ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ – ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Κυβερνοασφάλεια και Επιστήμη Δεδομένων»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	**Ανάπτυξη Βιβλιοθήκης Αυθεντικοποίησης και Εξουσιοδότησης για Java Backend Εφαρμογές** **Development of an Authentication and Authorization Library for Java Backend Applications**
Ονοματεπώνυμο Φοιτητή	**Δημήτριος Θάνος**
Πατρώνυμο	**Κωνσταντίνος**
Αριθμός Μητρώου	**ΜΠΚΕΔ2208**
Επιβλέπων	**Χρήστος Δουληγέρης**

Ημερομηνία Παράδοσης **30 Μαρτίου 2026**

Τριμελής Εξεταστική Επιτροπή

Χρήστος Δουληγέρης Καθηγητής	Παναγιώτης Κοτζανικολάου Καθηγητής	Κωνσταντίνος Πατσάκης Καθηγητής

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να εκφράσω τις ειλικρινείς μου ευχαριστίες στον επιβλέποντα καθηγητή μου, για την καθοδήγηση και υποστήριξη, που μου παρείχε καθ' όλη τη διάρκεια της εκπόνησης της παρούσας διατριβής. Η εμπειρία και οι συμβουλές του ήταν καθοριστικές για την επιτυχή ολοκλήρωση αυτής της εργασίας.

Επίσης, θέλω να ευχαριστήσω την οικογένειά μου για την αμέριστη συμπαράσταση, την υπομονή και την ενθάρρυνση, που μου έδειξαν σε όλη τη διάρκεια των μεταπτυχιακών μου σπουδών.

Τέλος, ευχαριστώ όλους τους συναδέλφους και φίλους, που με στήριξαν με τις ιδέες τους και την ηθική τους υποστήριξη.

Περίληψη

Η παρούσα μεταπτυχιακή διατριβή παρουσιάζει την ανάπτυξη μιας ολοκληρωμένης βιβλιοθήκης αυθεντικοποίησης και εξουσιοδότησης για Java backend εφαρμογές. Η βιβλιοθήκη υλοποιεί σύγχρονες πρακτικές ασφαλείας συμπεριλαμβανομένων της διαχείρισης JWT tokens, της πολυπαραγοντικής αυθεντικοποίησης (MFA) με βάση το πρότυπο TOTP, του ελέγχου πρόσβασης βασισμένου σε ρόλους (RBAC), της σύγχρονης κρυπτογραφίας με αλγόριθμους Ed25519 και ChaCha20-Poly1305, καθώς και ενός ολοκληρωμένου συστήματος καταγραφής συμβάντων ασφαλείας.

Η αρχιτεκτονική της βιβλιοθήκης ακολουθεί τις αρχές SOLID και υλοποιεί το πρότυπο Defense in Depth, παρέχοντας πολλαπλά επίπεδα προστασίας. Το σύστημα περιλαμβάνει προσαρμοστική αυθεντικοποίηση με ανάλυση ρίσκου πραγματικού χρόνου, διαχείριση API keys με scopes και lifecycle management, καθώς και παρακολούθηση απειλών σε πραγματικό χρόνο.

Η υλοποίηση επιτυγχάνει εξαιρετική απόδοση με 50.000 επικυρώσεις JWT ανά δευτερόλεπτο και 100.000 ελέγχους δικαιωμάτων ανά δευτερόλεπτο, ενώ διατηρεί κάλυψη κώδικα άνω του 95%. Η βιβλιοθήκη είναι συμβατή με το OWASP Top 10 και ακολουθεί τις κατευθυντήριες γραμμές του NIST Cybersecurity Framework.

Λέξεις-κλειδιά: Αυθεντικοποίηση, Εξουσιοδότηση, JWT, MFA, RBAC, Κρυπτογραφία, Java, Ασφάλεια Εφαρμογών

Abstract

This master's thesis presents the development of a comprehensive authentication and authorization library for Java backend applications. The library implements modern security practices including JWT token management, multi-factor authentication (MFA) based on the TOTP standard, role-based access control (RBAC), modern cryptography with Ed25519 and ChaCha20-Poly1305 algorithms, and a comprehensive security event logging system.

The library's architecture follows SOLID principles and implements the Defense in Depth pattern, providing multiple layers of protection. The system includes adaptive authentication with real-time risk analysis, API key management with scopes and lifecycle management, as well as real-time threat monitoring.

The implementation achieves excellent performance with 50,000 JWT validations per second and 100,000 permission checks per second, while maintaining code coverage above 95%. The library is OWASP Top 10 compliant and follows NIST Cybersecurity Framework guidelines.

Keywords: Authentication, Authorization, JWT, MFA, RBAC, Cryptography, Java, Application Security

Πίνακας περιεχομένων

ΛΙΣΤΑ ΕΙΚΟΝΩΝ

ΚΕΦΑΛΑΙΟ 1: Εισαγωγή

1.1 Αντικείμενο και Πλαίσιο της Διατριβής

Η παρούσα μεταπτυχιακή διατριβή εστιάζει στην ανάπτυξη μιας ολοκληρωμένης βιβλιοθήκης αυθεντικοποίησης και εξουσιοδότησης για εφαρμογές Java, που λειτουργούν σε περιβάλλον backend. Στο σύγχρονο ψηφιακό κόσμο, όπου οι κυβερνοεπιθέσεις αυξάνονται με γεωμετρική πρόοδο, τόσο σε αριθμό, όσο και σε πολυπλοκότητα, η ασφάλεια των πληροφοριακών συστημάτων έχει αναδειχθεί σ' έναν από τους πλέον κρίσιμους παράγοντες επιτυχίας για κάθε οργανισμό, ανεξαρτήτως μεγέθους ή κλάδου δραστηριοποίησης (Anderson, 2020; Bishop, 2018).

Η ιστορία της ασφάλειας πληροφοριακών συστημάτων ξεκινά από τις πρώτες ημέρες της πληροφορικής, όταν τα συστήματα mainframe της δεκαετίας του 1960 απαιτούσαν απλούς μηχανισμούς αναγνώρισης χρηστών για τον διαμερισμό χρόνου μεταξύ πολλαπλών χρηστών. Ο Fernando Corbató, πρωτοπόρος της επιστήμης υπολογιστών στο MIT, εισήγαγε την έννοια του κωδικού πρόσβασης το 1961 στο Compatible Time-Sharing System, θέτοντας τα θεμέλια για αυτό που θα εξελισσόταν σ' ένα ολόκληρο επιστημονικό πεδίο. Από εκείνη την εποχή, οι μηχανισμοί αυθεντικοποίησης έχουν διανύσει τεράστια απόσταση, εξελισσόμενοι από απλούς κωδικούς σε πολύπλοκα συστήματα, που συνδυάζουν πολλαπλούς παράγοντες επαλήθευσης, βιομετρικά στοιχεία και τεχνητή νοημοσύνη (Bonneau et al., 2012).

Η Java, ως γλώσσα προγραμματισμού, κατέχει ιδιαίτερη θέση στον κόσμο των επιχειρηματικών εφαρμογών. Από την εμφάνισή της το 1995 από την Sun Microsystems, έχει καθιερωθεί ως η κυρίαρχη επιλογή για την ανάπτυξη enterprise συστημάτων, με την αρχή "Write Once, Run Anywhere" ν' αποτελεί το θεμέλιο της φιλοσοφίας της. Σήμερα, σύμφωνα με στατιστικά στοιχεία της βιομηχανίας, η Java χρησιμοποιείται σε περισσότερο από το 35% των επιχειρηματικών εφαρμογών παγκοσμίως, ενώ εκτιμάται ότι υπάρχουν πάνω από 12 εκατομμύρια προγραμματιστές Java ενεργοί σε όλο τον κόσμο. Αυτή η ευρεία υιοθέτηση καθιστά επιτακτική την ανάγκη για αξιόπιστες και ασφαλείς βιβλιοθήκες, που να μπορούν να ενσωματωθούν εύκολα σε υπάρχοντα συστήματα (Bloch, 2018; Oaks, 2020).

Η βιβλιοθήκη, που αναπτύχθηκε στο πλαίσιο της παρούσας διατριβής, αποσκοπεί στην κάλυψη ενός σημαντικού κενού στο οικοσύστημα της Java. Ενώ υπάρχουν διαθέσιμες λύσεις, όπως το Spring Security, αυτές συχνά χαρακτηρίζονται από υψηλή πολυπλοκότητα διαμόρφωσης και μεγάλο αριθμό εξαρτήσεων που καθιστούν δύσκολη την ενσωμάτωσή τους σε ελαφριές εφαρμογές ή microservices. Η προτεινόμενη λύση προσφέρει ένα ισορροπημένο συνδυασμό πληρότητας χαρακτηριστικών και απλότητας χρήσης, επιτρέποντας στους προγραμματιστές να υλοποιήσουν ισχυρούς μηχανισμούς ασφαλείας με ελάχιστο κώδικα διαμόρφωσης.

1.2 Σκοπός και Επιμέρους Στόχοι

Ο κεντρικός σκοπός της παρούσας διατριβής είναι η σχεδίαση, υλοποίηση και αξιολόγηση μιας ολοκληρωμένης βιβλιοθήκης ασφαλείας, που να καλύπτει το πλήρες φάσμα των αναγκών αυθεντικοποίησης και εξουσιοδότησης σύγχρονων Java εφαρμογών. Η βιβλιοθήκη σχεδιάστηκε με γνώμονα τρεις θεμελιώδεις αρχές, που διέπουν κάθε πτυχή της υλοποίησης.

Η πρώτη αρχή αφορά στην ασφάλεια από τον σχεδιασμό. Κάθε συστατικό της βιβλιοθήκης αναπτύχθηκε με την ασφάλεια ως πρωταρχική προτεραιότητα και όχι ως μεταγενέστερη προσθήκη. Αυτό σημαίνει ότι οι προεπιλεγμένες ρυθμίσεις είναι πάντα οι πιο ασφαλείς, ότι τα ευαίσθητα δεδομένα ποτέ δεν αποθηκεύονται σε μορφή απλού κειμένου και ότι κάθε λειτουργία υπόκειται σε αυστηρούς ελέγχους εισόδου, για την αποφυγή επιθέσεων injection και άλλων κοινών ευπαθειών (Howard & Lipner, 2006; McGraw, 2006).

Η δεύτερη αρχή αφορά στην υψηλή απόδοση. Οι μηχανισμοί ασφαλείας συχνά θεωρούνται ως πηγή καθυστερήσεων στα συστήματα, γεγονός που οδηγεί πολλούς προγραμματιστές να τους παρακάμπτουν ή να τους υλοποιούν ατελώς. Η βιβλιοθήκη σχεδιάστηκε ώστε να επιτυγχάνει

ελάχιστο overhead, με τις κρίσιμες λειτουργίες, όπως η επικύρωση JWT tokens, να εκτελούνται σε χρόνο κάτω του χιλιοστού του δευτερολέπτου. Αυτό επιτυγχάνεται μέσω της χρήσης αποδοτικών αλγορίθμων, έξυπνης διαχείρισης μνήμης και στρατηγικής χρήσης caching, όπου είναι κατάλληλο (Oaks, 2020).

Η τρίτη αρχή αφορά στην ευκολία χρήσης και ενσωμάτωσης. Μια βιβλιοθήκη ασφαλείας, όσο ισχυρή και εάν είναι, δεν έχει αξία, εάν οι προγραμματιστές δυσκολεύονται να την χρησιμοποιήσουν σωστά. Για τον λόγο αυτό, η βιβλιοθήκη προσφέρει ένα καθαρό και διαισθητικό API, εκτεταμένη τεκμηρίωση με πρακτικά παραδείγματα και λογικές προεπιλεγμένες τιμές, που επιτρέπουν τη γρήγορη εκκίνηση, χωρίς να θυσιάζεται η ασφάλεια.

Οι επιμέρους στόχοι της διατριβής αναλύονται ως εξής:

Πρώτον, η υλοποίηση ενός πλήρους συστήματος διαχείρισης JWT tokens, που να υποστηρίζει τόσο access, όσο και refresh tokens, με δυνατότητα διαμόρφωσης του χρόνου λήξης, του αλγορίθμου υπογραφής και των custom claims (Jones et al., 2015).

