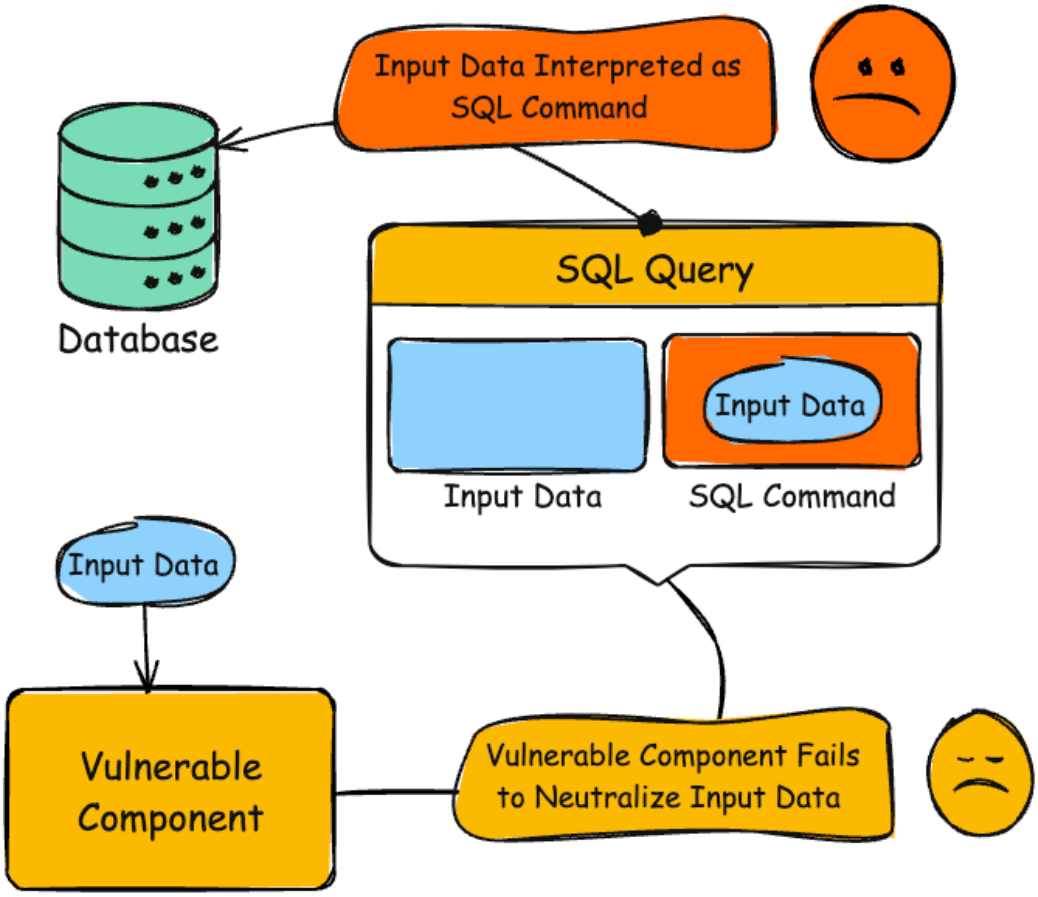
Unfortunately, this code example permits a SQL injection attack by incorporating the unsanitized input argument `username` into the SQL command, allowing an attacker to inject `validuser' OR '1'='1`. The `password` argument cannot be used to attack this program because it is passed to the `hashPassword()` function, which also sanitizes the input.

Σχήμα 4.2: Κανόνας 00 - IDS00-J. Εμποδίζοντας SQL injection [30]

Τα ασφαλή πρότυπα συγγραφής κώδικα παρέχουν μια λεπτομερή απαρίθμηση σφαλμάτων κωδικοποίησης που έχουν προκαλέσει γνωστές ευπάθειες, μαζί με τον τρόπο περιορισμού τους για τις πιο συχνά χρησιμοποιούμενες γλώσσες ανάπτυξης λογισμικού.

Γ. Διαχείριση αδυναμιών από το Αποθετήριο των γνωστών αδυναμιών CWE [27].

Η απαρίθμηση κοινών αδυναμιών (Common Weakness Enumeration - CWE) είναι ένα ενοποιημένο, μετρήσιμο σύνολο αδυναμιών ενός λογισμικού που επιτρέπει την αποτελεσματική συζήτηση, περιγραφή, επιλογή και χρήση εργαλείων και υπηρεσιών ασφάλειας λογισμικού που μπορούν να βρουν αυτές τις αδυναμίες στον πηγαίο κώδικα και τα λειτουργικά συστήματα. Το CWE επιτρέπει επίσης καλύτερη κατανόηση και διαχείριση των αδυναμιών του λογισμικού που έχουν σχέση με την αρχιτεκτονική και το σχεδιασμό. Απαριθμεί σχεδιαστικές και αρχιτεκτονικές αδυναμίες, καθώς και σφάλματα κωδικοποίησης και σχεδιασμού χαμηλού επιπέδου [32].



Σχήμα 4.3: CWE-89: Ακατάλληλη εξουδετέρωση ειδικών στοιχείων που χρησιμοποιούνται σε μια εντολή SQL ('SQL Injection') [32]

Υπάρχει μια σχέση μεταξύ του CWE και του CERT. Μέσω CERT αναπτύσσονται ασφαλή πρότυπα κωδικοποίησης για κοινώς χρησιμοποιούμενες γλώσσες προγραμματισμού όπως η C, C ++ και η Java μέσω μιας ευρείας κοινοτικής προσπάθειας που περιλαμβάνει μέλη των κοινοτήτων ανάπτυξης λογισμικού και ασφάλειας λογισμικού. Τα ασφαλή πρότυπα κωδικοποίησης CWE και CERT εκτελούν ξεχωριστούς αλλά αμοιβαία υποστηρικτικούς ρόλους. Με απλά λόγια, το CWE παρέχει ένα ολοκληρωμένο αποθετήριο γνωστών αδυναμιών, ενώ τα ασφαλή πρότυπα κωδικοποίησης CERT προσδιορίζουν ανασφαλείς κατασκευές κωδικοποίησης που, εάν υπάρχουν στον κώδικα, θα μπορούσαν να εκθέσουν μια αδυναμία ή ευπάθεια στο λογισμικό.

Δεν υπάρχουν όλες οι αδυναμίες που απαριθμούνται στο CWE σε κάποιο συγκεκριμένο ασφαλές πρότυπο κωδικοποίησης, επειδή δεν υπάρχουν όλες οι αδυναμίες σε κάθε γλώσσα και επειδή το CWE περιλαμβάνει επίσης θέματα σχεδιασμού υψηλού επιπέδου. Όλες οι κατευθυντήριες γραμμές για την κωδικοποίηση CERT δεν αντιστοιχίζονται απευθείας στις αδυναμίες του CWE, επειδή ορισμένα σφάλματα κωδικοποίησης μπορεί να εκδηλωθούν με διάφορους τρόπους που δεν συσχετίζονται άμεσα με οποιαδήποτε δεδομένη αδυναμία. Και τα δύο εργαλεία είναι απαραίτητα για την αξιολόγηση της ασφάλειας των συστημάτων λογισμικού. Οι οδηγίες σε ένα πρότυπο ασφαλούς κωδικοποίησης CERT διασταυρώνονται με καταχωρήσεις CWE. Αυτές οι διασταυρούμενες αναφορές δημιουργούνται μόνο για κατευθυντήριες γραμμές οι οποίες, αν παραβιαστούν, συμβάλλουν άμεσα στην αναφερόμενη αδυναμία [32].

4.3 Οι επιθέσεις παρείσδυσης για σάρωση ευπαθειών

Πολλές μεγάλες επιχειρήσεις αρχίζουν να αναθέτουν σε εξειδικευμένες εταιρείες ασφαλείας τον έλεγχο ασφάλειας για αποκάλυψη τυχών τρωτών σημείων στα δίκτυα και στα συστήματα τους. Κι αυτό ως αποτέλεσμα της αυξανόμενης συνειδητοποίησης της σημασίας της ασφάλειας μεταξύ των επιχειρήσεων και της εκθετικής αύξησης του κόστους των παραβιάσεων της ασφάλειας. Οι επιθέσεις παρείσδυσης είναι μία από αυτές τις βασικές υπηρεσίες ασφαλείας.

Η ηθική πειρατεία (ethical hacking), γνωστή και ως "white hat hacking", είναι η πρακτική της προσπάθειας διείσδυσης και εκμετάλλευσης ενός συστήματος για να εντοπιστούν οι αδυναμίες του και να ενισχύσει την ασφάλειά του. Ο τομέας αυτός χωρίζεται κυρίως σε δύο κατηγορίες: πρώτον τις επιθέσεις παρείσδυσης από εταιρείες εξειδικευμένες στην ασφάλεια πληροφοριών και δεύτερον για στρατιωτικές ή μυστικές υπηρεσίες. Οι επιθέσεις παρείσδυσης είναι μια νόμιμη, επιτρεπτή επίθεση σε ένα δίκτυο για την ανακάλυψη ευπαθειών. Οι οργανισμοί συνήθως ξεκινούν εκτελώντας μια αξιολόγηση ευπάθειας για να εντοπίσουν τυχόν αδύναμα σημεία στα λειτουργικά συστήματα, τις υπηρεσίες και το δίκτυό τους. Μετά τη σάρωση για ανακάλυψη ευπαθειών, ο ειδικός προσπαθεί να εκμεταλλευτεί αυτές τις ευπάθειες για να διαπιστωθεί αν είναι πραγματικές και αν χρειάζεται να επενδυθεί χρόνος και χρήμα για την επίλυσή τους [33].

4.4 Πώς λειτουργεί η επίθεση παρείσδυσης δικτύου

Τις επιθέσεις παρείσδυσης δικτύου τις αναλαμβάνουν οι αποκαλούμενοι και ηθικοί χάκερ γιατί για να τις πραγματοποιήσουν χρησιμοποιούν τεχνικές hacking. Οι ηθικοί χάκερ, (ή οι κόκκινες ομάδες όπως είναι γνωστοί) χρησιμοποιούν εργαλεία και τεχνικές hacking για να κάνουν μια ψευδή κυβερνοεπίθεση στο σύστημα υπολογιστών ενός οργανισμού. Ο στόχος ενός εισβολέα είναι να ξεπεράσει το τείχος προστασίας του οργανισμού και να αποκτήσει μη εξουσιοδοτημένη πρόσβαση αναζητώντας ένα ελάττωμα στην υποδομή της επιχείρησης.

Μια επίθεση παρείσδυσης δικτύου μπορεί να περιλαμβάνει επίθεση σε: εφαρμογές Ιστού, APIs (Application Programming Interface), τελικά σημεία και φυσικά στοιχεία ελέγχου. Οι προσομοιωμένες επιθέσεις στο λειτουργικό σύστημα μπορούν να αποκαλύψουν αδυναμίες ασφάλειας και να δείξουν στον οργανισμό πού υπάρχουν αδύναμα σημεία.

Οι ψεύτικες επιθέσεις βοηθούν τις ομάδες ασφαλείας να αποκαλύψουν σχετικές ευπάθειες ασφαλείας στην υποδομή του δικτύου. Κοινές απειλές που δοκιμάζονται είναι κατανεμημένες επιθέσεις άρνησης υπηρεσίας (Distributed Denial-of-Service - DDoS), επιθέσεις σε ένα σύστημα ονομάτων τομέα (Domain Name System - DNS), κακόβουλο λογισμικό, ηλεκτρονικό ψάρεμα και SQL injection.

Οι ομάδες δοκιμών λογισμικού χρησιμοποιούν επίσης εργαλεία για τη διεξαγωγή επανασύνδεσης και αυτοματοποίησης της διαδικασίας επίθεσης παρείσδυσης δικτύου. Συχνά χρησιμοποιούνται δύο τύποι δοκιμών: εσωτερικές και εξωτερικές.

Δοκιμές εσωτερικού δικτύου: Σε μια εσωτερική δοκιμή, οι ελεγκτές ενεργούν ως εσωτερικοί εισβολείς ή ως κάποιος που προσπαθεί να κάνει μια κακόβουλη ενέργεια με κλεμμένα διαπιστευτήρια. Ο κύριος σκοπός μιας τέτοιας δοκιμής είναι να βρεθούν τρωτά σημεία που μπορεί να χρησιμοποιήσει ένα άτομο ή ένας υπάλληλος εντός του οργανισμού. Συνηθέστερα θέματα ασφαλείας είναι η κλοπή πληροφοριών και η κατάχρηση δικαιωμάτων πρόσβασης σε ιδιωτικά ή και ευαίσθητα δεδομένα.

Δοκιμές εξωτερικού δικτύου: Οι υπηρεσίες επίθεσης παρείσδυσης εξωτερικού δικτύου μιμούνται εξωτερικούς εισβολείς που προσπαθούν να εισβάλουν στο δίκτυο. Αυτές οι επιθέσεις παρείσδυσης έχουν ως στόχο την ανακάλυψη θεμάτων ασφαλείας που σχετίζονται με το Διαδίκτυο, και γίνονται σε

διακομιστές, δρομολογητές, ιστοτόπους, εφαρμογές και υπολογιστές εργαζομένων, τα οποία αποτελούν πηγές κινδύνων [26].