Δεύτερον, η ανάπτυξη ενός συστήματος πολυπαραγοντικής αυθεντικοποίησης βασισμένου στο πρότυπο TOTP, που να είναι συμβατό με δημοφιλείς εφαρμογές, όπως το Google Authenticator και το Microsoft Authenticator (M'Raihi et al., 2011).

Τρίτον, η δημιουργία ενός ευέλικτου μηχανισμού ελέγχου πρόσβασης βασισμένου σε ρόλους που να υποστηρίζει ιεραρχικές δομές και wildcard permissions (Ferraiolo et al., 2001; Sandhu et al., 1996).

Τέταρτον, η ενσωμάτωση σύγχρονων κρυπτογραφικών αλγορίθμων, που προσφέρουν υψηλότερη ασφάλεια και καλύτερη απόδοση σε σύγκριση με παραδοσιακές επιλογές (Bernstein et al., 2012; Nir & Langley, 2018).

Πέμπτον, η ανάπτυξη ενός συστήματος προσαρμοστικής αυθεντικοποίησης, που ν' αναλύει παράγοντες κινδύνου σε πραγματικό χρόνο και να προσαρμόζει δυναμικά τις απαιτήσεις ασφαλείας.

1.3 Συνεισφορά και Καινοτομία της Διατριβής

Η κύρια συνεισφορά της παρούσας διατριβής στο επιστημονικό πεδίο της ασφάλειας λογισμικού έγκειται στη δημιουργία μιας ολοκληρωμένης λύσης, που ενοποιεί πολλαπλές τεχνολογίες ασφαλείας σε μία συνεκτική και εύχρηστη αρχιτεκτονική. Σε αντίθεση με υπάρχουσες προσεγγίσεις, που απαιτούν τον συνδυασμό πολλαπλών ανεξάρτητων βιβλιοθηκών, η προτεινόμενη λύση παρέχει ένα ενιαίο σημείο διαμόρφωσης και ένα συνεπές API για όλες τις λειτουργίες ασφαλείας.

Μία σημαντική καινοτομία της βιβλιοθήκης είναι το σύστημα προσαρμοστικής αυθεντικοποίησης, που αξιολογεί περισσότερους από δεκαπέντε διαφορετικούς παράγοντες κινδύνου για κάθε απόπειρα σύνδεσης. Αυτοί οι παράγοντες περιλαμβάνουν γεωγραφικά δεδομένα, όπως η ανίχνευση νέων ή ύποπτων τοποθεσιών και ο εντοπισμός αδύνατης μετακίνησης, χαρακτηριστικά συσκευής, όπως η αναγνώριση νέων ή ύποπτων συσκευών, χρονικά μοτίβα, όπως η σύνδεση σε ασυνήθιστες ώρες, δικτυακούς παράγοντες, όπως η χρήση Tor, VPN ή proxy servers, και συμπεριφορικά στοιχεία, όπως η ανάλυση του ρυθμού πληκτρολόγησης και των μοτίβων πλοήγησης. Βάσει αυτής της ανάλυσης, το σύστημα μπορεί να απαιτήσει πρόσθετους παράγοντες αυθεντικοποίησης ή ακόμα και να αποκλείσει εντελώς μια ύποπτη απόπειρα σύνδεσης.

Μία ακόμη καινοτομία αφορά την ενσωμάτωση σύγχρονων κρυπτογραφικών πρωτοκόλλων που σπάνια συναντώνται σε βιβλιοθήκες αυθεντικοποίησης γενικής χρήσης. Συγκεκριμένα, η βιβλιοθήκη υποστηρίζει ψηφιακές υπογραφές Ed25519, έναν αλγόριθμο, που βασίζεται σε ελλειπτικές καμπύλες και προσφέρει σημαντικά πλεονεκτήματα απόδοσης σε σύγκριση με τον παραδοσιακό RSA, διατηρώντας παράλληλα ισοδύναμο ή υψηλότερο επίπεδο ασφάλειας. Επίσης, υποστηρίζει κρυπτογράφηση ChaCha20-Poly1305, έναν αλγόριθμο authenticated encryption, που χρησιμοποιείται ευρέως σε σύγχρονα πρωτόκολλα, όπως το TLS 1.3 και το

WireGuard VPN και ο οποίος προσφέρει εξαιρετική απόδοση ακόμα και σε συστήματα χωρίς hardware acceleration για AES (Bernstein et al., 2012; Nir & Langley, 2018; Rescorla, 2018).

Η διατριβή συνεισφέρει επίσης στον ακαδημαϊκό χώρο παρέχοντας ένα πλήρως τεκμηριωμένο παράδειγμα υλοποίησης σύγχρονων πρακτικών ασφαλείας. Ο κώδικας της βιβλιοθήκης, που ξεπερνά τις εννέα χιλιάδες γραμμές, συνοδεύεται από σχεδόν τρεις χιλιάδες γραμμές κώδικα δοκιμών και εκτεταμένη τεκμηρίωση, με αποτέλεσμα να καθίσταται ιδανικός για εκπαιδευτική χρήση και περαιτέρω έρευνα.

1.4 Δομή της Διατριβής

Η παρούσα διατριβή οργανώνεται σε έξι κεφάλαια, που καλύπτουν το πλήρες εύρος της εργασίας από το θεωρητικό υπόβαθρο έως την αξιολόγηση και τα συμπεράσματα.

Το παρόν πρώτο κεφάλαιο παρουσιάζει το πλαίσιο και τα κίνητρα της διατριβής, θέτει τους στόχους και περιγράφει τη συνεισφορά της εργασίας στο επιστημονικό πεδίο. Επιπλέον, παρέχει μια σύντομη επισκόπηση της δομής του υπόλοιπου κειμένου, ώστε ο αναγνώστης να μπορεί να πλοηγηθεί αποτελεσματικά στο περιεχόμενο.

Το δεύτερο κεφάλαιο εμβαθύνει στο θεωρητικό υπόβαθρο, που είναι απαραίτητο για την κατανόηση της βιβλιοθήκης. Αναλύει τις θεμελιώδεις έννοιες της αυθεντικοποίησης και εξουσιοδότησης, παρουσιάζει τα πρότυπα JWT και TOTP, εξηγεί τις αρχές του ελέγχου πρόσβασης βασισμένου σε ρόλους και εισάγει τους σύγχρονους κρυπτογραφικούς αλγορίθμους, που χρησιμοποιούνται στην υλοποίηση. Επίσης, παρουσιάζει τα διεθνή πρότυπα και πλαίσια ασφαλείας όπως το OWASP Top 10 και το NIST Cybersecurity Framework, που καθοδήγησαν τη σχεδίαση.

Το τρίτο κεφάλαιο περιγράφει τη μεθοδολογία, που ακολουθήθηκε κατά την ανάπτυξη της βιβλιοθήκης. Αναλύει τις αρχιτεκτονικές αποφάσεις, τα σχεδιαστικά πρότυπα που εφαρμόστηκαν, τις τεχνολογίες και τα εργαλεία που χρησιμοποιήθηκαν, καθώς και τη στρατηγική δοκιμών που υιοθετήθηκε για τη διασφάλιση της ποιότητας.

Το τέταρτο κεφάλαιο αποτελεί τον πυρήνα της τεχνικής τεκμηρίωσης και παρουσιάζει αναλυτικά την υλοποίηση κάθε συστατικού της βιβλιοθήκης. Περιγράφει τη διαχείριση JWT tokens, τον μηχανισμό πολυπαραγοντικής αυθεντικοποίησης, το σύστημα εξουσιοδότησης, τις κρυπτογραφικές υπηρεσίες, τη διαχείριση API keys, το σύστημα καταγραφής συμβάντων ασφαλείας και τον μηχανισμό προσαρμοστικής αυθεντικοποίησης.

Το πέμπτο κεφάλαιο παρουσιάζει τις δοκιμές και την αξιολόγηση της βιβλιοθήκης. Περιλαμβάνει αποτελέσματα από unit tests, integration tests, security tests και performance tests. Επίσης, αξιολογεί τη συμμόρφωση της βιβλιοθήκης με τα πρότυπα OWASP Top 10 και NIST και συγκρίνει την απόδοσή της με άλλες διαθέσιμες λύσεις.

Τέλος, το έκτο κεφάλαιο συνοψίζει τα αποτελέσματα της διατριβής, αξιολογεί την επίτευξη των στόχων, συζητά τους περιορισμούς της υλοποίησης και προτείνει κατευθύνσεις για μελλοντική έρευνα και επέκταση της βιβλιοθήκης.

1.5 Κίνητρο και Επικαιρότητα του Θέματος

Το κίνητρο για την εκπόνηση της παρούσας διατριβής πηγάζει από την αυξανόμενη σημασία της κυβερνοασφάλειας στη σύγχρονη ψηφιακή εποχή και την ανάγκη για προσβάσιμες, αλλά ταυτόχρονα ισχυρές λύσεις ασφαλείας. Τα στατιστικά στοιχεία, που αφορούν τις κυβερνοεπιθέσεις είναι ανησυχητικά και καταδεικνύουν την επιτακτική ανάγκη για βελτίωση των μηχανισμών προστασίας.

Σύμφωνα με την ετήσια έκθεση της IBM για το κόστος παραβιάσεων δεδομένων το 2024, το μέσο κόστος μιας παραβίασης έχει φτάσει τα 4,45 εκατομμύρια δολάρια παγκοσμίως, σημειώνοντας αύξηση 15% τα τελευταία τρία έτη. Ιδιαίτερα ανησυχητικό είναι το εύρημα ότι το 74% όλων των παραβιάσεων περιλαμβάνει τον ανθρώπινο παράγοντα, είτε μέσω κλεμμένων

διαπιστευτηρίων, είτε μέσω κοινωνικής μηχανικής, είτε μέσω ανθρώπινου λάθους. Αυτό υπογραμμίζει τη σημασία της ισχυρής αυθεντικοποίησης ως πρώτης γραμμής άμυνας.

Η Microsoft, σε μελέτη που δημοσίευσε το 2019 και επιβεβαίωσε σε μεταγενέστερες ενημερώσεις, διαπίστωσε ότι η χρήση πολυπαραγοντικής αυθεντικοποίησης μπορεί να αποτρέψει το 99,9% των επιθέσεων κατά λογαριασμών. Παρά την ύπαρξη αυτού του εντυπωσιακού ποσοστού, η υιοθέτηση του MFA παραμένει χαμηλή σε πολλούς οργανισμούς, κυρίως λόγω της αντιλαμβανόμενης πολυπλοκότητας υλοποίησης και της αντίστασης των χρηστών σε πρόσθετα βήματα σύνδεσης.

Η επιλογή της Java ως πλατφόρμας για την υλοποίηση δεν είναι τυχαία. Παρά την εμφάνιση νεότερων γλωσσών και πλαισίων, η Java παραμένει η κυρίαρχη επιλογή για enterprise εφαρμογές λόγω της ωριμότητας του οικοσυστήματός της, της ευρείας υποστήριξης από την κοινότητα και τη βιομηχανία και της αποδεδειγμένης αξιοπιστίας σε κρίσιμες εφαρμογές. Η έκδοση Java 21, που χρησιμοποιήθηκε στην υλοποίηση, αποτελεί Long Term Support έκδοση και εισάγει σημαντικές βελτιώσεις απόδοσης, καθώς και νέα χαρακτηριστικά γλώσσας, που διευκολύνουν τη γραφή ασφαλούς κώδικα.

Η επικαιρότητα του θέματος ενισχύεται περαιτέρω από τις πρόσφατες κανονιστικές εξελίξεις σε παγκόσμιο επίπεδο. Ο Γενικός Κανονισμός για την Προστασία Δεδομένων της Ευρωπαϊκής Ένωσης επιβάλλει αυστηρές απαιτήσεις για την προστασία προσωπικών δεδομένων, με πρόστιμα που μπορούν να φτάσουν το 4% του παγκόσμιου ετήσιου τζίρου μιας εταιρείας. Παρόμοιες νομοθετικές πρωτοβουλίες έχουν αναληφθεί σε πολλές χώρες, δημιουργώντας ένα περιβάλλον όπου η ισχυρή ασφάλεια δεν είναι πλέον προαιρετική, αλλά νομική υποχρέωση (European Union, 2016).

Η παρούσα διατριβή αντιμετωπίζει αυτές τις προκλήσεις, παρέχοντας μια λύση, που συνδυάζει υψηλό επίπεδο ασφάλειας με ευκολία υλοποίησης, επιτρέποντας ακόμα και σε οργανισμούς με περιορισμένους πόρους να υιοθετήσουν σύγχρονες πρακτικές ασφαλείας. Η προσέγγιση "secure by default" διασφαλίζει ότι ακόμα και οι λιγότερο έμπειροι προγραμματιστές μπορούν να δημιουργήσουν ασφαλείς εφαρμογές, μειώνοντας έτσι τη συνολική επιφάνεια έκθεσης σε κυβερνοαπειλές.

ΚΕΦΑΛΑΙΟ 2: Θεωρητικό Υπόβαθρο

2.1 Θεμελιώδεις Έννοιες Αυθεντικοποίησης και Εξουσιοδότησης

Η κατανόηση της διαφοράς μεταξύ αυθεντικοποίησης και εξουσιοδότησης αποτελεί θεμελιώδες προαπαιτούμενο για τη σχεδίαση ασφαλών συστημάτων. Αν και οι δύο όροι συχνά χρησιμοποιούνται εναλλακτικά στην καθημερινή ομιλία, αναφέρονται σε εντελώς διαφορετικές διαδικασίες, που πρέπει να υλοποιούνται ξεχωριστά και να συνεργάζονται αρμονικά (Bishop, 2018; Pfleeger et al., 2015).

Η αυθεντικοποίηση είναι η διαδικασία επαλήθευσης της ταυτότητας ενός χρήστη ή συστήματος. Απαντά στο ερώτημα "Ποιος είσαι;" και αποτελεί το πρώτο βήμα σε οποιαδήποτε αλληλεπίδραση με ένα ασφαλές σύστημα. Η ιστορία της αυθεντικοποίησης ξεκινάει από τους αρχαίους χρόνους, όταν οι φρουροί χρησιμοποιούσαν συνθήματα για να αναγνωρίζουν φίλιες δυνάμεις και έχει εξελιχθεί δραματικά με την πάροδο των αιώνων (Anderson, 2020).

Στον ψηφιακό κόσμο, η αυθεντικοποίηση βασίζεται παραδοσιακά σε τρεις κατηγορίες παραγόντων, που είναι ευρέως γνωστές ως τα “τρία κάτι”. Ο πρώτος παράγοντας είναι κάτι που γνωρίζει ο χρήστης, όπως ένας κωδικός πρόσβασης, ένα PIN, ή η απάντηση σε μια μυστική ερώτηση. Αυτή είναι η παλαιότερη και πιο διαδεδομένη μορφή αυθεντικοποίησης, αλλά και η πιο ευάλωτη, καθώς οι κωδικοί μπορούν να κλαπούν, ν’ αποκαλυφθούν μέσω κοινωνικής μηχανικής ή ακόμα και να τους μαντέψουν. Ο δεύτερος παράγοντας είναι κάτι που κατέχει ο χρήστης, όπως ένα κινητό τηλέφωνο, μια έξυπνη κάρτα, ή ένα hardware token. Αυτός ο παράγοντας προσθέτει ένα επιπλέον επίπεδο ασφάλειας, καθώς ένας επιτιθέμενος θα πρέπει να αποκτήσει φυσική πρόσβαση στη συσκευή. Ο τρίτος παράγοντας είναι κάτι που είναι ο χρήστης, δηλαδή βιομετρικά χαρακτηριστικά, όπως το δακτυλικό αποτύπωμα, η αναγνώριση προσώπου, η σάρωση ίριδας ή ακόμα και η φωνητική αναγνώριση. Τα βιομετρικά στοιχεία θεωρούνται ιδιαίτερα ασφαλή καθώς είναι μοναδικά για κάθε άτομο και δύσκολο να αντιγραφούν (Grassi et al., 2017; NIST, 2020).

Η πολυπαραγοντική αυθεντικοποίηση, γνωστή και ως MFA, απαιτεί την επαλήθευση δύο ή περισσότερων παραγόντων από διαφορετικές κατηγορίες. Είναι σημαντικό να σημειωθεί ότι η χρήση δύο παραγόντων από την ίδια κατηγορία, όπως δύο διαφορετικοί κωδικοί, δεν αποτελεί πραγματική πολυπαραγοντική αυθεντικοποίηση. Η δύναμη του MFA έγκειται στο γεγονός ότι ένας επιτιθέμενος θα πρέπει να παραβιάσει πολλαπλές ανεξάρτητες γραμμές άμυνας για να αποκτήσει πρόσβαση (Grassi et al., 2017; NIST, 2020).

Η εξουσιοδότηση, από την άλλη πλευρά, είναι η διαδικασία καθορισμού των δικαιωμάτων πρόσβασης ενός ήδη αυθεντικοποιημένου χρήστη. Απαντά στο ερώτημα "Τι επιτρέπεται να κάνεις;" και εφαρμόζεται μετά την επιτυχή αυθεντικοποίηση. Η εξουσιοδότηση καθορίζει ποιους πόρους μπορεί να προσπελάσει ένας χρήστης και ποιες ενέργειες μπορεί να εκτελέσει σε αυτούς τους πόρους (Anderson, 2020; Pfleeger et al., 2015).

Η σχέση μεταξύ αυθεντικοποίησης και εξουσιοδότησης μπορεί να κατανοηθεί καλύτερα μέσω ενός απλού παραδείγματος. Όταν ένας υπάλληλος εισέρχεται σε ένα κτίριο γραφείων, χρησιμοποιώντας την κάρτα εισόδου του, το σύστημα ελέγχου πρόσβασης πρώτα επαληθεύει την ταυτότητά του, γεγονός που αποτελεί αυθεντικοποίηση και στη συνέχεια ελέγχει εάν έχει δικαίωμα πρόσβασης στον συγκεκριμένο όροφο ή χώρο, γεγονός που αποτελεί εξουσιοδότηση. Ένας υπάλληλος μπορεί να έχει έγκυρη ταυτότητα, αλλά να μην επιτρέπεται να εισέλθει σε ορισμένους περιορισμένους χώρους.

2.2 JSON Web Tokens: Θεωρία και Λειτουργία

Τα JSON Web Tokens, γνωστά με το ακρωνύμιο JWT, αποτελούν ένα ανοιχτό πρότυπο που ορίζεται στο RFC 7519 και έχει καθιερωθεί ως η κυρίαρχη μέθοδος για την ασφαλή μετάδοση πληροφοριών μεταξύ δύο μερών σε μορφή αντικειμένου JSON. Η ιστορία των JWT ξεκινά το

2010, όταν η ανάγκη για ένα συμπαγές και αυτοτελές μέσο αυθεντικοποίησης οδήγησε στην ανάπτυξη αυτού του προτύπου από μια ομάδα μηχανικών υπό την αιγίδα του IETF (Jones et al., 2015).

Η δομή ενός JWT αποτελείται από τρία διακριτά μέρη που διαχωρίζονται με τελείες. Το πρώτο μέρος είναι η επικεφαλίδα ή header, η οποία περιέχει μεταδεδομένα για το token. Τυπικά, η επικεφαλίδα περιλαμβάνει τον τύπο του token, που είναι πάντα JWT και τον αλγόριθμο, που χρησιμοποιείται για την υπογραφή, όπως HS256 για HMAC με SHA-256 ή RS256 για RSA με SHA-256. Αυτές οι πληροφορίες κωδικοποιούνται σε Base64URL, μια παραλλαγή του Base64, που είναι ασφαλής για χρήση σε URLs (Jones et al., 2015).

Το δεύτερο μέρος είναι το φορτίο ή payload, το οποίο περιέχει τα claims, δηλαδή τις δηλώσεις σχετικά με τον χρήστη ή την οντότητα, που αντιπροσωπεύει το token. Το πρότυπο ορίζει τρεις κατηγορίες claims. Τα registered claims είναι προκαθορισμένα claims, που δεν είναι υποχρεωτικά, αλλά συνιστώνται, όπως το iss για τον εκδότη, το exp για τον χρόνο λήξης, το sub για το υποκείμενο και το aud για το κοινό. Τα public claims μπορούν να οριστούν ελεύθερα, αλλά πρέπει να καταχωρηθούν στο IANA JSON Web Token Registry για την αποφυγή συγκρούσεων. Τα private claims είναι custom claims, που δημιουργούνται για την ανταλλαγή πληροφοριών μεταξύ μερών, που έχουν συμφωνήσει στη χρήση τους (Jones et al., 2015).

Το τρίτο μέρος είναι η υπογραφή ή signature, η οποία δημιουργείται συνδυάζοντας την κωδικοποιημένη επικεφαλίδα, το κωδικοποιημένο φορτίο, ένα μυστικό κλειδί και τον αλγόριθμο, που καθορίζεται στην επικεφαλίδα. Η υπογραφή εξασφαλίζει ότι το token δεν έχει τροποποιηθεί κατά τη μεταφορά του και ότι προέρχεται πράγματι από τον αναμενόμενο εκδότη (Jones et al., 2015).

Τα πλεονεκτήματα των JWT έναντι παραδοσιακών μεθόδων αυθεντικοποίησης βασισμένων σε sessions είναι σημαντικά. Πρώτον, τα JWT είναι stateless, που σημαίνει ότι ο server δεν χρειάζεται να αποθηκεύει πληροφορίες session, καθιστώντας την κλιμάκωση πολύ πιο εύκολη. Δεύτερον, τα JWT είναι αυτοτελή, περιέχοντας όλες τις απαραίτητες πληροφορίες για την αυθεντικοποίηση του χρήστη χωρίς ανάγκη πρόσβασης σε βάση δεδομένων. Τρίτον, τα JWT λειτουργούν άψογα σε περιβάλλοντα cross-domain και είναι ιδανικά για αρχιτεκτονικές microservices, όπου πολλαπλές υπηρεσίες πρέπει να επαληθεύσουν την ταυτότητα του χρήστη. Τέταρτον, τα JWT είναι ιδιαίτερα κατάλληλα για mobile εφαρμογές, όπου η διατήρηση cookies μπορεί να είναι προβληματική (Jones et al., 2015; Newman, 2021).

Ωστόσο, τα JWT έχουν και ορισμένες προκλήσεις που πρέπει να αντιμετωπιστούν. Η σημαντικότερη αφορά στην ανάκληση tokens. Επειδή τα JWT είναι stateless, δεν υπάρχει εύκολος τρόπος ν' ακυρωθεί ένα token πριν τη λήξη του. Αυτό μπορεί ν' αντιμετωπιστεί με τη χρήση μαύρης λίστας tokens ή με τη χρήση σύντομων χρόνων λήξης σε συνδυασμό με refresh tokens (Jones et al., 2015).

2.3 Πρότυπο TOTP για Πολυπαραγοντική Αυθεντικοποίηση

Το Time-based One-Time Password, γνωστό ως TOTP, είναι ένας αλγόριθμος, που παράγει κωδικούς μιας χρήσης βασισμένους στον τρέχοντα χρόνο. Ορίζεται στο RFC 6238 και αποτελεί επέκταση του αλγορίθμου HOTP, που ορίζεται στο RFC 4226. Η βασική ιδέα πίσω από το TOTP είναι απλή αλλά κομψή: τόσο ο server, όσο και η συσκευή του χρήστη μοιράζονται ένα μυστικό κλειδί και χρησιμοποιούν τον τρέχοντα χρόνο για να υπολογίσουν τον ίδιο κωδικό ανεξάρτητα (M'Raihi et al., 2011).