4.5 Η διαδικασία επίθεσης παρείσδυσης δικτύου

Συχνά μια επίθεση παρείσδυσης δικτύου ακολουθεί συγκεκριμένα βήματα. Είναι διαφορετικά πολλές φορές αλλά παραπλήσια τα βήματα που προτείνονται από διάφορους συγγραφείς. Η δοκιμή ολοκληρώνεται με μια αναφορά επίθεσης παρείσδυσης δικτύου, η οποία είναι μια λεπτομερής ανάλυση των ευρημάτων κινδύνων ασφαλείας και των συνεπακόλουθων επιχειρηματικών κινδύνων. Τα βήματα μιας επίθεσης παρείσδυσης είναι [26]:

1. **Συλλογή πληροφοριών και σχεδίαση**. Στο αρχικό στάδιο, οι ηθικοί χάκερ συζητούν με τους βασικούς ενδιαφερόμενους για το ποιος θα είναι ο γενικός στόχος της δοκιμής και ποιες ευπάθειες έχει εντοπίσει ο οργανισμός. Πριν από τις επιθέσεις παρείσδυσης θα πρέπει να γίνει αξιολόγηση των ευπαθειών.

   Από εκεί, οι ομάδες που εκτελούν τις επιθέσεις παρείσδυσης και οι ενδιαφερόμενοι αποφασίζουν ποιες επιθέσεις θα εκτελέσουν και τις μετρήσεις επιτυχίας που σχεδιάζουν να χρησιμοποιήσουν. Οι δοκιμαστές χρησιμοποιούν πολλά διαφορετικά εργαλεία και μεθοδολογίες για να εκτελέσουν τις ψεύτικες επιθέσεις, όπως σάρωση θυρών και χαρτογράφηση δικτύου (πχ. με χρήση εργαλείων όπως το nmap).

   Υπάρχουν τρεις τύποι επιθέσεων/δοκιμών παρείσδυσης. Ανάλογα με την οργάνωση, αυτά μπορούν να χρησιμοποιηθούν μεμονωμένα ή συνδυαστικά. Αναλυτικότερα [26]:

   - Μια δοκιμή «μαύρου κουτιού» διεξάγεται από την οπτική γωνία ενός μέσου χάκερ με ελάχιστη ή καθόλου εσωτερική γνώση σχετικά με το δίκτυο που τρέχει το σύστημα. Αυτός ο τύπος δοκιμών θα ήταν μια εξωτερική δοκιμή παρείσδυσης, δεδομένου ότι ο στόχος του είναι να εκμεταλλευτεί τις ευπάθειες που αντιμετωπίζουν προς τα έξω μέσα στο δίκτυο.

   - Η δοκιμή γκρίζου κουτιού αποτελεί τον τύπο δοκιμής διείσδυσης δικτύου που έχει περισσότερο εσωτερική εστίαση και στοχεύει στην απεικόνιση ενός χάκερ με πρόσβαση στο εσωτερικό σύστημα. Διατηρώντας επίσης ορισμένες από τις πτυχές ενός εξωτερικού χάκερ. Η δοκιμή γκρι κουτιού στοχεύει να είναι ένας κακός ηθοποιός σε έναν οργανισμό που μπορεί να έχει αυξημένα προνόμια που χρησιμοποιούνται με κακόβουλο τρόπο.

   - Τέλος, η δοκιμή λευκού κουτιού είναι ο πιο παρεμβατικός από τους τρεις τύπους δοκιμών ασφαλείας. Αυτή η δοκιμή εκτελείται για να απεικονίσει έναν ειδικό πληροφορικής ή κάποιον με πρόσβαση στον πηγαίο κώδικα του οργανισμού και σε όλα τα δεδομένα που ενδεχομένως αφορούν το σύστημα. Αυτή η δοκιμή συνήθως εκτελείται τελευταία για να ελεγχθεί η ακεραιότητα μιας αρχιτεκτονικής IT. Και επιπλέον βεβαιωθείτε ότι οι πιθανοί χάκερ και οι κυβερνοεπιθέσεις στο σύστημα στόχο είναι αδιαπέραστες.

2. **Πραγματοποίηση αναγνώρισης και ανακάλυψη**. Στη φάση αναγνώρισης και ανακάλυψης, οι δοκιμαστές διείσδυσης λαμβάνουν δεδομένα από την αναγνώριση για να εκτελέσουν ζωντανές δοκιμές και να ανακαλύψουν τις υπάρχουσες ευπάθειες μέσω τακτικών, όπως η κοινωνική μηχανική. Χρησιμοποιώντας παραπλανητικά εργαλεία για να χειραγωγήσουν άτομα ώστε να μοιράζονται πληροφορίες, οι δοκιμαστές διείσδυσης ελπίζουν να βρουν πού βρίσκονται τα αδύναμα σημεία και να στοχεύσουν αυτά τα τρωτά σημεία.

   Στο βήμα ανακάλυψης, οι δοκιμαστές διείσδυσης μπορούν να χρησιμοποιήσουν εργαλεία όπως σαρωτής θύρας και σαρωτής ευπάθειας. Ο σαρωτής θυρών προσδιορίζει ανοιχτές θύρες σε ένα

σύστημα όπου ενδέχεται να εισέλθουν χάκερ και ένας σαρωτής ευπάθειας εντοπίζει υπάρχουσες ευπάθειες σε ένα σύστημα.

3. **Εκτέλεση της επίθεσης παρείσδυσης δικτύου**. Αυτό το επόμενο βήμα είναι να τεθεί σε εφαρμογή όλη η προκαταρκτική εργασία που έχει γίνει μέχρι αυτό το σημείο. Σε αυτό το βήμα, η ομάδα των επιτιθέμενων επαγγελματικών που εκτελεί τις επιθέσεις παρείσδυσης δικτύου χρησιμοποιεί εργαλεία που μπορούν να εκμεταλλευτούν αδυναμίες και να επιχειρήσουν να υποκλέψουν δεδομένα. Ο σκοπός είναι να γίνει σαφές το μέγεθος της ζημιάς που μπορούν να προκαλέσουν οι ηθικοί χάκερ καθώς επίσης εάν αποκτήσουν πρόσβαση, να καθοριστεί το χρονικό διάστημα που μπορούν να παραμείνουν εντός του συστήματος.
   Η ομάδα που εκτελεί επιθέσεις παρείσδυσης μπορεί να ξεκινήσει δοκιμάζοντας μία ευπάθεια τη φορά, αλλά πρέπει να εκτελούνται και δοκιμές σε πολλαπλά τρωτά σημεία για να διασφαλιστεί ότι ακολουθείται μια ευρεία προσέγγιση ως προς την αντιμετώπιση αυτών των κινδύνων ασφαλείας.

   Ανάλυση και αναφορά των αποτελεσμάτων. Το τελευταίο βήμα είναι να τεκμηριωθούν ποιες επιθέσεις παρείσδυσης δικτύου πραγματοποιήθηκαν και ακολούθως να μελετηθούν τα αποτελέσματα καθεμιάς από αυτές τις επιθέσεις. Τέλος, πρέπει να συζητηθούν και τα βήματα αποκατάστασης των εντοπισμένων αδυναμιών με την ομάδα ασφάλειας πληροφοριών. Η αναφορά θα πρέπει να περιγράφει λεπτομερώς ολόκληρη τη διαδικασία από την αρχή της μέχρι το τέλος και να προσδιορίζει τα τρωτά σημεία, τα στοιχεία, τα δεδομένα και τις συστάσεις για τον οργανισμό. Αυτή η αναφορά είναι σημαντική για τον ιδιοκτήτη της επιχείρησης να έχει μια πλήρη εικόνα των αδυναμιών που έχουν εντοπιστεί και των κινδύνων που προκύπτουν από αυτές με τη μεγαλύτερη δυνατή ανάλυση τους ώστε να βοηθήσει να λάβει περαιτέρω τεκμηριωμένες αποφάσεις.

4.6 Τα οφέλη των επιθέσεων παρείσδυσης δικτύου εταιρειών

Τα οφέλη που αποκομίζουν οι εταιρείες από τις επιθέσεις παρείσδυσης των δικτύων τους είναι [26]:

1. **Προστασία των δεδομένων**. Ένας οργανισμός αντιμετωπίζει πολλές απειλές και η ύπαρξη προστασίας γύρω από τα δεδομένα του είναι ζωτικής σημασίας για την προστασία της επιχείρησής και των ευαίσθητων πληροφοριών της. Μια δοκιμή διείσδυσης δικτύου εντοπίζει όλα τα τρωτά σημεία και προστατεύει τα δεδομένα του οργανισμού από όλα τα πιθανά σημεία εισόδου.

2. **Κατανόηση των ελέγχων ασφαλείας από το προσωπικό των οργανισμών**. Πραγματοποιώντας επιθέσεις παρείσδυσης, ένας οργανισμός κατανοεί καλύτερα ποια στοιχεία ελέγχου ασφαλείας λειτουργούν και ποια πρέπει να ενισχυθούν. Οι επιθέσεις παρείσδυσης δικτύου δίνει επίσης στον οργανισμό τη δυνατότητα να αναλύσει το επίπεδο ασφαλείας του.

3. **Αποτροπή παραβιάσεων των δεδομένων**. Η προληπτική ανάλυση των τρωτών σημείων ενός δικτύου του οργανισμού διασφαλίζει ότι οι πιθανότητες παραβίασης δεδομένων έχουν σχεδόν εξαλειφθεί. Οι επιθέσεις παρείσδυσης βελτιώνουν τη συνολική ασφάλεια μέσω αξιολογήσεων ασφάλειας και σαρώσεων ασφάλειας στον κυβερνοχώρο.

Κεφάλαιο 5ο: Περιγραφή των Μεθοδολογιών και των Εργαλείων PenTest για RDBMS

5.1 Εισαγωγή

Οι επιχειρήσεις στην εποχή του ψηφιακού κόσμου μετασχηματίζονται και η επιτυχία τους βασίζεται στο ψηφιακό επιχειρείν. Προκειμένου οποιαδήποτε επιχείρηση να αποκτήσει ανταγωνιστικό πλεονέκτημα, η ασφάλεια των δεδομένων και των πληροφοριών αποτελεί σήμερα βασικό μέλημα για τις επιχειρήσεις.

Η τομέας της ασφάλειας των συστημάτων λογισμικού γίνεται περισσότερο σημαντικός σε σχέση με το παρελθόν λόγω της διαρκώς διευρυνόμενης συνδεσιμότητας υπολογιστών που βασίζονται στο Διαδίκτυο, της αυξανόμενης επεκτασιμότητας των συστημάτων και της ανεξέλεγκτης επέκτασης του μεγέθους και της πολυπλοκότητας των συστημάτων.

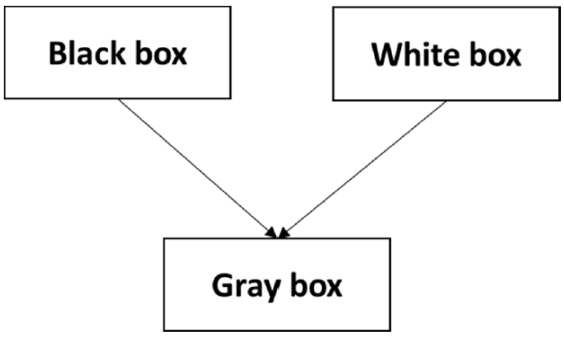
Για την προστασία των πληροφοριών οι επιχειρήσεις είναι επιβεβλημένη η συμμόρφωση τους με τα πρότυπα ασφαλείας. Συνεπώς χρησιμοποιούν τυποποιημένες διαδικασίες λειτουργίας μέσω σχολαστικά τεκμηριωμένων δομημένων μεθόδων.

Προσωπικές πληροφορίες και οικονομικά δεδομένα συναλλαγών πρέπει να προστατευτούν και επιβάλλεται η τήρηση μιας ενδελεχούς, μεθοδικής διαδικασίας που να προσφέρει προστασία έναντι πιθανών κινδύνων.

Η επίθεση παρείσδυσης θεωρείται μια καλά οργανωμένη μέθοδος αξιολόγησης των υπολογιστικών συστημάτων ενός οργανισμού, τα οποία είναι ένας συνδυασμός του υλικού, του λογισμικού αλλά και των ανθρώπων που τα χρησιμοποιούν. Προκειμένου να εντοπιστούν τρωτά σημεία ασφαλείας, απειλές και κίνδυνοι που μπορεί ένας εισβολέας να εκμεταλλευτεί σε εφαρμογές λογισμικού, δίκτυα ή εφαρμογές του Ιστού, αυτή η τεχνική περιλαμβάνει την ανάλυση του υπολογιστικού συστήματος της εταιρείας για την αναζήτηση ελαττωμάτων. Η αναζήτηση είναι πολύ-επίπεδη και κατά τις επιθέσεις παρείσδυσης γίνεται ενεργή εξέταση του συστήματος για τυχόν ευπάθειες. Εξετάζονται σφάλματα σχεδίασης, λανθασμένη ή κατώτερη διαμόρφωση του συστήματος, ελαττώματα στο υλικό και το λογισμικό αλλά και αδυναμίες στις διαδικασίες λειτουργίας.