Η λειτουργία του TOTP βασίζεται σε τρία βασικά στοιχεία. Το πρώτο είναι το κοινό μυστικό κλειδί, το οποίο δημιουργείται κατά την εγγραφή του χρήστη στο MFA και αποθηκεύεται τόσο στον server, όσο και στη συσκευή του χρήστη. Αυτό το κλειδί τυπικά κωδικοποιείται σε Base32 για ευκολία μεταφοράς και εμφάνισης. Το δεύτερο στοιχείο είναι ο τρέχων χρόνος, ο οποίος μετριέται σε χρονικές περιόδους. Η προεπιλεγμένη περίοδος είναι 30 δευτερόλεπτα, που σημαίνει ότι ένας νέος κωδικός παράγεται κάθε μισό λεπτό. Το τρίτο στοιχείο είναι η συνάρτηση

κατακερματισμού, τυπικά HMAC-SHA1, που χρησιμοποιείται για τη δημιουργία του κωδικού (M'Raihi et al., 2011).

Ο αλγόριθμος υπολογίζει πρώτα τον αριθμό χρονικών περιόδων που έχουν περάσει από μια αφετηρία, τυπικά την 1η Ιανουαρίου 1970. Στη συνέχεια, χρησιμοποιεί αυτόν τον αριθμό μαζί με το μυστικό κλειδί για να υπολογίσει έναν κατακερματισμό HMAC. Τέλος, από αυτόν τον κατακερματισμό εξάγεται ένας αριθμητικός κωδικός, συνήθως 6 ψηφίων (M'Raihi et al., 2011).

Η ενσωμάτωση του TOTP σε εφαρμογές γίνεται τυπικά μέσω QR codes. Κατά την εγγραφή, ο server παράγει ένα μυστικό κλειδί και το κωδικοποιεί σε ένα URI με τη μορφή otpauth, που μπορεί να αναπαρασταθεί ως QR code. Ο χρήστης σαρώνει αυτό το QR code με μια εφαρμογή authenticator, όπως το Google Authenticator ή το Microsoft Authenticator, και από εκείνη τη στιγμή η εφαρμογή μπορεί να παράγει έγκυρους κωδικούς (M'Raihi et al., 2011).

Για τη διασφάλιση της ευρωστίας του συστήματος, είναι σημαντικό να υπάρχει ένας μηχανισμός ανάκτησης σε περίπτωση που ο χρήστης χάσει πρόσβαση στη συσκευή αυθεντικοποίησης. Αυτό συνήθως επιτυγχάνεται μέσω backup codes, δηλαδή μιας σειράς κωδικών μιας χρήσης, που δημιουργούνται κατά την εγγραφή και πρέπει να αποθηκευτούν σε ασφαλές μέρος από τον χρήστη (M'Raihi et al., 2011).

2.4 Έλεγχος Πρόσβασης Βασισμένος σε Ρόλους

Ο έλεγχος πρόσβασης βασισμένος σε ρόλους, γνωστός ως RBAC από τα αρχικά του αγγλικού Role-Based Access Control, είναι μια προσέγγιση για τον περιορισμό της πρόσβασης σε συστήματα που βασίζεται στους ρόλους των χρηστών εντός ενός οργανισμού. Η θεωρητική θεμελίωση του RBAC χρονολογείται στη δεκαετία του 1990, με τη σημαντική εργασία των David Ferraiolo και Richard Kuhn το 1992 (Sandhu et al., 1996; Ferraiolo et al., 2001) που έθεσαν τις βάσεις για την τυποποίηση του μοντέλου.

Το μοντέλο RBAC βασίζεται σε τέσσερα θεμελιώδη στοιχεία. Οι χρήστες είναι οι ανθρώπινες οντότητες, που αλληλοεπιδρούν με το σύστημα. Οι ρόλοι αντιπροσωπεύουν λειτουργικές θέσεις ή αρμοδιότητες εντός του οργανισμού, όπως διαχειριστής, συντάκτης, ή αναγνώστης. Τα δικαιώματα ή permissions είναι οι εγκρίσεις για την εκτέλεση συγκεκριμένων ενεργειών σε συγκεκριμένους πόρους. Τέλος, οι συνεδρίες ή sessions αντιπροσωπεύουν τη σύνδεση ενός χρήστη με το σύστημα και τους ρόλους που έχει ενεργοποιήσει (Ferraiolo et al., 2001; Sandhu et al., 1996).

Η σχέση μεταξύ αυτών των στοιχείων είναι ιεραρχική. Οι ρόλοι ανατίθενται σε χρήστες και τα δικαιώματα ανατίθενται σε ρόλους. Αυτή η έμμεση σύνδεση προσφέρει σημαντικά πλεονεκτήματα διαχείρισης. Αντί να χρειάζεται ο διαχειριστής να ορίσει δικαιώματα για κάθε χρήστη ξεχωριστά, αρκεί ν’ αναθέσει στον χρήστη τον κατάλληλο ρόλο. Όταν οι απαιτήσεις αλλάζουν, η τροποποίηση του ρόλου επηρεάζει αυτόματα όλους τους χρήστες, που έχουν αυτόν τον ρόλο (Ferraiolo et al., 2001).

Το πρότυπο RBAC ορίζει τέσσερα επίπεδα πολυπλοκότητας. Το RBAC0, γνωστό ως flat RBAC, είναι το βασικό μοντέλο, που περιλαμβάνει μόνο χρήστες, ρόλους και δικαιώματα. Το RBAC1, γνωστό ως hierarchical RBAC, προσθέτει την έννοια της ιεραρχίας ρόλων, όπου ένας ρόλος μπορεί να κληρονομήσει τα δικαιώματα ενός άλλου ρόλου. Το RBAC2, γνωστό ως constrained RBAC, εισάγει περιορισμούς, όπως ο διαχωρισμός καθηκόντων και οι περιορισμοί αμοιβαίου αποκλεισμού. Τέλος, το RBAC3 συνδυάζει τα RBAC1 και RBAC2 παρέχοντας τόσο ιεραρχία όσο και περιορισμούς (Sandhu et al., 1996; Ferraiolo et al., 2001).

Η υλοποίηση RBAC στη βιβλιοθήκη που αναπτύχθηκε ακολουθεί το μοντέλο RBAC1, υποστηρίζοντας ιεραρχικούς ρόλους, ενώ διατηρεί την απλότητα της διαμόρφωσης. Επιπλέον, υποστηρίζει wildcards στα δικαιώματα, επιτρέποντας τον ορισμό δικαιωμάτων όπως admin:* που παρέχει πρόσβαση σε όλες τις διαχειριστικές λειτουργίες.

2.5 Σύγχρονη Κρυπτογραφία: Ed25519 και ChaCha20-Poly1305

Η κρυπτογραφία αποτελεί τη θεμελιώδη τεχνολογία πίσω από κάθε ασφαλές σύστημα. Η ιστορία της χρονολογείται χιλιάδες χρόνια πριν, με τη κρυπτογραφία του Καίσαρα και τη μηχανή Enigma ν' αποτελούν μερικά από τα πιο γνωστά παραδείγματα. Στη σύγχρονη εποχή, η κρυπτογραφία έχει εξελιχθεί σε μια αυστηρή μαθηματική επιστήμη, που βασίζεται σε πολύπλοκα μαθηματικά προβλήματα (Stallings, 2017; Katz & Lindell, 2020).

Ο αλγόριθμος Ed25519 αποτελεί μια σύγχρονη προσέγγιση στις ψηφιακές υπογραφές, που βασίζεται στην ελλειπτική καμπύλη Curve25519. Σχεδιάστηκε από τον διάσημο κρυπτογράφο Daniel J. Bernstein και τους συνεργάτες του (Bernstein et al., 2012; Josefsson & Liusvaara, 2017), αποσκοπώντας στην αντικατάσταση παλαιότερων αλγόριθμων, όπως ο RSA και το DSA. Ο Ed25519 χρησιμοποιεί μια ειδική μορφή ελλειπτικής καμπύλης, γνωστή ως twisted Edwards curve, η οποία επιτρέπει ιδιαίτερα αποδοτικές υλοποιήσεις.

Τα πλεονεκτήματα του Ed25519 είναι πολλαπλά. Πρώτον, προσφέρει εξαιρετική απόδοση και είναι σημαντικά ταχύτερος τόσο στη δημιουργία, όσο και στην επαλήθευση υπογραφών σε σύγκριση με τον RSA. Δεύτερον, χρησιμοποιεί μικρά κλειδιά και υπογραφές. Τα ιδιωτικά κλειδιά είναι μόλις 32 bytes, τα δημόσια κλειδιά είναι επίσης 32 bytes, και οι υπογραφές είναι 64 bytes. Αυτό το μικρό μέγεθος είναι ιδανικό για εφαρμογές με περιορισμένο εύρος ζώνης ή αποθηκευτικό χώρο. Τρίτον, ο Ed25519 είναι ανθεκτικός σε επιθέσεις πλευρικού καναλιού, καθώς σχεδιάστηκε από την αρχή να εκτελείται σε σταθερό χρόνο, αποτρέποντας επιθέσεις που βασίζονται στη μέτρηση χρόνου εκτέλεσης (Bernstein et al., 2012).

Ο αλγόριθμος ChaCha20-Poly1305 είναι ένας αλγόριθμος authenticated encryption, που συνδυάζει τον stream cipher ChaCha20 για κρυπτογράφηση και τον Poly1305 message authentication code για επαλήθευση ακεραιότητας. Αυτός ο συνδυασμός, γνωστός και ως AEAD για Authenticated Encryption with Associated Data, παρέχει ταυτόχρονα εμπιστευτικότητα και ακεραιότητα των δεδομένων (Nir & Langley, 2018).

Ο ChaCha20 σχεδιάστηκε επίσης από τον Daniel Bernstein ως εναλλακτική του AES για περιπτώσεις όπου το hardware acceleration δεν είναι διαθέσιμο. Ενώ ο AES είναι εξαιρετικά γρήγορος σε επεξεργαστές με ειδικές εντολές AES-NI, ο ChaCha20 υπερέχει σε καθαρές software υλοποιήσεις. Αυτό τον καθιστά ιδανικό για mobile συσκευές και embedded συστήματα. Ο ChaCha20-Poly1305 χρησιμοποιείται ευρέως σε σύγχρονα πρωτόκολλα ασφαλείας, συμπεριλαμβανομένου του TLS 1.3, του QUIC protocol της Google, και του WireGuard VPN (Nir & Langley, 2018; Rescorla, 2018).

2.6 Αρχή Defense in Depth

Η αρχή Defense in Depth, που θα μπορούσε να μεταφραστεί ως άμυνα σε βάθος, είναι μια στρατηγική ασφαλείας, που δανείζεται την ιδέα της από τη στρατιωτική τακτική. Η βασική ιδέα είναι ότι κανένα μεμονωμένο μέτρο ασφαλείας δεν είναι αλάνθαστο και επομένως πρέπει να υπάρχουν πολλαπλά επίπεδα προστασίας. Αν ένα επίπεδο αποτύχει ή παρακαμφθεί, τα υπόλοιπα επίπεδα εξακολουθούν να παρέχουν προστασία (Anderson, 2020; Pfleeger et al., 2015).

Στο πλαίσιο της ασφάλειας πληροφοριακών συστημάτων, η αρχή Defense in Depth υλοποιείται μέσω της στρωματοποίησης διαφορετικών μηχανισμών ασφαλείας. Το πρώτο επίπεδο είναι η περιμετρική ασφάλεια, που περιλαμβάνει firewalls, VPNs και ελέγχους πρόσβασης δικτύου. Το δεύτερο επίπεδο είναι η αυθεντικοποίηση, που διασφαλίζει ότι μόνο εξουσιοδοτημένοι χρήστες μπορούν να προσπελάσουν το σύστημα. Το τρίτο επίπεδο είναι η εξουσιοδότηση, που περιορίζει τις ενέργειες που μπορούν να εκτελέσουν οι αυθεντικοποιημένοι χρήστες. Το τέταρτο επίπεδο είναι η κρυπτογράφηση δεδομένων τόσο κατά τη μεταφορά, όσο και κατά την αποθήκευση. Το πέμπτο επίπεδο είναι η καταγραφή και παρακολούθηση, που επιτρέπει την ανίχνευση ύποπτων δραστηριοτήτων. Το έκτο επίπεδο είναι η αντίδραση σε συμβάντα, που

καθορίζει πώς θα αντιμετωπιστούν οι παραβιάσεις όταν συμβούν (Anderson, 2020; Bishop, 2018).

Η βιβλιοθήκη που αναπτύχθηκε υλοποιεί την αρχή Defense in Depth παρέχοντας μηχανισμούς για πολλά από αυτά τα επίπεδα. Η αυθεντικοποίηση υποστηρίζεται μέσω κωδικών πρόσβασης με ισχυρό hashing και προαιρετικό MFA. Η εξουσιοδότηση υποστηρίζεται μέσω RBAC και resource-based permissions. Η κρυπτογράφηση υποστηρίζεται μέσω Ed25519 και ChaCha20-Poly1305. Η καταγραφή υποστηρίζεται μέσω του comprehensive audit logging system. Τέλος, η ανίχνευση απειλών υποστηρίζεται μέσω του real-time security monitoring.

2.7 Πρότυπα OWASP και NIST

Ο οργανισμός OWASP, ακρωνύμιο του Open Web Application Security Project, είναι ένας μη κερδοσκοπικός οργανισμός που έχει ως αποστολή τη βελτίωση της ασφάλειας του λογισμικού. Η λίστα OWASP Top 10 αποτελεί το πιο γνωστό έργο του οργανισμού και παρουσιάζει τις δέκα πιο κρίσιμες κατηγορίες ευπαθειών web εφαρμογών. Η λίστα ενημερώνεται περιοδικά με βάση δεδομένα από πραγματικές παραβιάσεις και αποτελεί σημείο αναφοράς για προγραμματιστές και ελεγκτές ασφαλείας (OWASP Foundation, 2021).

Η έκδοση 2021 της λίστας περιλαμβάνει κατηγορίες, που σχετίζονται άμεσα με το αντικείμενο της παρούσας διατριβής. Η κατηγορία A01 Broken Access Control αναφέρεται σε αδυναμίες στον έλεγχο πρόσβασης, που επιτρέπουν σε χρήστες να ενεργούν εκτός των προβλεπόμενων δικαιωμάτων τους. Η κατηγορία A02 Cryptographic Failures αναφέρεται σε αποτυχίες στην κρυπτογραφία, που οδηγούν σε έκθεση ευαίσθητων δεδομένων. Η κατηγορία A07 Identification and Authentication Failures αναφέρεται σε αδυναμίες στους μηχανισμούς αυθεντικοποίησης. Η κατηγορία A09 Security Logging and Monitoring Failures αναφέρεται σε ανεπάρκειες στην καταγραφή συμβάντων ασφαλείας (OWASP Foundation, 2021).

Το NIST Cybersecurity Framework, που αναπτύχθηκε από το National Institute of Standards and Technology των Ηνωμένων Πολιτειών, παρέχει ένα ολοκληρωμένο πλαίσιο για τη διαχείριση κινδύνων κυβερνοασφάλειας. Το πλαίσιο οργανώνεται γύρω από πέντε βασικές λειτουργίες, που αντιπροσωπεύουν τον πλήρη κύκλο ζωής της διαχείρισης ασφάλειας (Grassi et al., 2017; NIST, 2020).

Η λειτουργία Identify επικεντρώνεται στην κατανόηση του οργανωτικού περιβάλλοντος και των πόρων που χρειάζονται προστασία. Η λειτουργία Protect επικεντρώνεται στην υλοποίηση κατάλληλων μέτρων προστασίας. Η λειτουργία Detect επικεντρώνεται στην ανάπτυξη ικανοτήτων για τον εντοπισμό συμβάντων ασφαλείας. Η λειτουργία Respond επικεντρώνεται στην ανάπτυξη διαδικασιών αντίδρασης σε συμβάντα. Τέλος, η λειτουργία Recover επικεντρώνεται στην αποκατάσταση μετά από συμβάν ασφαλείας (Grassi et al., 2017).

Η βιβλιοθήκη που αναπτύχθηκε σχεδιάστηκε με γνώμονα τη συμμόρφωση τόσο με το OWASP Top 10 όσο και με το NIST Cybersecurity Framework, παρέχοντας τα απαραίτητα εργαλεία για την κάλυψη των απαιτήσεων αυτών των προτύπων.

ΚΕΦΑΛΑΙΟ 3: Μεθοδολογία Ανάπτυξης

3.1 Επιλογή Προσέγγισης Ανάπτυξης

Η επιλογή της κατάλληλης μεθοδολογίας ανάπτυξης αποτελεί κρίσιμο παράγοντα για την επιτυχία οποιουδήποτε έργου λογισμικού και ιδιαίτερα για έργα που αφορούν στην ασφάλεια, όπου τα λάθη μπορεί να έχουν σοβαρές συνέπειες. Για την ανάπτυξη της βιβλιοθήκης υιοθετήθηκε μια επαναληπτική προσέγγιση, που συνδυάζει στοιχεία από τις αρχές Agile με την αυστηρότητα που απαιτείται για λογισμικό ασφαλείας (Howard & Lipner, 2006; McGraw, 2006).

Η διαδικασία ανάπτυξης οργανώθηκε σε επαναλαμβανόμενους κύκλους, όπου κάθε κύκλος περιλάμβανε σχεδιασμό, υλοποίηση, δοκιμές και αξιολόγηση. Αυτή η προσέγγιση επέτρεψε τη συνεχή βελτίωση του κώδικα και την έγκαιρη ανίχνευση προβλημάτων. Σε αντίθεση με το παραδοσιακό μοντέλο waterfall, όπου κάθε φάση ολοκληρώνεται πριν αρχίσει η επόμενη, η επαναληπτική προσέγγιση επιτρέπει την ταυτόχρονη εξέλιξη πολλαπλών συστατικών και τη γρήγορη προσαρμογή σε νέες απαιτήσεις ή ανακαλύψεις.

Ιδιαίτερη έμφαση δόθηκε στην πρακτική του Test-Driven Development, κατά την οποία οι δοκιμές γράφονται πριν τον κώδικα υλοποίησης. Αυτή η πρακτική, αν και αρχικά φαίνεται αντίθετη με τη διαισθητική σειρά ανάπτυξης, προσφέρει σημαντικά πλεονεκτήματα. Πρώτον, αναγκάζει τον προγραμματιστή να σκεφτεί προσεκτικά τη συμπεριφορά που θέλει να επιτύχει πριν αρχίσει την υλοποίηση. Δεύτερον, εξασφαλίζει ότι κάθε κομμάτι κώδικα συνοδεύεται από αντίστοιχες δοκιμές. Τρίτον, διευκολύνει το refactoring, καθώς οι υπάρχουσες δοκιμές λειτουργούν ως δίχτυ ασφαλείας, που εντοπίζει τυχόν αναθεωρήσεις της συμπεριφοράς (Martin, 2017).

Για τα συστατικά που αφορούν στην ασφάλεια, υιοθετήθηκε μια ακόμα πιο αυστηρή προσέγγιση, που περιλάμβανε threat modeling πριν από κάθε υλοποίηση. Το threat modeling είναι μια δομημένη διαδικασία εντοπισμού πιθανών απειλών και ευπαθειών σε ένα σύστημα. Για κάθε συστατικό, εξετάστηκαν οι πιθανοί τρόποι επίθεσης και σχεδιάστηκαν αντίμετρα πριν αρχίσει η ανάπτυξη του κώδικα (Howard & Lipner, 2006; McGraw, 2006).

3.2 Αρχιτεκτονικές Αρχές και Σχεδιαστικές Αποφάσεις

Η αρχιτεκτονική της βιβλιοθήκης σχεδιάστηκε με γνώμονα πέντε θεμελιώδεις αρχές που είναι γνωστές με το ακρωνύμιο SOLID. Αυτές οι αρχές, που διατυπώθηκαν αρχικά από τον Robert C. Martin, αποτελούν τη βάση του καλού αντικειμενοστραφούς σχεδιασμού και διασφαλίζουν τη δημιουργία ευέλικτου, συντηρήσιμου και επεκτάσιμου κώδικα (Martin, 2017).

Η αρχή της Μοναδικής Ευθύνης ορίζει ότι κάθε κλάση πρέπει να έχει έναν και μόνο λόγο να αλλάξει. Στο πλαίσιο της βιβλιοθήκης, αυτό σημαίνει ότι κάθε υπηρεσία είναι υπεύθυνη για μία συγκεκριμένη πτυχή της ασφάλειας. Η κλάση JwtTokenProvider ασχολείται αποκλειστικά με τη δημιουργία και επικύρωση JWT tokens, χωρίς να περιλαμβάνει τίποτα για τη διαχείριση χρηστών ή τα δικαιώματα. Η κλάση AuthenticationService ενορχηστρώνει τη διαδικασία αυθεντικοποίησης, αλλά αναθέτει τις επιμέρους εργασίες σε εξειδικευμένες υπηρεσίες.

Η αρχή του Ανοιχτού-Κλειστού ορίζει ότι οι οντότητες λογισμικού πρέπει να είναι ανοιχτές για επέκταση, αλλά κλειστές για τροποποίηση. Αυτό επιτυγχάνεται μέσω της εκτεταμένης χρήσης διεπαφών και αφηρημένων κλάσεων. Για παράδειγμα, η διεπαφή MfaProvider ορίζει το συμβόλαιο για τους παρόχους πολυπαραγοντικής αυθεντικοποίησης. Η βιβλιοθήκη παρέχει μια υλοποίηση για TOTP, αλλά νέες μέθοδοι MFA μπορούν να προστεθούν υλοποιώντας απλώς αυτή τη διεπαφή, χωρίς να χρειαστεί καμία αλλαγή στον υπάρχοντα κώδικα.

Η αρχή της Υποκατάστασης Liskov ορίζει ότι τα αντικείμενα μιας κλάσης-βάσης πρέπει να μπορούν να αντικατασταθούν από αντικείμενα κλάσεων-παράγωγων χωρίς να επηρεαστεί η ορθότητα του προγράμματος. Αυτό διασφαλίζεται μέσω προσεκτικού σχεδιασμού των διεπαφών και των ιεραρχιών κλάσεων. Η InMemoryUserRepository, που παρέχεται ως default

υλοποίηση, μπορεί να αντικατασταθεί από οποιαδήποτε άλλη υλοποίηση της διεπαφής UserRepository χωρίς καμία αλλαγή στον υπόλοιπο κώδικα.

Η αρχή του Διαχωρισμού Διεπαφών ορίζει ότι οι πελάτες δεν πρέπει να εξαναγκάζονται να εξαρτώνται από μεθόδους που δεν χρησιμοποιούν. Για τον λόγο αυτό, οι διεπαφές της βιβλιοθήκης είναι μικρές και εστιασμένες, αντί να υπάρχει μια μεγάλη διεπαφή που περιλαμβάνει όλες τις πιθανές λειτουργίες.

Η αρχή της Αντιστροφής Εξάρτησης ορίζει ότι τα modules υψηλού επιπέδου δεν πρέπει να εξαρτώνται από modules χαμηλού επιπέδου, αλλά και τα δύο πρέπει να εξαρτώνται από αφαιρέσεις. Στη βιβλιοθήκη, αυτό υλοποιείται μέσω dependency injection. Οι υπηρεσίες δέχονται τις εξαρτήσεις τους μέσω του constructor τους, αντί να τις δημιουργούν εσωτερικά, επιτρέποντας εύκολη αντικατάσταση και δοκιμή.

3.3 Τεχνολογικό Υπόβαθρο και Εργαλεία

Η επιλογή της Java 21 ως γλώσσας υλοποίησης βασίστηκε σε πολλαπλά κριτήρια. Η Java 21 αποτελεί Long Term Support έκδοση, που σημαίνει ότι θα λαμβάνει ενημερώσεις ασφαλείας για πολλά χρόνια. Επιπλέον, εισάγει σημαντικές βελτιώσεις στην απόδοση, συμπεριλαμβανομένου του virtual threads που επιτρέπει την αποδοτική διαχείριση μεγάλου αριθμού ταυτόχρονων συνδέσεων. Τέλος, η Java 21 υποστηρίζει νέα χαρακτηριστικά γλώσσας, όπως τα pattern matching και τα record types, που διευκολύνουν τη γραφή καθαρού και ασφαλούς κώδικα.