Η λειτουργική δοκιμή ασφαλείας είναι διαφορετική της δοκιμής/επίθεσης παρείσδυσης. Μια επίθεση παρείσδυσης βοηθά στον προσδιορισμό του βαθμού δυσκολίας που θα είχε ένας εισβολέας να εισβάλει στο δίκτυο υπολογιστών ενός οργανισμού αποκτώντας μη εξουσιοδοτημένη πρόσβαση στα δεδομένα και τα συστήματα πληροφοριών του, σε αντιδιαστολή με μια δοκιμή ασφαλείας η οποία βοηθά στην αξιολόγηση του τρόπου συμπεριφοράς κατά τον έλεγχο ασφαλείας του συστήματος. Σε μια δοκιμή παρείσδυσης, ένας χρήστης χρησιμοποιεί είτε αυτοματοποιημένα εργαλεία είτε ανθρώπινες χειροκίνητες τεχνικές ή και συνδυασμό των δύο τεχνικών για να μιμηθεί μια μη εξουσιοδοτημένη επίθεση στο σύστημα που δοκιμάζεται.

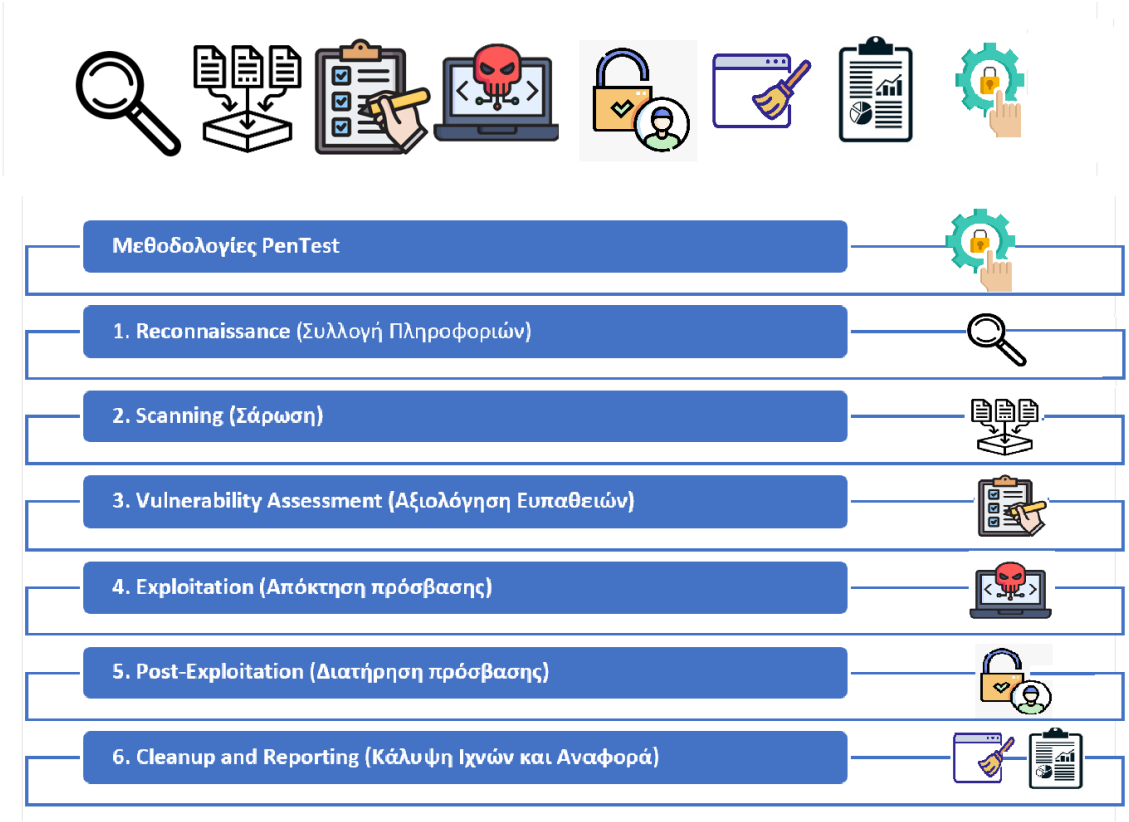
Υπάρχουν διάφοροι τύποι επιθέσεων/δοκιμών παρείσδυσης, οι δοκιμές μαύρο-κουτί, άσπρο-κουτί και γκρι-κουτί, όπου η κατηγοριοποίηση τους γίνεται βάση των διαθέσιμων πληροφορίων. Στη δοκιμή μαύρου-κουτιού, ο ελεγκτής δεν έχει προηγούμενες γνώσεις για τα συστήματα που θα δοκιμαστούν και οι παραβιάσεις της ασφάλειας εντοπίζονται από την αρχή. Στη δοκιμή άσπρο-κουτί, ο ελεγκτής έχει όλες τις πληροφορίες σχετικά με τα συστήματα που ελέγχονται. Στη δοκιμή γκρι-κουτί, ο ελεγκτής έχει μερική γνώση του συστήματος και κάποιες αναγκαίες πληροφορίες πρέπει να τις αναγνωρίσει ο ίδιος [6] [34].



Σχήμα 5.1: Penetration testing τύποι επιθέσεων/δοκιμών [6]

5.2 Μεθοδολογίες Penetration Testing

Η θεματολογία των μεθοδολογιών για Penetration Testing αναλύεται εκτενώς ως κρίσιμο εργαλείο για την ανίχνευση και την εκμετάλλευση ευπαθειών στις βάσεις δεδομένων. Οι επιθέσεις παρείσδυσης αναφέρονται σε διαδικασίες ελέγχου ασφάλειας που στοχεύουν στη δοκιμή της ανθεκτικότητας ενός συστήματος ή εφαρμογής απέναντι σε επιθέσεις από μη εξουσιοδοτημένους χρήστες. Συχνά μια επίθεση παρείσδυσης δικτύου ακολουθεί συγκεκριμένα βήματα. Ακολουθεί ένας συγκερασμός προτάσεων που έχουν διαφορές μεταξύ τους αλλά είναι παραπλήσιες [8] [26] [35] [36]:



Σχήμα 5.2: Τα στάδια Penetration testing

1. **Reconnaissance (Συλλογή Πληροφοριών)**: Στο πρώτο στάδιο, το στάδιο του προγραμματισμού και της προετοιμασίας, οι ηθικοί χάκερ συζητούν με βασικούς ενδιαφερόμενους ποιος θα είναι ο γενικός στόχος της δοκιμής και ποιες ευπάθειες έχει εντοπίσει ο οργανισμός. Πριν από το pen test θα πρέπει να γίνει μια αξιολόγηση ευπάθειας. Από εκεί, η ομάδα των επαγγελματιών που θα εκτελέσει τις επιθέσεις παρείσδυσης και οι ενδιαφερόμενοι αποφασίζουν ποιες δοκιμές θα εκτελέσουν και τις μετρήσεις επιτυχίας που σχεδιάζουν να χρησιμοποιήσουν. Χρησιμοποιούνται πολλά διαφορετικά εργαλεία και μεθοδολογίες (π.χ. το Nmap), για να εκτελεστούν οι ψεύτικες επιθέσεις, όπως σάρωση θυρών, αναγνώριση των ενεργών υπηρεσιών και χαρτογράφηση δικτύου. Επιπρόσθετα γίνεται συλλογή πληροφοριών για το σύστημα στόχο, όπως εκδόσεις λογισμικού, ανοικτές θύρες, και διαμόρφωση των συστημάτων. Χρησιμοποιούνται οι τρεις τύποι δοκιμών: μαύρο-κουτί, άσπρο-κουτί και γκρι-κουτί. Ανάλογα με την περίπτωση, μπορεί να χρησιμοποιηθεί μεμονωμένα κάποιος τύπος ή συνδυαστικά περισσότεροι τύποι δοκιμών.

2. **Scanning (Σάρωση)**: Στο στάδιο της σάρωσης γίνεται εντοπισμός στόχων μέσω της σάρωσης δικτύων για ανοικτές θύρες και υπηρεσίες, και ανίχνευση πιθανών ευπαθειών στο RDBMS με τη χρήση εργαλείων. Εργαλεία όπως το sqlmap μπορούν να βρουν ευπάθειες τύπου SQL Injection, ενώ άλλα εργαλεία μπορούν να εντοπίσουν αδύναμους κωδικούς πρόσβασης ή εσφαλμένες διαμορφώσεις των συστημάτων.

3. **Vulnerability Assessment (Αξιολόγηση Ευπαθειών):** Είναι το στάδιο στο οποίο γίνεται η αναγνώριση αρχικά, η ανάλυση στη συνέχεια και τέλος η αξιολόγηση των ευπαθειών που ανακαλύφθηκαν στο στάδιο της σάρωσης. Αυτό το στάδιο περιλαμβάνει την εκτίμηση του κινδύνου και την προτεραιοποίηση των ευπαθειών με βάση τη σοβαρότητά τους. Η ευπάθεια που εντοπίστηκε θα πρέπει να σταθμιστεί χρησιμοποιώντας ένα σύστημα βαθμολόγησης για να ελεγχθεί η σοβαρότητά της και οι επιπτώσεις στο σύστημα. Το Common Vulnerability Scoring System (CVSS) είναι ένα δημοσιευμένο πρότυπο που χρησιμοποιείται από οργανισμούς σε όλο τον κόσμο. Το πρότυπο CVSS παρέχει έναν τρόπο καταγραφής των κύριων χαρακτηριστικών μιας ευπάθειας και υπολογίζει μια αριθμητική βαθμολογία που αντικατοπτρίζει τη σοβαρότητά της. Η αριθμητική βαθμολογία μπορεί στη συνέχεια να αντιστοιχηθεί με μια ποιοτική αναπαράσταση (όπως χαμηλή, μεσαία, υψηλή και κρίσιμη) για να βοηθήσει τους οργανισμούς να αξιολογήσουν σωστά και να ιεραρχήσουν τις διαδικασίες διαχείρισης ευπάθειας [37].

4. **Exploitation (Απόκτηση Πρόσβασης)**: Σε αυτό το στάδιο, επιτυγχάνεται η μη εξουσιοδοτημένη είσοδος στα συστήματα με στόχο την εκμετάλλευση των αδυναμιών τους. Για την εκμετάλλευση αναγνωρισμένων ευπαθειών πραγματοποιούνται πραγματικές επιθέσεις στο σύστημα με τη χρήση εργαλείων όπως το Metaspoit για την ανάπτυξη και εκτέλεση exploits που επιβεβαιώνουν την ύπαρξη των ευπαθειών.

5. **Post-Exploitation (Διατήρηση Πρόσβασης)**: Είναι το στάδιο όπου διατηρείτε η πρόσβαση με την εγκατάσταση μηχανισμών παραμονής για συνεχή πρόσβαση και εξαγωγή δεδομένων. Αξιολογείται τόσο η έκταση της πρόσβασης που επιτεύχθηκε όσο και οι πληροφορίες που αποσπάστηκαν. Αυτό το στάδιο περιλαμβάνει τη διαχείριση των συστημάτων που έχουν υποκλαπεί και την εκτίμηση της επίπτωσης της επίθεσης.

6. **Clean up and Reporting (Κάλυψη Ιχνών και Αναφορά)**: Στο 6^{ο} στάδιο γίνεται η αφαίρεση στοιχείων και εργαλείων για αποφυγή εντοπισμού της δοκιμής και γίνεται η καταγραφή των ευρημάτων καθώς και η παροχή συστάσεων. Δημιουργείται μια αναλυτική αναφορά στην οποία περιγράφονται τα ευρήματα των δοκιμών που περιλαμβάνουν τις ευπάθειες που εντοπίστηκαν, τις επιθέσεις που πραγματοποιήθηκαν, καθώς και κάθε προτεινόμενη διορθωτική ενέργεια για την ενίσχυση της ασφάλειας του συστήματος.

Η ανάγκη και η σημασία των επιθέσεων παρείσδυσης είναι σημαντική στην:

1. **Ανίχνευση Ευπαθειών:** Οι επιθέσεις παρείσδυσης αποτελούν κρίσιμο εργαλείο για την ανίχνευση ευπαθειών και αδυναμιών στα συστήματα βάσεων δεδομένων.
2. **Εκμετάλλευση Αδυναμιών:** Μέσω των επιθέσεων παρείσδυσης υπάρχει η δυνατότητα αξιοποίησης τυχόν ευπαθειών για την προσομοίωση πραγματικών επιθέσεων.

Βασική παράμετρος αποτελεί και η συνεχής εκπαίδευση και ενημέρωση των επαγγελματιών ασφαλείας σχετικά με τις νέες τεχνικές και μεθοδολογίες.

5.3 Εργαλεία PenTest για RDBMS

Υπάρχουν διάφορα εργαλεία που χρησιμοποιούνται για το penetration testing σε συστήματα διαχείρισης βάσεων δεδομένων (RDBMS). Τα εργαλεία αυτά είναι κρίσιμα για την εντοπισμό ευπαθειών και τη δοκιμή της ανθεκτικότητας των συστημάτων RDBMS έναντι επιθέσεων. Υπάρχουν διάφορες κατηγορίες εργαλείων. Παρακάτω αναφέρονται διάφορα παραδείγματα με Αυτοματοποιημένα Εργαλεία Εύρεσης Ευπαθειών (Automated Vulnerability Scanners), με Εργαλεία Χειροκίνητων Δοκιμών (Manual Testing Tools), με Εργαλεία που είναι Πλατφόρμες Εκμετάλλευσης (Exploitation Frameworks) και με Εργαλεία Παρακολούθησης και Καταγραφής (Monitoring και Logging Tools) [8] [6] [36]:

1. **SQLMap**: Είναι ένα ανοιχτού κώδικα εργαλείο, γραμμένο σε python, ανίχνευσης και εκμετάλλευσης ευπαθειών SQL Injection. Έχει τη δυνατότητα να εξάγει δεδομένα από τη βάση και να επιβεβαιώσει την ύπαρξη ευπαθειών. Μέσω SQLMap μπορούν να πραγματοποιηθούν επιθέσεις παρείσδυσης αυτοματοποιώντας τη διαδικασία εντοπισμού και εκμετάλλευσης ελαττωμάτων τύπου SQL injection και ελέγχου διακομιστών μιας βάσης δεδομένων. Περιλαμβάνει μια καλή μηχανή ανίχνευσης, αρκετές εξειδικευμένες λειτουργίες και μια ευρεία γκάμα διακοπτών που δίνουν τη δυνατότητα για τη λήψη δεδομένων αλλά και δακτυλικών αποτυπωμάτων της βάσης δεδομένων, την πρόσβαση στο σύστημα διαχείρισης αρχείων και εκτέλεσης εντολών του λειτουργικού συστήματος [38].

2. **Metasploit**: Είναι μια πλατφόρμα που παρέχει διάφορα modules για επιθέσεις σε RDBMS μεταξύ άλλων για εκμετάλλευση ευπαθειών. Αποτελεί ένα ιδιαίτερα χρήσιμο εργαλείο για την επιβεβαίωση και την εκμετάλλευση των ευπαθειών που έχουν εντοπιστεί. Το πιο χρησιμοποιούμενο framework επιθέσεων παρείσδυσης. Είναι ένα προϊόν συνεργασίας μεταξύ της κοινότητας ανοιχτού κώδικα και της εταιρείας κυβερνο-ασφάλειας Rapid7. Ως εργαλείο εκτός από την εύρεση ευπαθειών, βοηθάει στη διαχείριση αξιολογήσεων ασφάλειας και τη βελτίωση της ευαισθητοποίησης σχετικά με την ασφάλεια [39].

3. **Nmap**: Είναι ένα δωρεάν και ανοιχτού κώδικα εργαλείο δικτυακής αναγνώρισης και σάρωσης ανοικτών θυρών και υπηρεσιών. Το Nmap (Network Mapper) είναι βοηθητικό πρόγραμμα χρήσιμο για την αναγνώριση των RDBMS που είναι ενεργά σε ένα δίκτυο, ποιες υπηρεσίες με αναλυτικές πληροφορίες για το όνομα της εφαρμογής καθώς και την έκδοση που προσφέρουν, τα λειτουργικά συστήματα που εκτελούνται με πληροφορίες για την έκδοση του λειτουργικού συστήματος, τον τύπο των φίλτρων πακέτων/τείχους προστασίας που είναι σε χρήση, και πολλά άλλα χαρακτηριστικά. Είναι χρήσιμο για εργασίες όπως για την απογραφή ενός δικτύου, τη διαχείριση χρονοδιαγραμμάτων αναβάθμισης υπηρεσιών και τη παρακολούθηση του χρόνου λειτουργίας ενός κεντρικού υπολογιστή ή μιας υπηρεσίας. Είναι κατάλληλο για σάρωση μεγάλων δικτύων, αλλά λειτουργεί εξίσου καλά και σε μεμονωμένους κεντρικούς υπολογιστές. Το Nmap εκτελείται σε όλα τα μεγάλα λειτουργικά συστήματα υπολογιστών καθώς είναι διαθέσιμο για Linux, Windows και Mac OS X. Εκτός της Nmap εφαρμογής που εκτελείται σε γραμμή εντολών, υπάρχει και η σουίτα Nmap που περιλαμβάνει προηγμένο γραφικό

περιβάλλον. Επιπλέον η έκδοση με το γραφικό περιβάλλον περιλαμβάνει διάφορα βοηθητικά εργαλεία όπως πρόγραμμα προβολής αποτελεσμάτων το Zenmap, μεταφοράς δεδομένων, ανακατεύθυνσης και εντοπισμού σφαλμάτων το Ncat, σύγκρισης των αποτελεσμάτων σάρωσης το Ndiff και ανάλυσης δημιουργίας πακέτων και απόκρισης το Nping. Το Nmap ανακηρύχθηκε Προϊόν Ασφαλείας της Χρονιάς από το Linux Journal, το Info World, το LinuxQuestions.Org και το Codetalker Digest. Συμμετείχε ακόμη και σε δώδεκα ταινίες , συμπεριλαμβανομένων των The Matrix Reloaded, Die Hard 4, Girl With the Dragon Tattoo και The Bourne Ultimatum [40].

4. **Burp Suite**: Εργαλείο για δοκιμές ασφάλειας εφαρμογών ιστού, που περιλαμβάνει δυνατότητες για ελέγχους ευπαθειών SQL Injection. Ένα ολοκληρωμένο εργαλείο για την εκτέλεση δοκιμών ασφάλειας σε web εφαρμογές, που περιλαμβάνει εργαλεία μεταξύ άλλων για τον εντοπισμό και την εκμετάλλευση ευπαθειών, συμπεριλαμβανομένων των ευπαθειών που σχετίζονται με τις βάσεις δεδομένων. Το Burp Suite είναι ένα πακέτο λογισμικού αφιερωμένο σε ελέγχους ασφάλειας ιστού (για εκτελέσεις επιθέσεων παρείσδυσης ιστού). Αναπτύχθηκε από την PortSwigger, μια κορυφαία εταιρεία στον κόσμο της ασφάλειας ιστού. Το Burp Suite, που συχνά αναφέρεται απλώς ως Burp, είναι βελτιστοποιημένο και σχεδιασμένο για να ανταποκρίνεται στις ανάγκες των επαγγελματιών που εκτελούν επιθέσεις παρείσδυσης και είναι το πιο ευρέως χρησιμοποιούμενο εργαλείο στον τομέα του. Είναι ένα αρθρωτό εργαλείο που επιτρέπει τη διενέργεια χειροκίνητων και αυτοματοποιημένων δοκιμών, βοηθώντας τους επαγγελματίες που εκτελούν επιθέσεις παρείσδυσης να εντοπίζουν αποτελεσματικά τα τρωτά σημεία σε εφαρμογές Ιστού [41] [42].

5. **DbProtect**: Επαγγελματικό εργαλείο για την προστασία και τον έλεγχο των βάσεων δεδομένων. Το Trustwave DbProtect αξιολογεί προληπτικά τις απειλές για βάσεις δεδομένων βρίσκοντας τυχόν τρωτά σημεία στις βάσεις δεδομένων ακόμα και αυτών που βρίσκονται στο cloud, που θα μπορούσαν να προκαλέσουν τη παραβίαση δεδομένων. Αυτοματοποιεί την ασφάλεια κρίσιμων δεδομένων αποκαλύπτοντας τρωτά σημεία που αναζητούν οι εισβολείς, περιορίζοντας την πρόσβαση των χρηστών στα πιο ευαίσθητα δεδομένα και ειδοποιώντας για ύποπτες δραστηριότητες, εισβολές και παραβιάσεις πολιτικής [43].

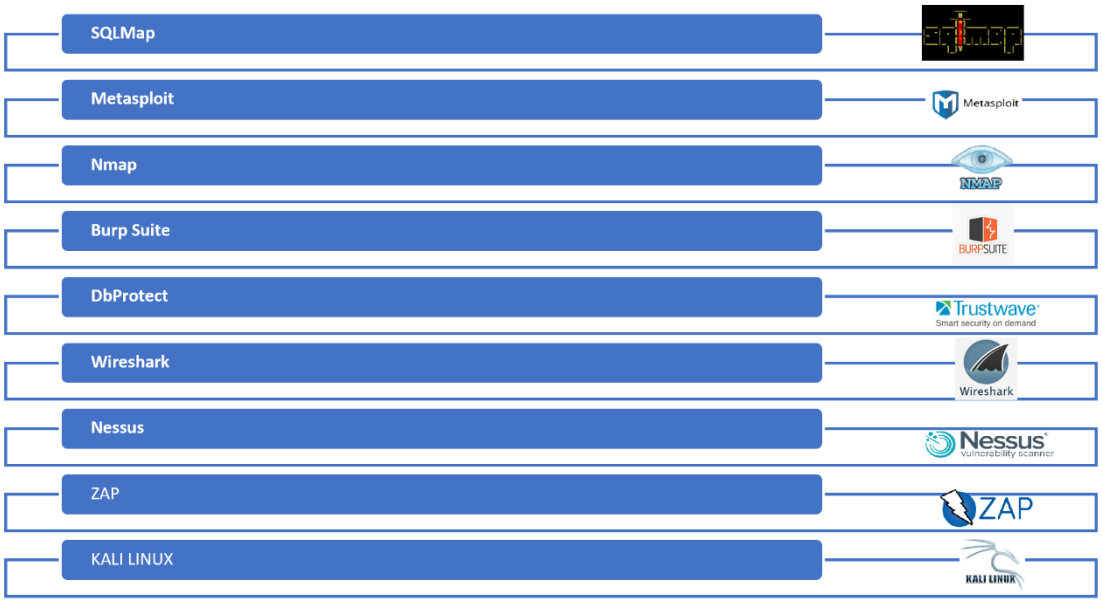
6. **Wireshark**: Εργαλείο ανάλυσης δικτυακής κυκλοφορίας που βοηθά στην παρακολούθηση και ανάλυση των πακέτων δεδομένων που διακινούνται στο δίκτυο. Μπορεί να χρησιμοποιηθεί για την ανίχνευση μη ασφαλών μεταδόσεων δεδομένων. Το Wireshark είναι ο πιο δημοφιλής αναλυτής πρωτοκόλλου δικτύου στον κόσμο. Ανιχνεύει ότι γίνεται σε ένα δίκτυο. Είναι το πρότυπο σε πολλούς κλάδους και εκπαιδευτικά ιδρύματα. Η ανάπτυξη του Wireshark συνεχίζεται χάρη στις συνεισφορές ειδικών δικτύωσης σε όλο τον κόσμο. Αποτελεί τη συνέχεια ενός έργου που ξεκίνησε το 1998 [44].

7. **Nessus**: Αυτοματοποιημένο εργαλείο εύρεσης ευπαθειών που περιλαμβάνει modules για ελέγχους ασφάλειας βάσεων δεδομένων. Το Nessus είναι το πιο ολοκληρωμένο εργαλείο αξιολόγησης ευπάθειας στην αγορά σήμερα. Το Nessus Professional βοηθάει στην αυτοματοποίηση της διαδικασίας σάρωσης ευπάθειας, εξοικονομώντας χρόνο στις ομάδες IT [45].

8. **ZAP**: Είναι ένας σαρωτής ασφαλείας διαδικτυακών εφαρμογών ανοιχτού κώδικα. Το ZAP (Zed Attack Proxy), παλαιότερα γνωστό ως OWASP ZAP, προορίζεται για χρήση τόσο από αρχάριους στην ασφάλεια εφαρμογών όσο και από ομάδες που διενεργούν επιθέσεις παρείσδυσης [46].

9. **Kali Linux**: Το Kali Linux δημιουργήθηκε από την εταιρεία ασφάλειας Offensive Security. Αποτελεί μία έκδοση Linux, που περιέχει μια μεγάλη γκάμα εργαλείων hacking ώστε να αποτελεί ένα εξειδικευμένο εργαλείο επιθέσεων παρείσδυσης. Είναι ένα λειτουργικό σύστημα

σχεδιασμένο για (ηθικό) hacking που βασίζεται στη διανομή Debian Linux. Να σημειωθεί πως το Debian είναι μια από τις καλύτερες διανομές Linux, και το Ubuntu ίσως η πιο δημοφιλής διανομή Linux για υπολογιστές προέρχεται επίσης από το Debian, συνεπώς οι εντολές στο παράθυρο του τερματικού είναι ίδιες. Αυτό δίνει το πλεονέκτημα η βάση των χρηστών που είναι εξοικειωμένοι με τις εντολές στο παράθυρο του τερματικού να είναι μεγαλύτερη [18].



Σχήμα 5.3: Εργαλεία για Penetration testing

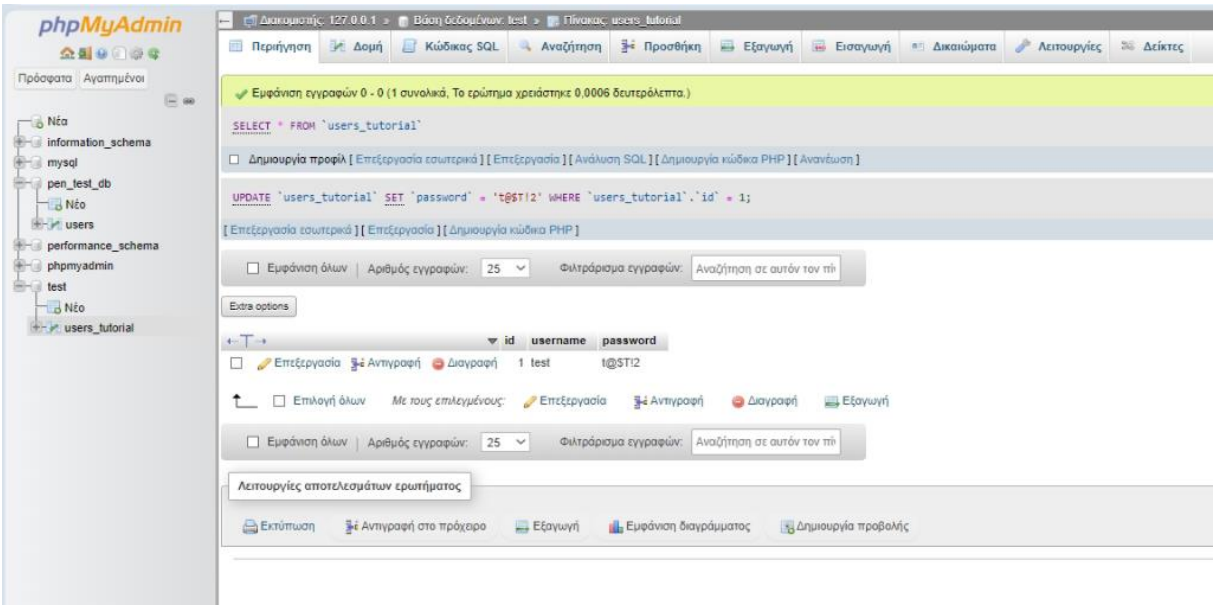
Οι μεθοδολογίες και τα εργαλεία για pen testing σε RDBMS είναι κρίσιμα για την εξασφάλιση της ασφάλειας των βάσεων δεδομένων. Η χρήση των παραπάνω μεθοδολογιών και εργαλείων επιτρέπει την αναγνώριση και εκμετάλλευση ευπαθειών, βοηθώντας τους οργανισμούς να κατανοήσουν τα τρωτά σημεία τους ώστε να πάρουν κάθε απαραίτητο μέτρο που χρειάζεται για την καλύτερη προστασία των δεδομένων τους.