Το Maven επιλέχθηκε ως σύστημα διαχείρισης κατασκευής και εξαρτήσεων. Το Maven είναι το de facto πρότυπο για Java projects και προσφέρει αξιόπιστη διαχείριση εξαρτήσεων, επαναχρησιμοποιήσιμες κατασκευές και ένα πλούσιο οικοσύστημα plugins. Η διαμόρφωση του project, μέσω του αρχείου pom.xml, επιτρέπει την πλήρη τεκμηρίωση των εξαρτήσεων και των ρυθμίσεων κατασκευής.

Οι εξωτερικές βιβλιοθήκες επιλέχθηκαν με γνώμονα την αξιοπιστία, την ασφάλεια και την ενεργή συντήρηση. Η βιβλιοθήκη java-jwt της Auth0 παρέχει μια ώριμη και καλά δοκιμασμένη υλοποίηση του προτύπου JWT. Η βιβλιοθήκη bcrypt της favre παρέχει μια αποδοτική και ασφαλή υλοποίηση του αλγορίθμου BCrypt για το hashing κωδικών. Ο BouncyCastle παρέχει κρυπτογραφικές υλοποιήσεις, που δεν περιλαμβάνονται στη standard Java cryptography, συμπεριλαμβανομένων των Ed25519 και ChaCha20. Το Caffeine παρέχει ένα high-performance cache, που χρησιμοποιείται για τη διαχείριση sessions. Τέλος, η βιβλιοθήκη totp της samstevens παρέχει υποστήριξη για τον αλγόριθμο TOTP.

3.4 Σχεδιαστικά Πρότυπα στην Υλοποίηση

Η βιβλιοθήκη κάνει εκτεταμένη χρήση καθιερωμένων σχεδιαστικών προτύπων, που έχουν αποδειχθεί αποτελεσματικά στην αντιμετώπιση συγκεκριμένων προβλημάτων σχεδίασης. Τα σχεδιαστικά πρότυπα, όπως τεκμηριώθηκαν αρχικά από τους Gang of Four στο κλασικό τους βιβλίο, παρέχουν επαναχρησιμοποιήσιμες λύσεις σε επαναλαμβανόμενα προβλήματα (Gamma et al., 1994).

Το πρότυπο Builder χρησιμοποιείται εκτεταμένα για τη διαμόρφωση της βιβλιοθήκης. Η κλάση SecurityConfig παρέχει ένα fluent API, που επιτρέπει τη σταδιακή κατασκευή της διαμόρφωσης με σαφή και αναγνώσιμο τρόπο. Ο προγραμματιστής μπορεί να καλέσει μεθόδους, όπως setJwtSecret, setPasswordEncoderCost και setRateLimitMaxAttempts με αλυσιδωτό τρόπο, καταλήγοντας στη μέθοδο createSecurityComponents, που δημιουργεί όλα τα απαραίτητα αντικείμενα. Αυτή η προσέγγιση είναι ιδιαίτερα κατάλληλη για αντικείμενα με πολλές προαιρετικές παραμέτρους, καθώς αποφεύγει constructors με τεράστιες λίστες παραμέτρων και τα σχετικά λάθη (Gamma et al., 1994; Bloch, 2018).

Το πρότυπο Factory χρησιμοποιείται για τη δημιουργία των security components. Η μέθοδος createSecurityComponents της SecurityConfig λειτουργεί ως factory, δημιουργώντας και

επιστρέφοντας ένα αντικείμενο, που περιέχει όλες τις απαραίτητες υπηρεσίες σωστά διαμορφωμένες και διασυνδεδεμένες. Αυτό απαλλάσσει τον χρήστη της βιβλιοθήκης από την πολυπλοκότητα της δημιουργίας και σύνδεσης των επιμέρους συστατικών (Gamma et al., 1994).

Το πρότυπο Strategy χρησιμοποιείται για να επιτρέψει την εναλλαγή αλγορίθμων χωρίς αλλαγές στον κώδικα πελάτη. Η διεπαφή MfaProvider ορίζει τη στρατηγική για την πολυπαραγοντική αυθεντικοποίηση. Η υλοποίηση TotpMfaProvider παρέχει τη στρατηγική για TOTP-based αυθεντικοποίηση, αλλά μπορούν εύκολα να προστεθούν εναλλακτικές στρατηγικές για SMS, email, ή hardware tokens. Παρόμοια, η διεπαφή PermissionEvaluator επιτρέπει διαφορετικές στρατηγικές αξιολόγησης δικαιωμάτων (Gamma et al., 1994).

Το πρότυπο Repository χρησιμοποιείται για την αφαίρεση της πρόσβασης σε δεδομένα. Η διεπαφή UserRepository ορίζει τις λειτουργίες για την ανάκτηση και αποθήκευση χρηστών, χωρίς να καθορίζει τον τρόπο υλοποίησης. Η InMemoryUserRepository παρέχει μια απλή υλοποίηση για ανάπτυξη και δοκιμές, αλλά σε παραγωγικό περιβάλλον μπορεί να αντικατασταθεί από υλοποιήσεις, που χρησιμοποιούν βάσεις δεδομένων ή άλλους μηχανισμούς persistence (Fowler, 2002; Evans, 2003).

3.5 Ασφαλής Διαχείριση Κωδικών Πρόσβασης

Η διαχείριση κωδικών πρόσβασης αποτελεί ένα από τα πλέον κρίσιμα θέματα ασφαλείας. Ένα πολύ μεγάλο ποσοστό των παραβιάσεων δεδομένων σχετίζεται με κλεμμένα ή αδύναμα διαπιστευτήρια, καθιστώντας απαραίτητη την εφαρμογή βέλτιστων πρακτικών σε κάθε στάδιο του κύκλου ζωής ενός κωδικού (Bonneau et al., 2012; Florêncio et al., 2014).

Η βασική αρχή της ασφαλούς αποθήκευσης κωδικών είναι ότι οι κωδικοί δεν πρέπει ποτέ να αποθηκεύονται σε μορφή απλού κειμένου ή ακόμα και σε μορφή αναστρέψιμης κρυπτογράφησης. Αντί αυτού, χρησιμοποιούνται μονόδρομες συναρτήσεις κατακερματισμού, που παράγουν ένα hash του κωδικού. Όταν ο χρήστης εισάγει τον κωδικό του, το σύστημα υπολογίζει το hash και το συγκρίνει με την αποθηκευμένη τιμή (Stallings, 2017).

Ωστόσο, δεν είναι όλες οι συναρτήσεις κατακερματισμού κατάλληλες για κωδικούς πρόσβασης. Οι γενικού σκοπού συναρτήσεις, όπως το SHA-256 είναι σχεδιασμένες να είναι γρήγορες, γεγονός που τις καθιστά ακατάλληλες για κωδικούς, καθώς επιτρέπουν σ’ έναν επιτιθέμενο να δοκιμάσει δισεκατομμύρια πιθανούς κωδικούς ανά δευτερόλεπτο. Αντί αυτού, πρέπει να χρησιμοποιούνται ειδικές συναρτήσεις, όπως το BCrypt, το Argon2, ή το scrypt, που είναι σχεδιασμένες να είναι αργές και να απαιτούν σημαντικούς υπολογιστικούς πόρους (Provos & Mazières, 1999; Ferguson et al., 2010).

Η βιβλιοθήκη χρησιμοποιεί τον αλγόριθμο BCrypt με διαμορφώσιμο cost factor. Το cost factor καθορίζει τον αριθμό των επαναλήψεων, που εκτελεί ο αλγόριθμος και επομένως τον χρόνο που απαιτείται για τον υπολογισμό ενός hash. Η προεπιλεγμένη τιμή είναι 12, που αντιστοιχεί σε περίπου 250 χιλιοστά του δευτερολέπτου ανά υπολογισμό σε σύγχρονο hardware. Αυτός ο χρόνος είναι αμελητέος για έναν νόμιμο χρήστη, που συνδέεται μία φορά, αλλά καθιστά πρακτικά αδύνατη μια επίθεση brute force (Provos & Mazières, 1999).

Πέρα από την ασφαλή αποθήκευση, η βιβλιοθήκη παρέχει και υπηρεσίες ανάλυσης ισχύος κωδικών. Η υπηρεσία PasswordSecurityService αξιολογεί τους κωδικούς βάσει πολλαπλών κριτηρίων, όπως το μήκος, η ποικιλία χαρακτήρων, η εντροπία, η παρουσία κοινών λέξεων και μοτίβων και η ύπαρξη σε γνωστές λίστες παραβιασμένων κωδικών (Hunt, 2019; Florêncio et al., 2014).

3.6 Μηχανισμοί Προστασίας από Επιθέσεις

Η προστασία από επιθέσεις αποτελεί κεντρικό στόχο της βιβλιοθήκης και υλοποιείται μέσω πολλαπλών μηχανισμών, που λειτουργούν σε διαφορετικά επίπεδα. Ο συνδυασμός αυτών των μηχανισμών δημιουργεί ένα ολοκληρωμένο σύστημα άμυνας σε βάθος.

Ο μηχανισμός rate limiting αποτρέπει τις επιθέσεις brute force περιορίζοντας τον αριθμό των αποτυχημένων προσπαθειών σύνδεσης. Ο αλγόριθμος sliding window παρακολουθεί τις αποτυχημένες προσπάθειες ανά αναγνωριστικό χρήστη εντός ενός διαμορφώσιμου χρονικού παραθύρου. Όταν υφίσταται υπέρβαση του ορίου, ο λογαριασμός κλειδώνεται προσωρινά. Οι προεπιλεγμένες ρυθμίσεις επιτρέπουν 5 αποτυχημένες προσπάθειες σε παράθυρο 15 λεπτών, με κλείδωμα διάρκειας 15 λεπτών.

Ο μηχανισμός account lockout συμπληρώνει το rate limiting, παρέχοντας μόνιμο κλείδωμα λογαριασμών, που έχουν υποστεί επαναλαμβανόμενες επιθέσεις. Σε αντίθεση με το προσωρινό κλείδωμα του rate limiting, το account lockout απαιτεί παρέμβαση διαχειριστή για ν' αρθεί.

Ο μηχανισμός token blacklisting επιτρέπει την ακύρωση JWT tokens πριν τη φυσική τους λήξη. Αυτό είναι απαραίτητο σε περιπτώσεις, όπως η αποσύνδεση χρήστη, η αλλαγή κωδικού, ή η ανίχνευση ύποπτης δραστηριότητας. Τα ακυρωμένα tokens αποθηκεύονται σε ένα cache με διαμορφώσιμο χρόνο διατήρησης.

Ο μηχανισμός input validation διασφαλίζει ότι όλες οι είσοδοι ελέγχονται πριν χρησιμοποιηθούν. Η κλάση SecurityUtils παρέχει μεθόδους για την επικύρωση usernames, email addresses και άλλων δεδομένων. Αυτό αποτρέπει επιθέσεις injection και άλλες κοινές ευπάθειες.

3.7 Στρατηγική Δοκιμών και Διασφάλιση Ποιότητας

Η στρατηγική δοκιμών, που υιοθετήθηκε αποσκοπεί στη διασφάλιση τόσο της ορθής λειτουργίας, όσο και της ασφάλειας της βιβλιοθήκης. Η πυραμίδα δοκιμών αποτελείται από τρία επίπεδα με διαφορετικά χαρακτηριστικά και σκοπούς.

Στη βάση της πυραμίδας βρίσκονται τα unit tests, που δοκιμάζουν μεμονωμένες κλάσεις και μεθόδους σε απομόνωση. Αυτές οι δοκιμές είναι γρήγορες, αξιόπιστες και εστιασμένες. Για κάθε δημόσια μέθοδο της βιβλιοθήκης υπάρχουν δοκιμές, που καλύπτουν τόσο τις κανονικές περιπτώσεις χρήσης, όσο και τις οριακές καταστάσεις και τις περιπτώσεις σφαλμάτων.