Κεφάλαιο 6ο: Παρουσίαση Μιας Μελέτης Περίπτωσης Εφαρμογής

6.1 Εισαγωγή

Η εργασία μετά την βιβλιογραφική καταγραφή επιχειρεί να δοκιμάσει κάποια από τα ευρήματα της. Έτσι επιλέχθηκαν 2 εργαλεία που αντιπροσωπευτικά μπορούν να καταδείξουν τις δυνατότητες εύρεσης ευπαθειών σε Διαδικτυακές Βάσεις Δεδομένων.

6.2 Το πειραματικό περιβάλλον



Σχήμα 6.1: Η πειραματική Βάση Δεδομένων μέσα από το διαχειριστικό περιβάλλον του phpMyAdmin

Ο SQL κώδικας της Βάσης Δεδομένων είναι ο παρακάτω:

-- phpMyAdmin SQL Dump

-- version 5.2.1

-- https://www.phpmyadmin.net/

--

-- Εξυπηρετητής: 127.0.0.1

-- Χρόνος δημιουργίας: 19 Οκτ 2024 στις 18:17:36

-- Έκδοση διακομιστή: 10.4.32-MariaDB

-- Έκδοση PHP: 8.2.12

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";

START TRANSACTION;

SET time_zone = "+00:00";

```
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Βάση δεδομένων: `test`
--

-- --------------------------------------------------------

--
-- Δομή πίνακα για τον πίνακα `users_tutorial`
--

CREATE TABLE `users_tutorial` (
  `id` int(11) NOT NULL,
  `username` varchar(20) NOT NULL,
  `password` varchar(20) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;

--
-- Άδειασμα δεδομένων του πίνακα `users_tutorial`
--

INSERT INTO `users_tutorial` (`id`, `username`, `password`) VALUES
(1, 'test', 't@$T!2');

--
-- Ευρετήρια για άχρηστους πίνακες
--
```

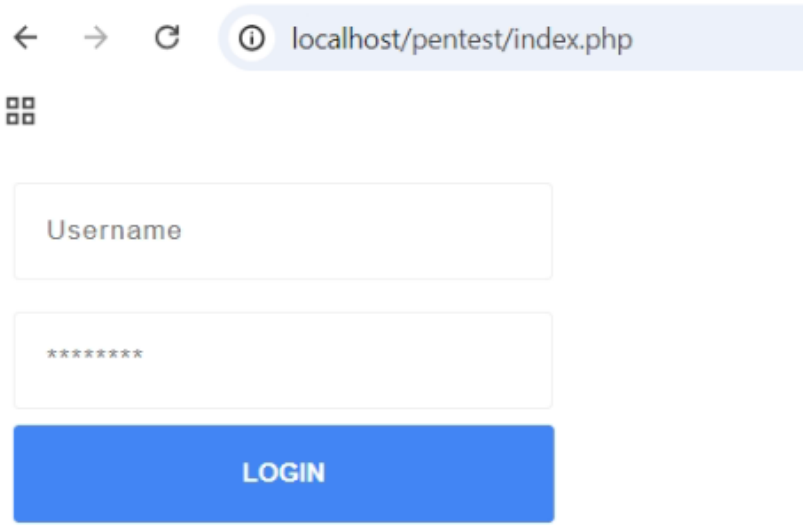
```
--
-- Ευρετήρια για πίνακα `users_tutorial`
--
ALTER TABLE `users_tutorial`
  ADD PRIMARY KEY (`id`);


--
-- AUTO_INCREMENT για άχρηστους πίνακες
--


--
-- AUTO_INCREMENT για πίνακα `users_tutorial`
--
ALTER TABLE `users_tutorial`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3;
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

Στη συνέχεια παρατίθεται ο κώδικας της σελίδας **Index.php** που δημιουργήθηκε, η οποία περιέχει αντίμετρα για SQLi επιθέσεις:

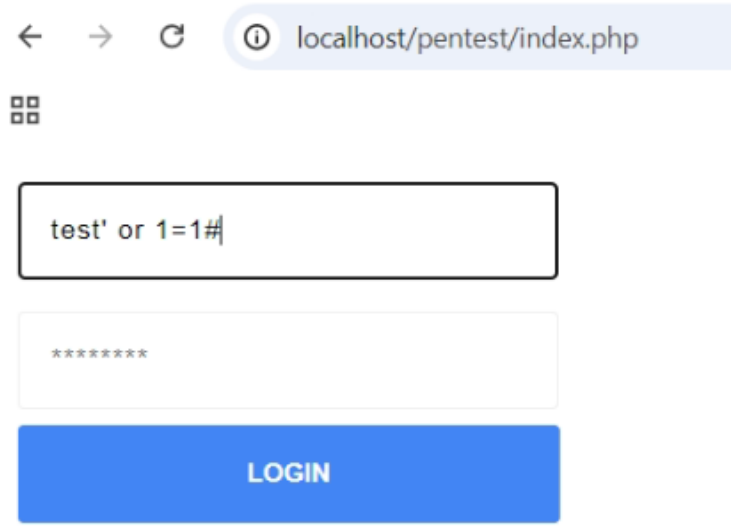
```
<?php
$hostname = "localhost";
$username = "root";
$password = "";
$dbname = "test";
$conn = mysqli_connect($hostname, $username, $password, $dbname);
if(!$conn) {
        die("Unable to connect");
```

```
}
if($_POST) {
	$uname = $_POST["username"];
	$pass = $_POST["password"];
	//Making sure that SQL Injection doesn't work
	$uname = mysqli_real_escape_string($conn, $uname);//test or 1=1
	$pass = mysqli_real_escape_string($conn, $pass);
	$sql = "SELECT * FROM users_tutorial WHERE username = '$uname' AND password = '$pass'";
	$result = mysqli_query($conn, $sql);
	if(mysqli_num_rows($result) == 1) {
		echo "Welcome, user!";
	} else {
		echo "Incorrect Username/Password";
	}
}
?>
<!DOCTYPE html>
<html>
<head>
	<title>Login Portal</title>
	<style type="text/css">
		input[type=text],input[type=password] {
			padding: 16px;
			margin: 8px;
			border: 1px solid #f1f1f1;
			letter-spacing: 1px;
			border-radius: 3px;
			width: 240px;
		}
		input[type=submit] {
			margin-left: 8px;
			width: 274px;
```

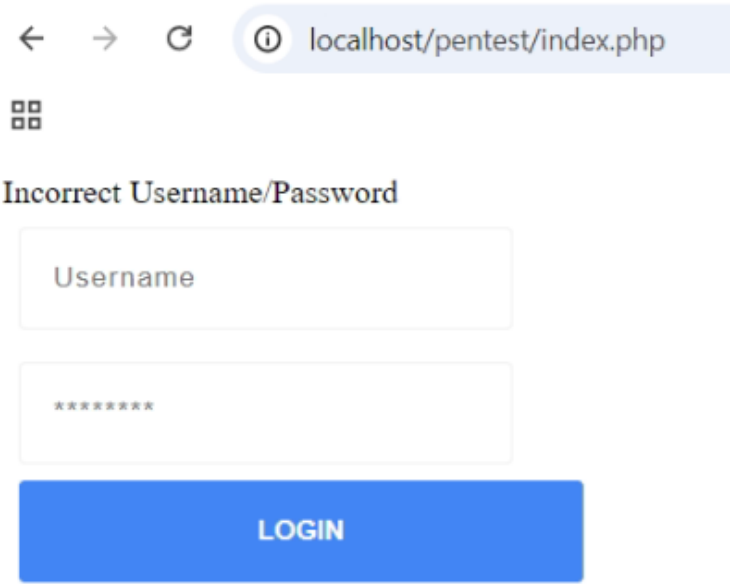
```
                    border-radius: 3px;
                    border: 1px solid #4285f4;
                    background-color: #4285f4;
                    padding: 16px;
                    color: white;
                    font-weight: 600;
                    cursor: pointer;
                }
        </style>
</head>
<body>
        <form action method="POST" autocomplete="off">
                <input type="text" name="username" placeholder="Username" /><br />
                <input type="password" name="password" placeholder="********" /><br />
                <input type="submit" name="login" value="LOGIN" />
        </form>
</body>
</html>
```



Σχήμα 6.2: Η φόρμα εισαγωγής της σελίδας για έλεγχο με δεδομένα ελέγχου της ετοιμότητας της σελίδας για επίθεση SQLi



Σχήμα 6.3: Τα δεδομένα ελέγχου της ετοιμότητας της σελίδας για επίθεση SQLi

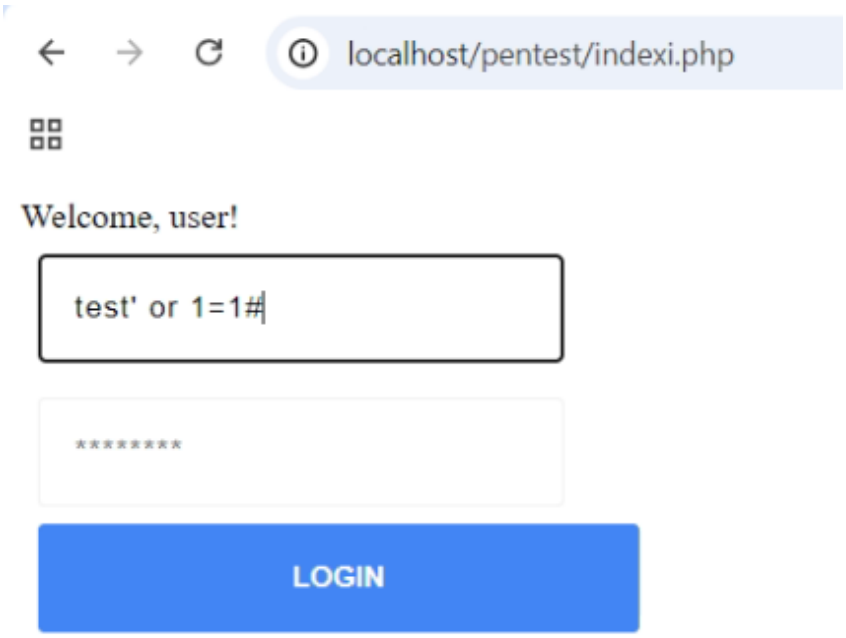


Σχήμα 6.4: Το μήνυμα λάθους εισαγωγής δεδομένων που εμφανίζεται στη σελίδα μετά από επίθεση SQLi

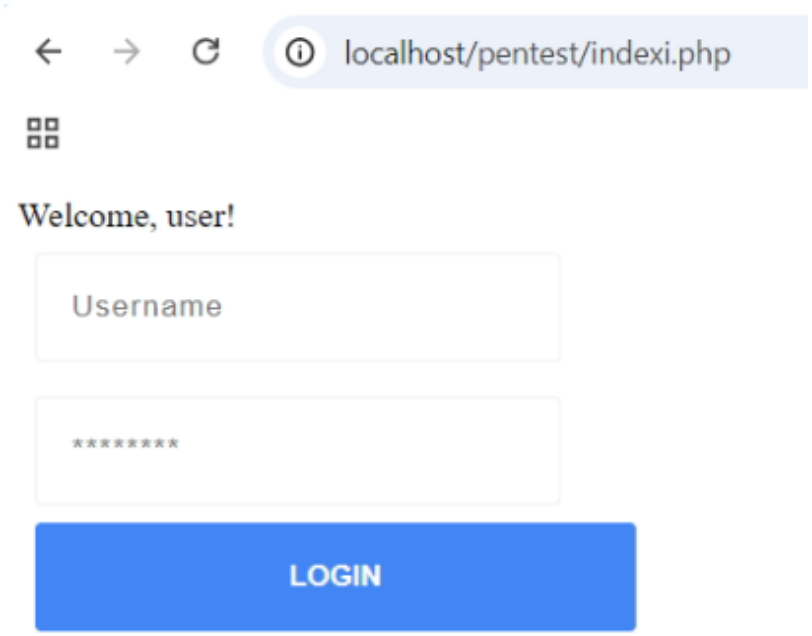
Αντίστοιχα ο κώδικας της σελίδας **Indexi.php** που δημιουργήθηκε για να είναι ευάλωτος σε SQLi:

```
<?php
$hostname = "localhost";
$username = "root";
$password = "";
$dbname = "test";
$conn = mysqli_connect($hostname, $username, $password, $dbname);
if(!$conn) {
	die("Unable to connect");
}
if($_POST) {
	$uname = $_POST["username"];
	$pass = $_POST["password"];
	//Making sure that SQL Injection doesn't work
	//$uname = mysqli_real_escape_string($conn, $uname);//test or 1=1
	//$pass = mysqli_real_escape_string($conn, $pass);
	$sql = "SELECT * FROM users_tutorial WHERE username = '$uname' AND password = '$pass'";
	$result = mysqli_query($conn, $sql);
	if(mysqli_num_rows($result) == 1) {
		echo "Welcome, user!";
	} else {
		echo "Incorrect Username/Password";
	}
}
?>
<!DOCTYPE html>
<html>
<head>
	<title>Login Portal</title>
	<style type="text/css">
		input[type=text],input[type=password] {
```

```
                padding: 16px;
                margin: 8px;
                border: 1px solid #f1f1f1;
                letter-spacing: 1px;
                border-radius: 3px;
                width: 240px;
            }
            input[type=submit] {
                margin-left: 8px;
                width: 274px;
                border-radius: 3px;
                border: 1px solid #4285f4;
                background-color: #4285f4;
                padding: 16px;
                color: white;
                font-weight: 600;
                cursor: pointer;
            }
        </style>
</head>
<body>
        <form action method="POST" autocomplete="off">
            <input type="text" name="username" placeholder="Username" /><br />
            <input type="password" name="password" placeholder="********" /><br />
            <input type="submit" name="login" value="LOGIN" />
        </form>
</body>
</html>
```



Σχήμα 6.5: Η φόρμα εισαγωγής της σελίδας που είναι ευάλωτη σε SQLi επιθέσεις



Σχήμα 6.6: Η SQLi επίθεση είναι επιτυχής αποκτώντας πρόσβαση στην εφαρμογή και στη Βάση Δεδομένων

6.3 Επιλογή Συστήματος για Μελέτη Περίπτωσης

Η μελέτη περίπτωσης αφορά μια web εφαρμογή που χρησιμοποιεί MySQL για τη διαχείριση των δεδομένων της. Βασιζόμενοι στη [47], η ερευνητική μελέτη διεξήχθη σε τέσσερα βήματα.

Τα τέσσερα βήματα που αποτέλεσαν την πειραματική μελέτη:

1. Προετοιμασία: Στο βήμα αυτό γίνεται η ρύθμιση του περιβάλλοντος δοκιμών. Η εφαρμογή και η Βάση Δεδομένων που θα εξεταστούν, καθώς και οι επιθέσεις διείσδυσης και τα αντίστοιχα εργαλεία που θα δοκιμαστούν θα πρέπει να εγκατασταθούν στο περιβάλλον δοκιμών.
2. Εκτέλεση: Χρήση των εργαλείων για εκμετάλλευση και εύρεση τυχόν αδυναμιών στη Βάση Δεδομένων.
3. Επαλήθευση: Πραγματοποίηση χειροκίνητων δοκιμών για να βεβαιωθούν ότι τα τρωτά σημεία που βρέθηκαν από τα εργαλεία είναι πραγματικά.
4. Ανάλυση: Αξιολόγηση των αποτελεσμάτων.

6.4 Παραδείγματα Παρείσδυσης

6.4.1 Προετοιμασία

Αρχικά δημιουργήθηκε μια απλή εφαρμογή εισόδου με βάση το [48]. Είναι μιας απλή ιστοσελίδα που επικοινωνεί με μια βάση δεδομένων MySQL. Δημιουργήθηκαν δύο παραλλαγές της, μία που είναι ευάλωτη σε SQLi και μία που λαμβάνει αντίμετρα. Τρέχει τοπικά με χρήση XAMPP και εκκίνηση/ενεργοποίηση του εξυπηρετητή Apache και της βάσης δεδομένων MySQL. Έτσι στο localhost (127.0.0.1) στον υποφάκελο pentest υπάρχουν δύο σελίδες η index.php και η indexi.php. Η διαφορά τους είναι στην ύπαρξη στην πρώτη στον PHP κώδικα η συνάρτηση mysqli_real_escape_string ($uname = mysqli_real_escape_string($conn, $uname);) για να αντιμετωπίζει SQLi επιθέσεις, ενώ η δεύτερη είναι ευάλωτη σε τέτοιου είδους επιθέσεις.

Στη συνέχεια έγινε εγκατάσταση των εφαρμογών ανοιχτού κώδικα sqlmap (https://sqlmap.org/) και nmap (https://nmap.org/) από τις αντίστοιχες σελίδες.

Πριν την εκτέλεση μέσα από το XAMPP ενεργοποιήθηκαν τοπικά ο εξυπηρετητής Apache και η βάση δεδομένων MySQL.

6.4.2 Εκτέλεση

Με τη χρήση του Nmap ως εργαλείο αναγνώρισης των RDBMS που είναι ενεργά σε ένα δίκτυο και την χρήση των εργαλείων όπως το SQLMap για να εντοπιστούν ευπάθειες SQL Injection και μη ασφαλής αποθήκευση κωδικών πρόσβασης.

Στη συνέχεια έτρεξε το Zenmap που είναι μια γραφική διεπαφή χρήστη για το Nmap. Είναι ένα δωρεάν λογισμικό ανοιχτού κώδικα που βοηθά την εκκίνηση και λειτουργία του Nmap.

Υπάρχουν διάφορες επιλογές εκτέλεσης της. Μέσα από το γραφικό περιβάλλον του Zenmap εκτελέστηκε το profile: Intense Scan. Ουσιαστικά εκτελέστηκε η εντολή: nmap -T4 -A -v 127.0.0.1

Η πληροφορία που εξήχθη ήταν:

PORT: 80/tcp

STATE SERVICE: open http

VERSION: Apache httpd 2.4.58 ((Win64) OpenSSL/3.1.3 PHP/8.2.12)

PORT: 135/tcp

STATE SERVICE: open msrpc

VERSION: Microsoft Windows RPC

PORT: 443/tcp

STATE SERVICE: open ssl/http

VERSION: Apache httpd 2.4.58 ((Win64) OpenSSL/3.1.3 PHP/8.2.12)

(Issuer: commonName=localhost, Public Key type: rsa, Public Key bits: 1024 , Signature Algorithm: sha1WithRSAEncryption, Not valid before: 2009-11-10T23:48:47, Not valid after: 2019-11-08T23:48:47, MD5: a0a4:4cc9:9e84:b26f:9e63:9f9e:d229:dee0, SHA-1: b023:8c54:7a90:5bfa:119c:4e8b:acca:eacf:3649:1ff6)

PORT: 445/tcp

STATE SERVICE: open microsoft-ds?

VERSION:

PORT: 3306/tcp

STATE SERVICE: open mysql

VERSION: MariaDB 5.5.5-10.4.32

(mysql-info: Protocol: 10, Version: 5.5.5-10.4.32-MariaDB, Thread ID: 10, Capabilities flags: 63486, Some Capabilities: ODBCClient, Support41Auth, DontAllowDatabaseTableColumn, Speaks41ProtocolOld, ConnectWithDatabase, LongColumnFlag, SupportsLoadDataLocal, Speaks41ProtocolNew, FoundRows, InteractiveClient, IgnoreSpaceBeforeParenthesis, SupportsCompression, SupportsTransactions, IgnoreSigpipes, SupportsMultipleResults, SupportsAuthPlugins, SupportsMultipleStatments, Auth Plugin Name: mysql_native_password , Status: Autocommit)

PORT: 5432/tcp

STATE SERVICE: open postgresql

VERSION: PostgreSQL DB

Πληροφορίες που προέκυψαν αφορούν τον εξυπηρετητή της διεύθυνσης που δόθηκε και οι βάσεις που αυτός τρέχει, και είναι πληροφορίες χρήσιμες σε έναν επιτιθέμενο.

Στη συνέχεια, με το εργαλείο sqlmap εκτελούμε την παρακάτω εντολή για να προσδιορίσουμε ευπάθειες αυθεντικοποίησης μιας σελίδας που περιέχει πεδία για εισαγωγή στοιχείων για είσοδο σε κάποια διαδικτυακή εφαρμογή:

python sqlmap.py -u "http://localhost/pentest/indexi.php" --data="username=' or '1'='1"

Τα ευρήματα της εκτέλεσης του sqlmap ήταν:

sqlmap resumed the following injection point(s) from stored session:

Parameter: username (POST)

Type: boolean-based blind

Title: MySQL RLIKE boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause

Payload: username=' or '1'='1' RLIKE (SELECT (CASE WHEN (3471=3471) THEN '' or 0x31=0x31 ELSE 0x28 END))-- XHfG

Type: error-based

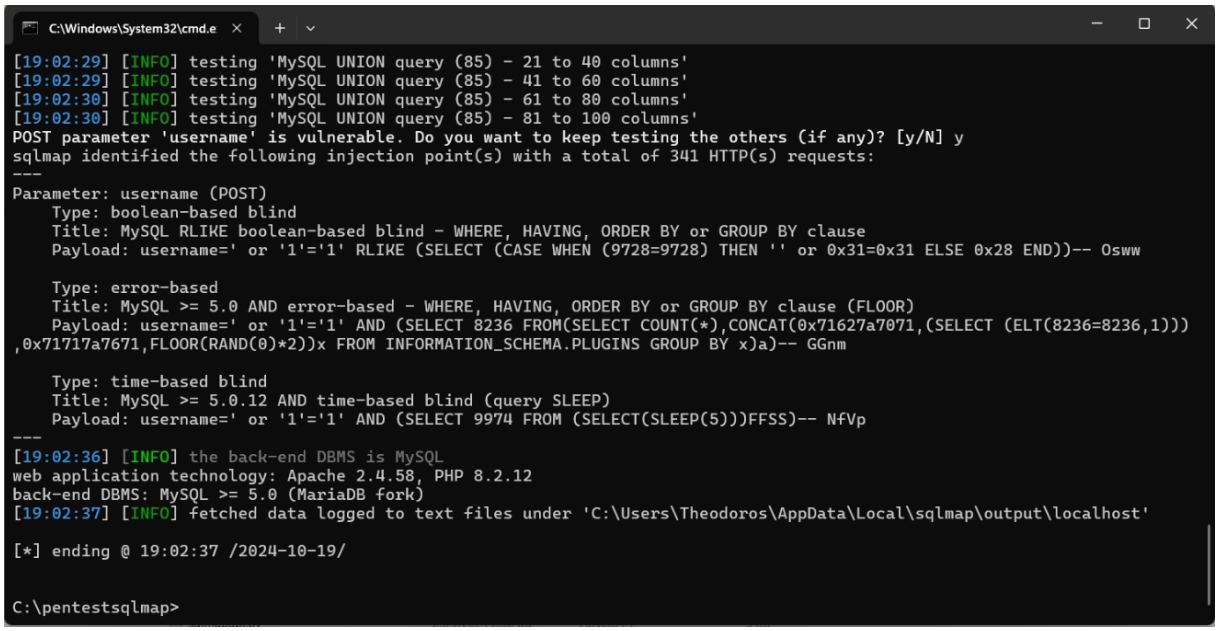
Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)

Payload: username=' or '1'='1' AND (SELECT 9333 FROM(SELECT COUNT(*),CONCAT(0x716a7a6a71,(SELECT (ELT(9333=9333,1))),0x716a766b71,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a)-- GQPo

Type: time-based blind

Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)

Payload: username=' or '1'='1' AND (SELECT 7045 FROM (SELECT(SLEEP(5)))zjUV)-- KSUs



Σχήμα 6.7: Χρήση του sqlmap για εύρεση ευπαθειών αυθεντικοποίησης μιας ιστοσελίδας

Όταν προστέθηκε στον PHP κώδικα η συνάρτηση mysqli_real_escape_string ($uname = mysqli_real_escape_string($conn, $uname);), τα ευρήματα της εκτέλεσης του sqlmap ήταν:

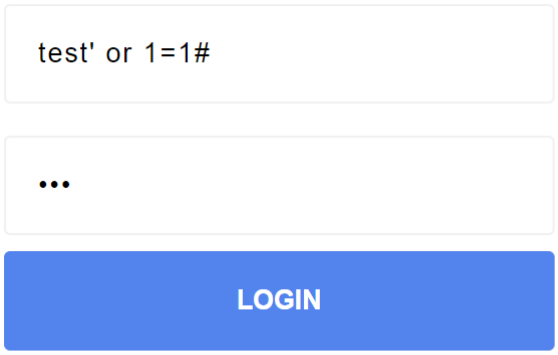
[CRITICAL] all tested parameters do not appear to be injectable. Try to increase values for '--level'/'--risk' options if you wish to perform more tests. If you suspect that there is some kind of protection mechanism involved (e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper=space2comment') and/or switch '--random-agent'

Δηλαδή ο έλεγχος από το εργαλείο έδειξε πως δεν υπάρχει ευπάθεια σε SQLi επιθέσεις.

6.4.3 Επαλήθευση

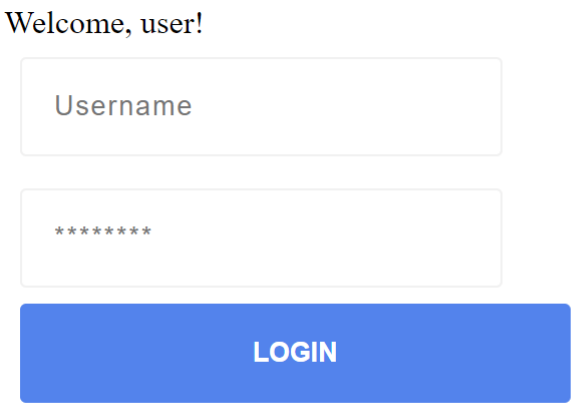
Ακολουθώντας τη μεθοδολογία το 3 βήμα είναι η πραγματοποίηση χειροκίνητων δοκιμών για να επιβεβαίωση πως τα τρωτά σημεία που βρέθηκαν από τα εργαλεία είναι πραγματικά.

Έτσι δοκιμάζουμε χειροκίνητα την εισαγωγή δεδομένων, στη σελίδα indexi.php στο πεδίο username test’ or 1=1# και στο password οποιαδήποτε τιμή.



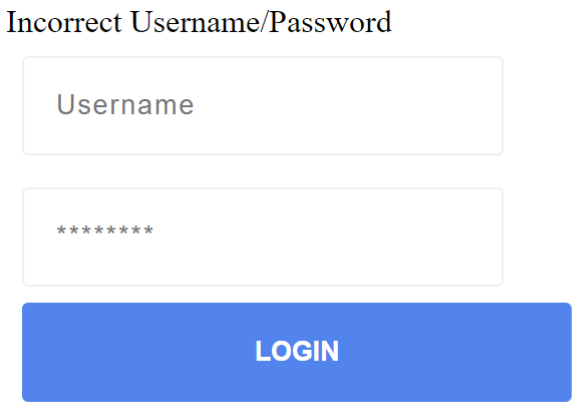
Σχήμα 6.8: Δοκιμή για εύρεση ευπάθειας SQLi

Και πράγματι η εφαρμογή δέχεται τα δεδομένα εισόδου του επιτιθέμενου ως σωστά και τον βάζει στο περιβάλλον της εφαρμογής.



Σχήμα 6.9: SQLi ευπάθεια

Αντίστοιχα, δοκιμάζουμε χειροκίνητα την εισαγωγή δεδομένων, στη σελίδα index.php και η χρήση SQLi τεχνικών αποδεικνύεται ανεπιτυχής.



Σχήμα 6.10: Αντιμετώπιση SQLi επιθέσεων με χρήση αντίμετρων

6.4.4 Ανάλυση

Με την εκμετάλλευση των ευπαθειών, κατέστη δυνατή η πρόσβαση στα δεδομένα της βάσης και η απόκτηση πρόσβασης με δικαιώματα διαχειριστή.