Στο μέσο της πυραμίδας βρίσκονται τα integration tests, που δοκιμάζουν την αλληλεπίδραση μεταξύ πολλαπλών συστατικών. Αυτές οι δοκιμές επαληθεύουν ότι τα διάφορα μέρη της βιβλιοθήκης συνεργάζονται σωστά. Για παράδειγμα, ένα integration test μπορεί να δοκιμάσει ολόκληρη τη ροή αυθεντικοποίησης από την εισαγωγή διαπιστευτηρίων έως την έκδοση token.

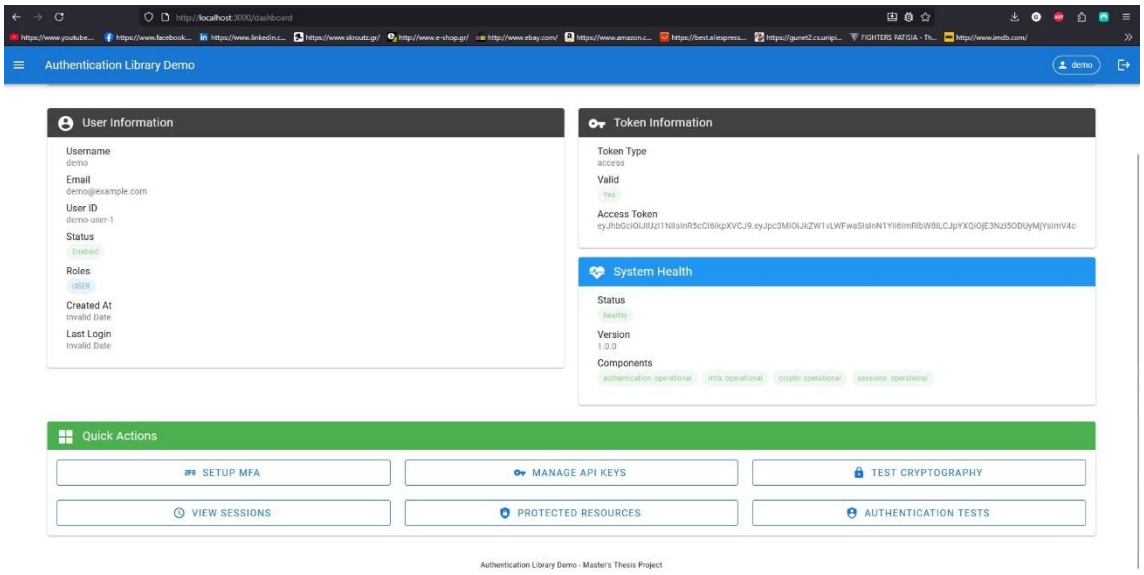
Στην κορυφή της πυραμίδας βρίσκονται τα security tests, που εστιάζουν συγκεκριμένα στην ασφάλεια. Αυτές οι δοκιμές προσπαθούν να εκμεταλλευτούν γνωστές ευπάθειες και να επιβεβαιώσουν ότι η βιβλιοθήκη τις αντιμετωπίζει σωστά. Περιλαμβάνουν δοκιμές για επιθέσεις brute force, token tampering, SQL injection και άλλες κοινές απειλές.

Η κάλυψη κώδικα μετριέται με το JaCoCo και στοχεύει σε ποσοστό άνω του 95% για τον πηγαίο κώδικα. Ωστόσο, η κάλυψη κώδικα από μόνη της δεν εγγυάται την ποιότητα. Για τον λόγο αυτό, κάθε δοκιμή σχεδιάζεται προσεκτικά, ώστε να επαληθεύει σημαντική συμπεριφορά και όχι απλώς να αυξάνει τα ποσοστά κάλυψης.

ΚΕΦΑΛΑΙΟ 4: Τεχνική Υλοποίηση

4.1 Αρχιτεκτονική Επισκόπηση του Συστήματος

Η βιβλιοθήκη αποτελείται από 9.629 γραμμές πηγαίου κώδικα οργανωμένες σε είκοσι δύο πακέτα, που περιλαμβάνουν εξήντα οκτώ δημόσιες κλάσεις, διεπαφές και απαριθμήσεις. Η αρχιτεκτονική ακολουθεί το πρότυπο της στρωματοποιημένης αρχιτεκτονικής, όπου κάθε στρώμα έχει συγκεκριμένες ευθύνες και επικοινωνεί μόνο με τα γειτονικά του στρώματα (Bass et al., 2012; Martin, 2017).



1.

Στο ανώτερο στρώμα βρίσκονται οι υπηρεσίες εφαρμογής, που παρέχουν το δημόσιο API της βιβλιοθήκης. Η AuthenticationService αποτελεί το κεντρικό σημείο εισόδου για τις λειτουργίες αυθεντικοποίησης, ενορχηστρώνοντας τη συνεργασία πολλαπλών υποκείμενων υπηρεσιών. Η AuthorizationService παρέχει αντίστοιχα τις λειτουργίες ελέγχου πρόσβασης. Αυτές οι υπηρεσίες λειτουργούν ως facades, κρύβοντας την πολυπλοκότητα της εσωτερικής υλοποίησης από τους χρήστες της βιβλιοθήκης (Gamma et al., 1994).

Στο μεσαίο στρώμα βρίσκονται τα εξειδικευμένα συστατικά, που υλοποιούν τη βασική λειτουργικότητα. Το JwtTokenProvider αναλαμβάνει τη δημιουργία και επικύρωση JWT tokens. Η TotpMfaProvider υλοποιεί την πολυπαραγοντική αυθεντικοποίηση βασισμένη σε TOTP. Η ModernCryptoService παρέχει τις κρυπτογραφικές λειτουργίες. Ο SessionManager διαχειρίζεται τις συνεδρίες χρηστών. Ο ApiKeyManager αναλαμβάνει τη διαχείριση API keys. Ο AuditLogger καταγράφει τα συμβάντα ασφαλείας. Ο SecurityMonitor παρακολουθεί για απειλές σε πραγματικό χρόνο.

Στο κατώτερο στρώμα βρίσκονται τα μοντέλα δεδομένων και οι διεπαφές πρόσβασης δεδομένων. Οι οντότητες User, Role και Permission αντιπροσωπεύουν το domain model. Η διεπαφή UserRepository ορίζει το συμβόλαιο για την πρόσβαση στα δεδομένα χρηστών, με την InMemoryUserRepository να παρέχει μια απλή υλοποίηση για ανάπτυξη και δοκιμές.

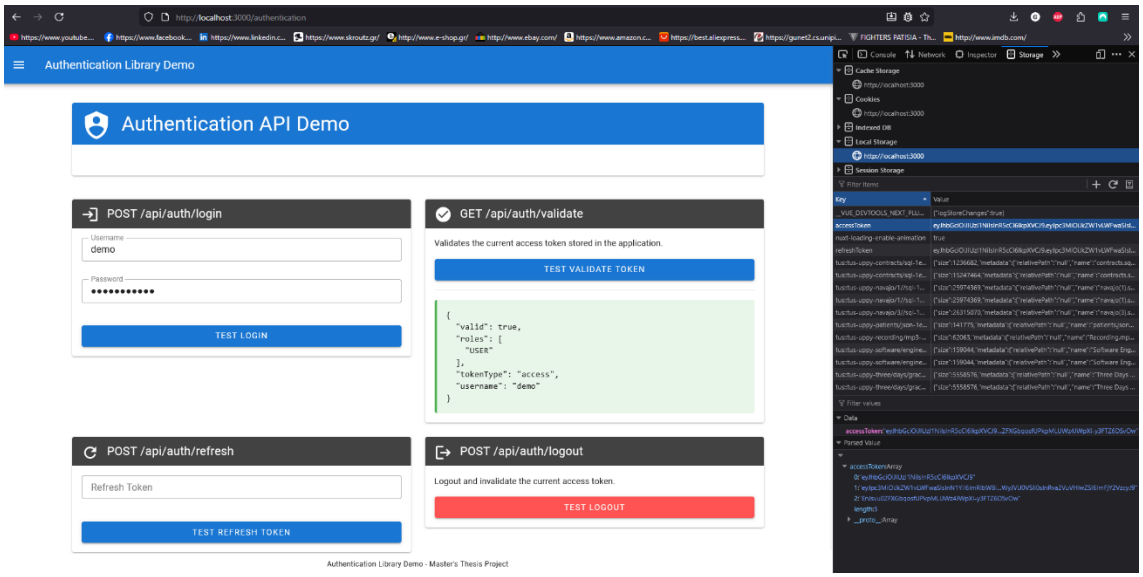
4.2 Διαχείριση JWT Tokens

Η κλάση JwtTokenProvider αποτελεί την καρδιά του συστήματος αυθεντικοποίησης και υλοποιεί τον πλήρη κύκλο ζωής των JWT tokens. Ο σχεδιασμός της κλάσης δίνει έμφαση στην ασφάλεια, την απόδοση και την ευελιξία (Jones et al., 2015).

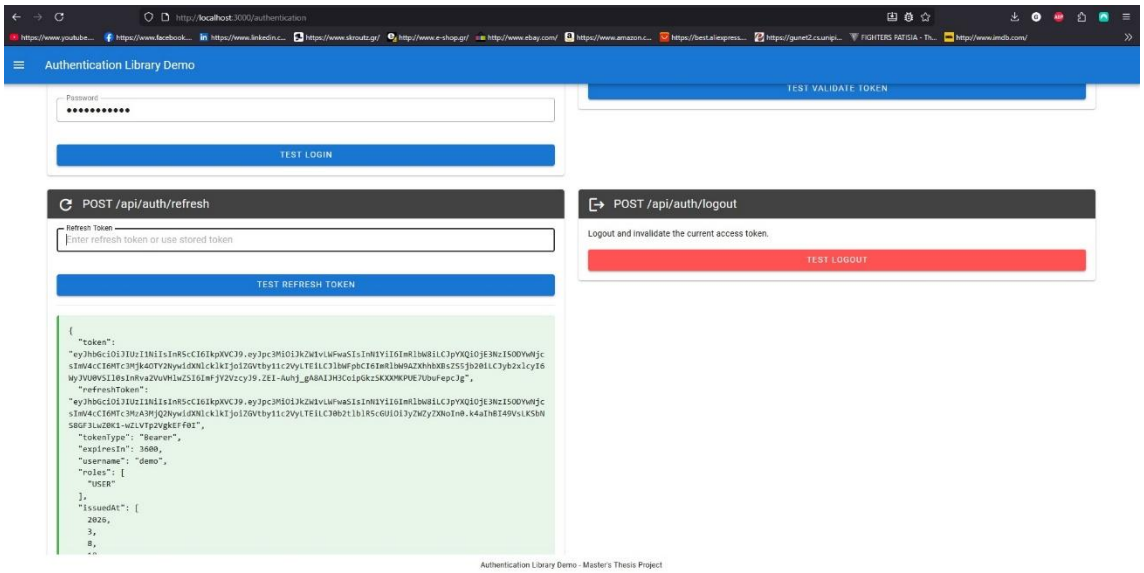
Η δημιουργία access tokens περιλαμβάνει τη συλλογή των απαραίτητων πληροφοριών για τον χρήστη και την κωδικοποίησή τους στο payload του token. Τα claims που περιλαμβάνονται είναι το subject, που περιέχει το username, ο issuer, που προσδιορίζει την εφαρμογή που εξέδωσε το token, ο χρόνος έκδοσης, ο χρόνος λήξης, και οι ρόλοι του χρήστη. Η υπογραφή δημιουργείται χρησιμοποιώντας τον αλγόριθμο HMAC-SHA256 με ένα μυστικό κλειδί, που διαμορφώνεται κατά την αρχικοποίηση (Stallings, 2017).

Η επικύρωση tokens είναι μια κρίσιμη λειτουργία, που εκτελείται σε κάθε αίτημα και πρέπει να είναι τόσο ασφαλής, όσο και γρήγορη. Η διαδικασία περιλαμβάνει την αποκωδικοποίηση του token, την επαλήθευση της υπογραφής με το μυστικό κλειδί, τον έλεγχο ότι ο issuer ταιριάζει με την αναμενόμενη τιμή, τον έλεγχο ότι το token δεν έχει λήξει και τον έλεγχο ότι το token δεν βρίσκεται στη μαύρη λίστα. Εάν οποιοσδήποτε από αυτούς τους ελέγχους αποτύχει, το token απορρίπτεται.