Τα κενά ασφαλείας αποτελούν μια μεγάλη πρόκληση και απαιτείται αυξημένη προσοχή σε κάθε διαδικτυακή εφαρμογή ειδικά αυτές που έχουν επικοινωνία με βάσεις δεδομένων. Πολλές από αυτές τις βάσεις δεδομένων περιέχουν πολύτιμες πληροφορίες, όπως προσωπικά δεδομένα και οικονομικές φύσης πληροφορίες, που τις καθιστούν συχνό στόχο επιθέσεων. Με την επίθεση επιχειρείται η πρόσβαση στα δεδομένα που βρίσκονται στη βάση δεδομένων και για αυτό χρησιμοποιούνται διάφορες τεχνικές. Ο επιτιθέμενος εκτός από τεχνικές γνώσεις που πρέπει να έχει βάζει και μια δόση δημιουργικότητας αλλά παράλληλα χρειάζεται και την τύχη, αλλά και η ύπαρξη της ευπάθειας στις

βάσεις δεδομένων και στις εφαρμογές Ιστού στις μέρες μας χρειάζονται την αμέλεια ή το ανθρώπινο λάθος.

6.5 Μελλοντικές επεκτάσεις και άλλες τεχνικές παρείσδυσης

Μέσω δοκιμών παρείσδυσης γίνεται αναζήτηση λογικών ελαττωμάτων που υπάρχουν σε Βάσεις Δεδομένων. Η μεγάλη ανάπτυξη εφαρμογών Ιστού και η ποικιλία υποδομών και υπηρεσιών που τις φιλοξενούν καθιστούν ευάλωτες τις εφαρμογές και τις Βάσεις Δεδομένων τους.

Οι δοκιμές παρείσδυσης πρέπει να δοκιμαστούν σε όλους τους τύπους πλατφορμών και τεχνολογιών που υποστηρίζουν εφαρμογές Ιστού και τις Βάσεις Δεδομένων τους. Αυτές οι δοκιμές, όπως αναφέρθηκε στην εργασία, περιλαμβάνουν ελέγχους που αφορούν μη εξουσιοδοτημένη πρόσβαση εισβολέων που αποκτούν δικαιώματα σε δεδομένα και τα αυτόματα εργαλεία δεν μπορούν να εντοπίσουν τα τρωτά σημεία. Καθώς υπάρχουν πλειάδα εργαλείων για να πραγματοποιηθούν οι δοκιμές παρείσδυσης και τα οποία τρέχουν σε διάφορα περιβάλλοντα μπορούν να πραγματοποιηθούν πολλά σενάρια δοκιμών και μέσω αυτών να βρεθούν τρωτά σημεία αλλά και να μετρηθούν τυχόν συνέπειες μιας ευπάθειας.

6.6 Συστάσεις για Βελτίωση της Ασφάλειας

Οι διαδικτυακές εφαρμογές από τη φύση τους πρέπει να είναι διαθέσιμες διαρκώς. Πολλές από αυτές τις εφαρμογές Ιστού είναι εξατομικευμένες και αυτό τις κάνει να έχουν λιγότερες δοκιμές με συνέπεια μεγαλύτερα ευαισθησία σε κενά ασφαλείας. Αυτό τις καθιστά και πιο επιρρεπείς σε επιθέσεις. Οι εισβολείς αναζητούν αδυναμίες και κενά που μπορεί να υπάρχουν κι αν αντιληφθούν κάποια ευπάθεια τότε μπορούν εύκολα να τις εκμεταλλευτούν. Επιπρόσθετα, όλα τα σύγχρονα συστήματα βάσεων δεδομένων όπως τα Microsoft SQL Server, Oracle, MySQL, PostgreSQL και IBM DB2 μπορούν να προσπελαστούν μέσω συγκεκριμένων θυρών (π.χ. η MySQL από τη θύρα 3306) και οποιοσδήποτε μπορεί να προσπαθήσει να συνδεθεί με τις βάσεις δεδομένων παρακάμπτοντας αποτελεσματικά τους μηχανισμούς ασφαλείας που χρησιμοποιεί το λειτουργικό σύστημα. Αυτές οι θύρες που είναι γνωστές στους εισβολείς, παραμένουν ανοιχτές για να επιτρέπουν την επικοινωνία των βάσεων δεδομένων με το υπόλοιπο δίκτυο αποτελούν όμως ταυτόχρονα μια ευπάθεια ασφαλείας. Σε κάθε περίπτωση καλό είναι για την καλύτερη ασφάλεια τους να φροντίζουν για την εφαρμογή των παρακάτω προτάσεων:

1. **Ενίσχυση των Μηχανισμών Επαλήθευσης Εισόδου**: Χρήση προετοιμασμένων δηλώσεων και αποφυγή δυναμικών SQL.
2. **Κρυπτογράφηση Δεδομένων:** Εφαρμογή κρυπτογράφησης για την αποθήκευση ευαίσθητων δεδομένων.
3. **Τακτικοί Έλεγχοι Ασφάλειας:** Διεξαγωγή τακτικών επιθέσεων παρείσδυσης και αξιολόγηση των πολιτικών ασφαλείας.

Κεφάλαιο 7ο: Συμπεράσματα

Η ανάπτυξη του ηλεκτρονικού εμπορίου μέσω της ανάπτυξης του Παγκόσμιου Ιστού είναι μια πραγματικότητα και η ανάπτυξη αυτή συνεχίζεται αμείωτη. Πολλές και διαφορετικές τεχνολογίες και πλατφόρμες υποστηρίζουν διαδικτυακές εφαρμογές ηλεκτρονικού εμπορίου. Η τεράστια ανάπτυξη και η πολυπλοκότητα που παρουσιάζουν τέτοιες εφαρμογές με την ποικιλία των τεχνολογιών που τις υποστηρίζουν κάνουν ιδιαίτερα κρίσιμη την ασφάλεια τους. Ευαίσθητα προσωπικά δεδομένα επιβάλλεται να προστατευτούν, και οι πολιτικές αντιμετώπισης επιθέσεων προβάλλονται αναδεικνύοντας τη σημασία της ασφάλειας στο σημερινό ψηφιακό κόσμο.

Η ασφάλεια των Συστημάτων Βάσεων Δεδομένων αποτελεί έναν από τους κρίσιμους τομείς στον χώρο της πληροφορικής. Η ασφάλεια των συστημάτων αποτελεί σημαντική προτεραιότητα στον κυβερνοχώρο, για το λόγο αυτό οι επιχειρήσεις αναζητούν ανθρώπους να πραγματοποιήσουν επιθέσεις παρείσδυσης για να χρησιμοποιήσουν τις δεξιότητές τους στο hacking ώστε να αποκαλύψουν κενά ασφαλείας που θα μπορούσαν να εκμεταλλευτούν εισβολείς και να προστατεύσουν έτσι ευαίσθητα ψηφιακά δεδομένα. Οι επιθέσεις παρείσδυσης αποκαλύπτουν αδυναμίες που οδηγούν σε πολιτικές αντιμετώπισης τους ώστε να ενισχυθεί η ασφάλεια των συστημάτων. Οι απειλές στον κυβερνοχώρο εξελίσσονται συνεχώς για να παρακάμψουν τα βελτιωμένα συστήματα ασφαλείας και η πραγματοποίηση κακόβουλων επιθέσεων θα παραμένει μια διαρκής ανησυχία με στόχο να εκβιαστούν οι εταιρείες.

Οι επιθέσεις παρείσδυσης θεωρούνται μια καλά οργανωμένη μέθοδος αξιολόγησης των υπολογιστικών συστημάτων ενός οργανισμού, τα οποία είναι ένας συνδυασμός του υλικού, του λογισμικού αλλά και των ανθρώπων που τα χρησιμοποιούν. Προκειμένου να εντοπιστούν τρωτά σημεία ασφαλείας, απειλές και κίνδυνοι που μπορεί ένας εισβολέας να εκμεταλλευτεί σε εφαρμογές λογισμικού, δίκτυα ή εφαρμογές του Ιστού, αυτή η τεχνική περιλαμβάνει την ανάλυση του υπολογιστικού συστήματος της εταιρείας για την αναζήτηση ελαττωμάτων. Η αναζήτηση είναι πολύ-επίπεδη και κατά τις επιθέσεις παρείσδυσης γίνεται ενεργή εξέταση του συστήματος για τυχόν ευπάθειες. Εξετάζονται σφάλματα σχεδίασης, λανθασμένη ή κατώτερη του επιθυμητού διαμόρφωση του συστήματος, ελαττώματα στο υλικό και το λογισμικό αλλά και αδυναμίες στις διαδικασίες λειτουργίας. Υπάρχουν διάφοροι τύποι επιθέσεων/δοκιμών παρείσδυσης. Η πραγματοποίηση επιθέσεων παρείσδυσης είναι χρήσιμες προκειμένου να εξεταστεί η αποτελεσματικότητα των μέτρων ασφαλείας ενός συστήματος, όπως και για τον μετριασμό του κινδύνου, την ευαισθητοποίηση σχετικά με την ασφάλεια γενικότερα αλλά και αποτελεί βοήθεια για το προσωπικό ώστε να μάθει πως να ενεργεί έναντι κακόβουλων ενεργειών και να διασφαλίσει τη συμμόρφωση για την αντιμετώπιση επιθέσεων. Μέσω της σωστής αξιολόγησης των προβλημάτων ασφάλειας, της χρήσης αποτελεσματικών εργαλείων και μεθοδολογιών PenTest, και της εφαρμογής προληπτικών μέτρων, είναι δυνατόν να ενισχυθεί η ασφάλεια των Βάσεων Δεδομένων και να προστατευθούν τα κρίσιμα δεδομένα από κακόβουλες επιθέσεις και να διασφαλιστεί η ασφάλεια και κατά συνέπεια και η ποιότητα των συστημάτων.

7.1 Επίλογος

Ανακεφαλαιώνοντας τα κύρια σημεία του κάθε κεφαλαίου, στο 1ο κεφάλαιο έγινε μια εισαγωγική αναφορά στη σημασία των δεδομένων και των Βάσεων Δεδομένων στο σύγχρονο κόσμο. Επίσης δόθηκε μια αναφορά στην ασφάλεια των συστημάτων Βάσεων Δεδομένων, και στη συμβολή των δοκιμών παρείσδυσης για τον εντοπισμό και την αξιολόγηση των ευπαθειών στα συστήματα αυτά.

Στο 2ο Κεφάλαιο έγινε καταγραφή και αξιολόγηση των προβλημάτων ασφαλείας των Συστημάτων Βάσεων Δεδομένων. Αρχικά δόθηκε μια γενική εισαγωγή στα Προβλήματα Ασφάλειας των RDBMS και στους τρόπους προστασίας για την αποτροπή προβλημάτων Ασφάλειας των RDBMS. Στη συνέχεια αναλύθηκαν προβλήματα ασφάλειας, όπως επιθέσεις SQL Injection, Privilege Escalation, Data Breaches, Denial of Service (DoS), Weak Authentication Mechanisms, Buffer Overflows, κρυπτογράφηση και προστασία δεδομένων καθώς και Auditing και Monitoring.

Στο 3ο Κεφάλαιο έγινε μια επισκόπηση των δημοφιλέστερων RDBMS γίνεται στο κεφάλαιο αυτό. Στη συνέχεια του κεφαλαίου αναφέρονται μηχανισμοί προστασίας έναντι των ευπαθειών. Τέλος δόθηκε μια συγκριτική αξιολόγηση των καταγεγραμμένων ευπαθειών ασφαλείας για τα πιο διαδεδομένα RDBMS γίνεται στο κεφάλαιο αυτό.

Στο 4ο Κεφάλαιο αναλύθηκαν οι επιθέσεις παρείσδυσης. Πως λειτουργούν, ποια είναι η διαδικασία εκτέλεσης μιας επίθεσης παρείσδυσης δικτύου και ποια τα οφέλη που αποκομίζουν οι εταιρείες από τις δοκιμές αυτές.

Στο 5ο Κεφάλαιο έγινε περιγραφή των Μεθοδολογιών και των Εργαλείων PenTest για RDBMS. Αναλύθηκαν τα στάδια μιας επίθεσης Παρείσδυσης και αναφέρθηκαν διάφορα παραδείγματα εργαλείων όπως Αυτοματοποιημένα Εργαλεία Εύρεσης Ευπαθειών (Automated Vulnerability Scanners), Εργαλεία Χειροκίνητων Δοκιμών (Manual Testing Tools), Εργαλεία που είναι Πλατφόρμες Εκμετάλλευσης (Exploitation Frameworks) και Εργαλεία Παρακολούθησης και Καταγραφής (Monitoring και Logging Tools).

Στο 6ο Κεφάλαιο 6 έγινε η παρουσίαση μιας μελέτης Περίπτωσης Εφαρμογής.

Κλείνοντας στο 7ο Κεφάλαιο παρουσιάστηκε μια επισκόπηση της έρευνας των Επιθέσεων Παρείσδυσης (Penetration testing) σε Δικτυακές Βάσεις Δεδομένων και δόθηκαν τα συμπεράσματα της έρευνας και τα πλεονεκτήματα που οι δοκιμές λογισμικού προσδίδουν.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] E. Ferrari and B. Thuraisingham, *Security and Privacy for Web Databases and Services*, vol. 2992. 2004, p. 28. doi: 10.1007/978-3-540-24741-8_3.

[2] Cameron Hashemi-Pour, "What is the CIA triad (confidentiality, integrity and availability)?," TechTarget. Accessed: Aug. 21, 2024. [Online]. Available: https://www.techtarget.com/whatis/definition/Confidentiality-integrity-and-availability-CIA

[3] E. Bertino and R. Sandhu, "Database security - concepts, approaches, and challenges," *IEEE Transactions on Dependable and Secure Computing*, vol. 2, no. 1, pp. 2–19, Mar. 2005, doi: 10.1109/TDSC.2005.9.

[4] Χρήστου Σοφία, "Η «Κ» στο στρατηγείο «Acropolis»: Ασκηση κυβερνοασφάλειας με 1.000 γκουρού," Jun. 23, 2024. [Online]. Available: https://www.kathimerini.gr/society/reportaz/563089456/i-k-sto-stratigeio-acropolis-askisi-kyvernoasfaleias-me-1-000-gkoyroy

[5] "Το Deepfake ως απειλή στον κυβερνοχώρο – Πως μπορεί να αντιμετωπιστεί;," GigiGov Community Portal. [Online]. Available: https://digi.gov.gr/to-deepfake-os-apeili-ston-kyvernochoro-pos-borei-na-antimetopistei/

[6] H. M. Z. A. Shebli and B. D. Beheshti, "A study on penetration testing process and tools," in *2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, 2018, pp. 1–7. doi: 10.1109/LISAT.2018.8378035.

[7] A. Nikiforova, "Data Security as a Top Priority in the Digital World: Preserve Data Value by Being Proactive and Thinking Security First," in *Springer Proceedings in Complexity*, Springer International Publishing, 2023, pp. 3–15. doi: 10.1007/978-3-031-19560-0_1.

[8] D. Litchfield, C. Anley, J. Heasman, and B. Grindlay, *The Database Hacker's Handbook: Defending Database Servers*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2005.

[9] V. Božić, *VULNERABILITIES OF RELATIONAL DATABASES*. 2023. doi: 10.13140/RG.2.2.28051.76322.

[10] N. Joshi, T. Sheth, V. Shah, J. Gupta, and S. Mujawar, "A Detailed Evaluation of SQL Injection Attacks, Detection and Prevention Techniques," in *2022 5th International Conference on Advances in Science and Technology (ICAST)*, 2022, pp. 352–357. doi: 10.1109/ICAST55766.2022.10039662.

[11] Q. Li, W. Li, J. Wang, and M. Cheng, "A SQL Injection Detection Method Based on Adaptive Deep Forest," *IEEE Access*, vol. 7, pp. 145385–145394, 2019, doi: 10.1109/ACCESS.2019.2944951.

[12] J. Tadhani, V. Vekariya, V. Sorathiya, S. Alshathri, and W. El-Shafai, "Securing web applications against XSS and SQLi attacks using a novel deep learning approach," *Scientific Reports*, vol. 14, Jan. 2024, doi: 10.1038/s41598-023-48845-4.

[13] K. Elshazly, Y. Fouad, M. Saleh, and A. Sewisy, "A Survey of SQL Injection Attack Detection and Prevention," *Journal of Computer and Communications*, vol. 02, pp. 1–9, Jan. 2014, doi: 10.4236/jcc.2014.28001.

[14] PenTestical, "SQL Injection - All-in-One Fuzzing Wordlist." Accessed: Aug. 14, 2024. [Online]. Available: https://github.com/PenTestical/sqli

[15] C. Biswal and Dr. S. Pani, "Cyber-Crime Prevention Methodology," 2020, p. 291. doi: 10.1002/9781119711629.ch14.

[16] D. Hartley, "SQL Injection Attacks and Defense," in *SQL Injection Attacks and Defense (Second Edition)*, J. Clarke, Ed., Boston: Syngress, 2012, pp. 1–547. doi: 10.1016/B978-1-59-749963-7.00001-3.

[17] David Kennedy (ReL1K) @HackingDave, *The Social-Engineer Toolkit (SET)*. (2020). TrustedSec. [Online]. Available: https://github.com/trustedsec/social-engineer-toolkit

[18] Abhijit, "Fuzzing: Initial process of buffer overflow." Accessed: Aug. 22, 2024. [Online]. Available: https://abhijit-pal.medium.com/fuzzing-an-initial-process-of-buffer-overflow-e44d6c543eea

[19] ReasonLabs, "What are Stack canaries?" [Online]. Available: https://cyberpedia.reasonlabs.com/EN/stack%20canaries.html

[20] Sharon Shea, “address space layout randomization (ASLR).” [Online]. Available: https://www.techtarget.com/searchsecurity/definition/address-space-layout-randomization-ASLR
[21] IBM, “Address space layout randomization.” [Online]. Available: https://www.ibm.com/docs/en/zos/3.1.0?topic=overview-address-space-layout-randomization
[22] Blue Goat Cyber, “DEP’s Role in Preventing Buffer Overflow Attacks.” [Online]. Available: https://bluegoatcyber.com/blog/deps-role-in-preventing-buffer-overflow-attacks/
[23] “Πώς βοηθά η χρήση ενός Object Relational Mapper (ORM) στον μετριασμό των τρωτών σημείων ένεσης στη συνέχεια; ΚΥΒΕΡΝΑΣΦΑΛΕΙΑ, EITC/IS/WASF ΒΑΣΙΚΕΣ ΑΡΧΕΣ ΑΣΦΑΛΕΙΑΣ ΕΦΑΡΜΟΓΩΝ ΙΣΤΟΥ, ΕΠΙΘΕΣΕΙΣ TLS, ΑΣΦΑΛΕΙΑ ΕΠΙΠΕΔΟΥ ΜΕΤΑΦΟΡΑΣ,” ΕΥΡΩΠΑΪΚΗ ΑΚΑΔΗΜΙΑ ΠΙΣΤΟΠΟΙΗΣΗΣ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΩΝ. Accessed: Jun. 30, 2024. [Online]. Available: https://el.eitca.org/cybersecurity/eitc-is-wasf-web-applications-security-fundamentals/tls-attacks/transport-layer-security/examination-review-transport-layer-security/how-does-using-an-object-relational-mapper-orm-help-mitigate-sequel-injection-vulnerabilities/
[24] Laurenz Albe, “Abusing SECURITY DEFINER functions in PostgreSQL.” [Online]. Available: https://www.cybertec-postgresql.com/en/abusing-security-definer-functions/
[25] N. Anwar and S. Kar, “Review Paper on Various Software Testing Techniques & Strategies,” *Global Journal of Computer Science and Technology*, pp. 43–49, May 2019, doi: 10.34257/GJCSTCVOL19IS2PG43.
[26] Teaganne Finn, Amanda Downie, “What is network penetration testing?,” IBM. [Online]. Available: https://www.ibm.com/topics/network-penetration-testing
[27] X. Lingzi and L. Zhi, “An Overview of Source Code Audit,” in *2015 International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration*, 2015, pp. 26–29. doi: 10.1109/ICIICII.2015.94.
[28] ANADEA, “What is Code Audit: Deep Dive into Software Quality.” Accessed: Aug. 14, 2024. [Online]. Available: https://anadea.info/blog/what-is-code-audit
[29] “The CERT Division,” Carnegie Mellon University - Software Engineering Institute. Accessed: Aug. 12, 2024. [Online]. Available: https://www.sei.cmu.edu/about/divisions/cert/
[30] J. B. Dhruv Mohindra, “Rule 00. Input Validation and Data Sanitization (IDS) - IDS00-J. Prevent SQL injection.” Carnegie Mellon University - Software Engineering Institute, Jun. 11, 2024. Accessed: Aug. 12, 2024. [Online]. Available: https://wiki.sei.cmu.edu/confluence/display/java/IDS00-J.+Prevent+SQL+injection
[31] D. S. Dhruv Mohindra, “Rule 00. Input Validation and Data Sanitization (IDS).” Carnegie Mellon University - Software Engineering Institute, May 01, 2023. Accessed: Aug. 12, 2024. [Online]. Available: https://wiki.sei.cmu.edu/confluence/pages/viewpage.action?pageId=88487865
[32] “Common Weakness Enumeration - CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection’).” [Online]. Available: https://cwe.mitre.org/data/definitions/89.html
[33] OccupyTheWeb, *Linux Basics for Hackers: Getting Started with Networking, Scripting, and Security in Kali7*. 2018.
[34] A. Bacudio, X. Yuan, B. Chu, and M. Jones, “An Overview of Penetration Testing,” *International Journal of Network Security & Its Applications*, vol. 3, pp. 19–38, Nov. 2011, doi: 10.5121/ijnsa.2011.3602.
[35] J. D. Pierce, M. J. Warren, and X. Corray, “A critical review of penetration testing methodologies,” 2004. [Online]. Available: https://api.semanticscholar.org/CorpusID:59667577
[36] D.R. T STEPHENS, *The PenTest Primer: Navigating Your First Penetration Test and Crafting an Impactful Report*. 2023.
[37] “Common Vulnerability Scoring System SIG.” Forum of Incident Response and Security Teams, Inc., 2024 2015. [Online]. Available: https://www.first.org/cvss/
[38] *sqlmap*. [Online]. Available: https://sqlmap.org/
[39] *metasploit*. [Online]. Available: https://www.metasploit.com/
[40] *Nmap: Discover your network*. [Online]. Available: https://nmap.org/
[41] *Burp Suite*. [Online]. Available: https://portswigger.net/burp

[42] “Introduction to Burp Suite, the Tool Dedicated to Web Application Security.” Accessed: Aug. 14, 2024. [Online]. Available: https://www.vaadata.com/blog/introduction-to-burp-suite-the-tool-dedicated-to-web-application-security/
[43] *DbProtect*. [Online]. Available: https://www.trustwave.com/en-us/services/database-security/dbprotect/
[44] *Wireshark*. [Online]. Available: https://www.wireshark.org/
[45] *Nessus*. [Online]. Available: https://www.tenable.com/products/nessus/nessus-professional/evaluate
[46] *ZAP (Zed Attack Proxy)*. Accessed: Aug. 21, 2024. [Online]. Available: https://www.zaproxy.org/
[47] N. Antunes and M. Vieira, “Comparing the Effectiveness of Penetration Testing and Static Code Analysis on the Detection of SQL Injection Vulnerabilities in Web Services,” in *2009 15th IEEE Pacific Rim International Symposium on Dependable Computing*, 2009, pp. 301–306. doi: 10.1109/PRDC.2009.54.
[48] “Code that has a secured login via PHP and SQL and prevents SQL Injection of any level.” Accessed: Sep. 20, 2024. [Online]. Available: https://github.com/keida1412/php-secured-login-with-sql-injection-prevention