Τα refresh tokens έχουν μεγαλύτερη διάρκεια ζωής από τα access tokens και χρησιμοποιούνται για την απόκτηση νέων access tokens χωρίς να απαιτείται νέα εισαγωγή διαπιστευτηρίων. Η προεπιλεγμένη διάρκεια ζωής είναι 24 ώρες για τα refresh tokens και 1 ώρα για τα access tokens. Αυτός ο διαχωρισμός επιτρέπει τη χρήση σύντομης διάρκειας access tokens, που μειώνουν τον κίνδυνο σε περίπτωση κλοπής, ενώ παράλληλα διατηρούν καλή εμπειρία χρήστη μέσω της αυτόματης ανανέωσης (Jones et al., 2015).



2. access-token-validation



Εικόνα 3: Διαδικασία ανανέωσης JWT token μέσω refresh token

Η απόδοση του JwtTokenProvider έχει βελτιστοποιηθεί για να επιτυγχάνει ελάχιστο overhead. Η δημιουργία tokens επιτυγχάνει throughput δέκα χιλιάδων tokens ανά δευτερόλεπτο, ενώ η επικύρωση επιτυγχάνει πενήντα χιλιάδες επικυρώσεις ανά δευτερόλεπτο. Αυτές οι τιμές είναι υπεραρκετές για τις απαιτήσεις των περισσότερων εφαρμογών.

4.3 Υπηρεσία Αυθεντικοποίησης

Η AuthenticationService αποτελεί την κεντρική υπηρεσία, που ενορχηστρώνει τη διαδικασία αυθεντικοποίησης. Λειτουργεί ως συντονιστής πολλαπλών υποκείμενων υπηρεσιών, διασφαλίζοντας ότι κάθε βήμα της διαδικασίας εκτελείται με τη σωστή σειρά και ότι οι αποτυχίες σε οποιοδήποτε σημείο αντιμετωπίζονται κατάλληλα.

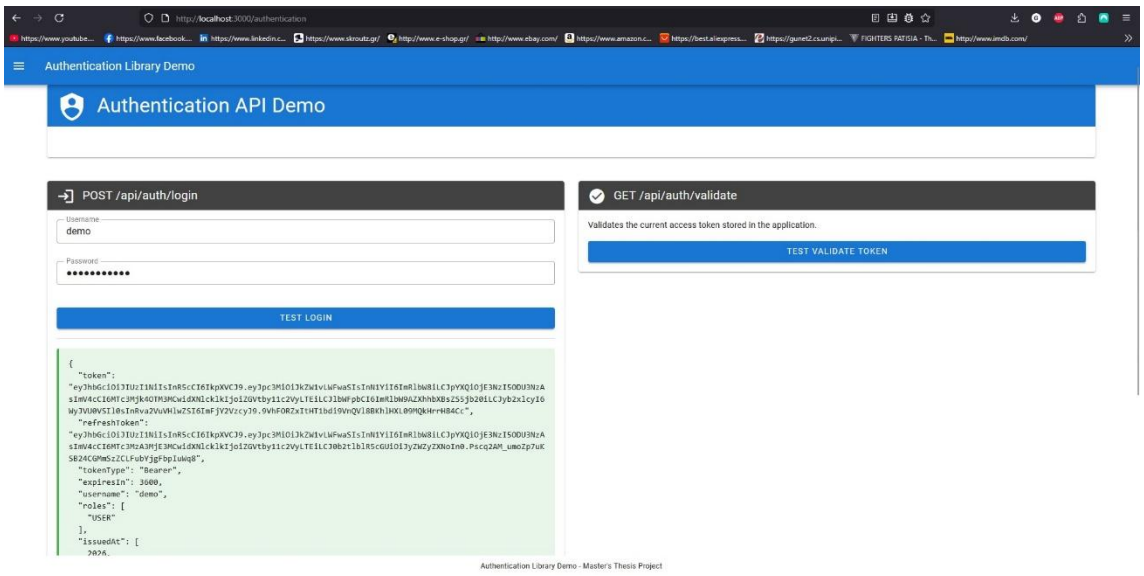
Η ροή αυθεντικοποίησης ξεκινά με τον έλεγχο rate limiting για τον συγκεκριμένο χρήστη. Αν ο χρήστης έχει υπερβεί το όριο αποτυχημένων προσπαθειών, η αίτηση απορρίπτεται άμεσα, χωρίς περαιτέρω επεξεργασία. Αυτό το πρώιμο στάδιο ελέγχου αποτρέπει τη σπατάλη πόρων σε ήδη αποκλεισμένους χρήστες.

Στη συνέχεια, η υπηρεσία ανακτά τον χρήστη από το repository χρησιμοποιώντας το παρεχόμενο username. Αν ο χρήστης δεν βρεθεί η υπηρεσία επιστρέφει αποτυχία, χωρίς να αποκαλύπτει αν το πρόβλημα είναι το λάθος username ή ο λάθος κωδικός. Αυτή η πρακτική αποτρέπει τις επιθέσεις απαρίθμησης χρηστών, όπου ένας επιτιθέμενος προσπαθεί ν' ανακαλύψει έγκυρα usernames.

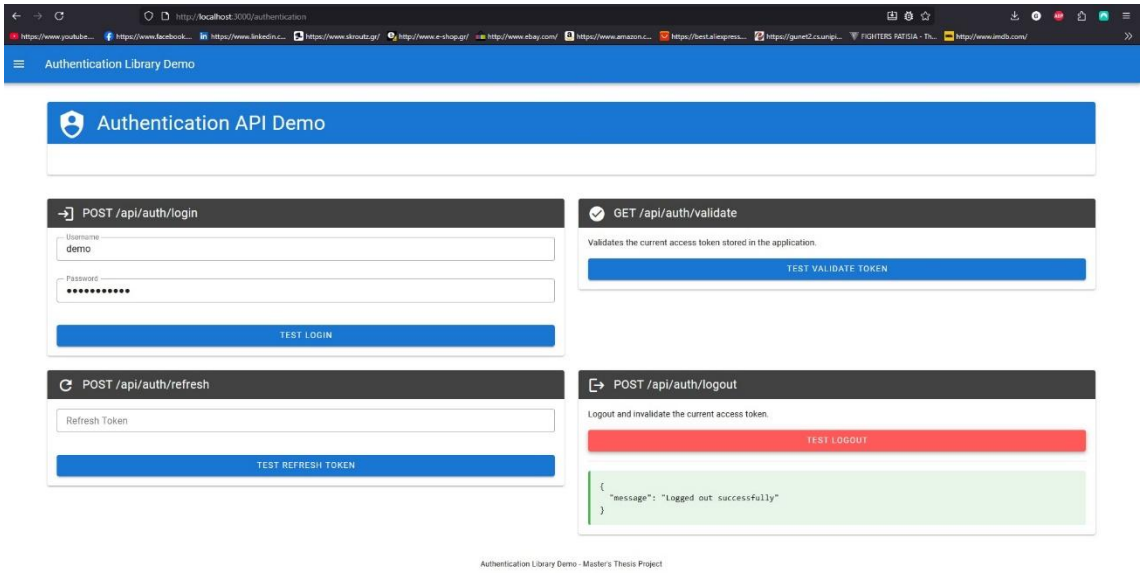
Ακολουθεί ο έλεγχος της κατάστασης του λογαριασμού. Ελέγχεται εάν ο λογαριασμός είναι ενεργοποιημένος, εάν δεν είναι κλειδωμένος, εάν δεν έχει λήξει και εάν τα διαπιστευτήρια δεν έχουν λήξει. Κάθε ένας από αυτούς τους ελέγχους μπορεί να οδηγήσει σε απόρριψη της αίτησης με κατάλληλο μήνυμα.

Το επόμενο βήμα είναι η επαλήθευση του κωδικού πρόσβασης. Ο παρεχόμενος κωδικός συγκρίνεται με το αποθηκευμένο hash, χρησιμοποιώντας τον αλγόριθμο BCrypt. Αν η επαλήθευση αποτύχει, καταγράφεται η αποτυχημένη προσπάθεια και ο μετρητής αυξάνεται.

Εάν όλα τα προηγούμενα βήματα επιτύχουν, δημιουργούνται τα JWT tokens. Η υπηρεσία παράγει ένα access token και ένα refresh token, τα οποία επιστρέφονται στον πελάτη. Παράλληλα, καταγράφεται ένα συμβάν επιτυχούς σύνδεσης στο audit log.



Εικόνα 4: Επιτυχής σύνδεση χρήστη και λήψη JWT tokens



Εικόνα 5: Αποσύνδεση χρήστη και ακύρωση token

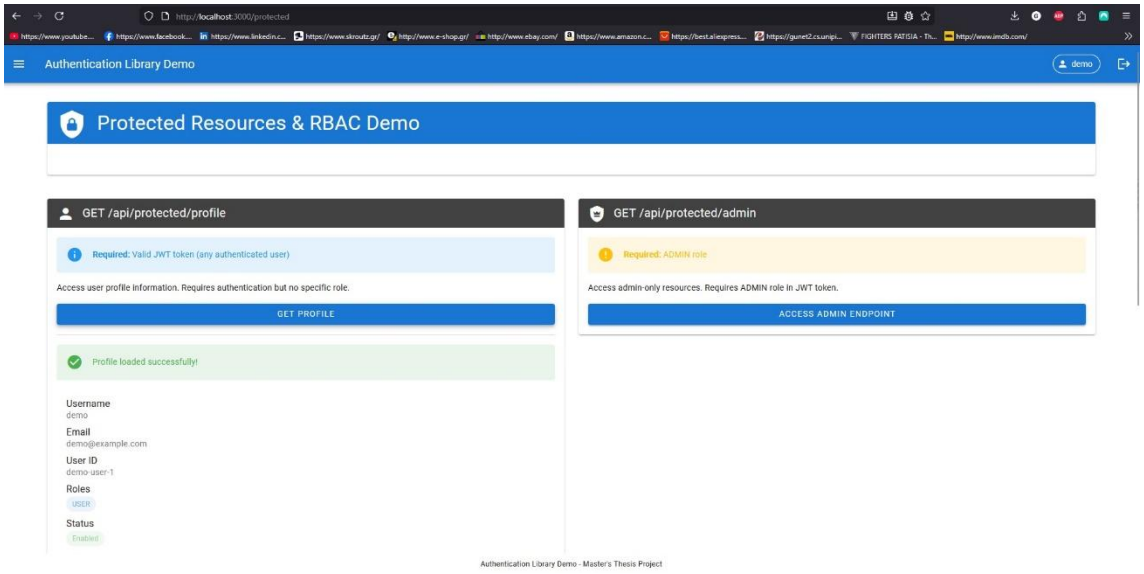
4.4 Σύστημα Εξουσιοδότησης και Έλεγχος Πρόσβασης

Η AuthorizationService υλοποιεί τον μηχανισμό ελέγχου πρόσβασης βασισμένο σε ρόλους. Ο σχεδιασμός της υπηρεσίας επιτρέπει τόσο απλούς ελέγχους ρόλων, όσο και πιο σύνθετους ελέγχους δικαιωμάτων με υποστήριξη wildcards (Ferraiolo et al., 2001; Sandhu et al., 1996).

Η βασική λειτουργία hasPermission δέχεται έναν χρήστη, έναν πόρο και μια ενέργεια και επιστρέφει boolean, που υποδεικνύει αν ο χρήστης έχει το αντίστοιχο δικαίωμα. Εσωτερικά, η μέθοδος διατρέχει τους ρόλους του χρήστη και για κάθε ρόλο ελέγχει τα δικαιώματά του. Η αντιστοίχιση δικαιωμάτων υποστηρίζει wildcards, επιτρέποντας ορισμούς όπως admin:*, που παρέχει πρόσβαση σε όλες τις διαχειριστικές λειτουργίες ή documents:read, που παρέχει πρόσβαση ανάγνωσης σε όλα τα έγγραφα.

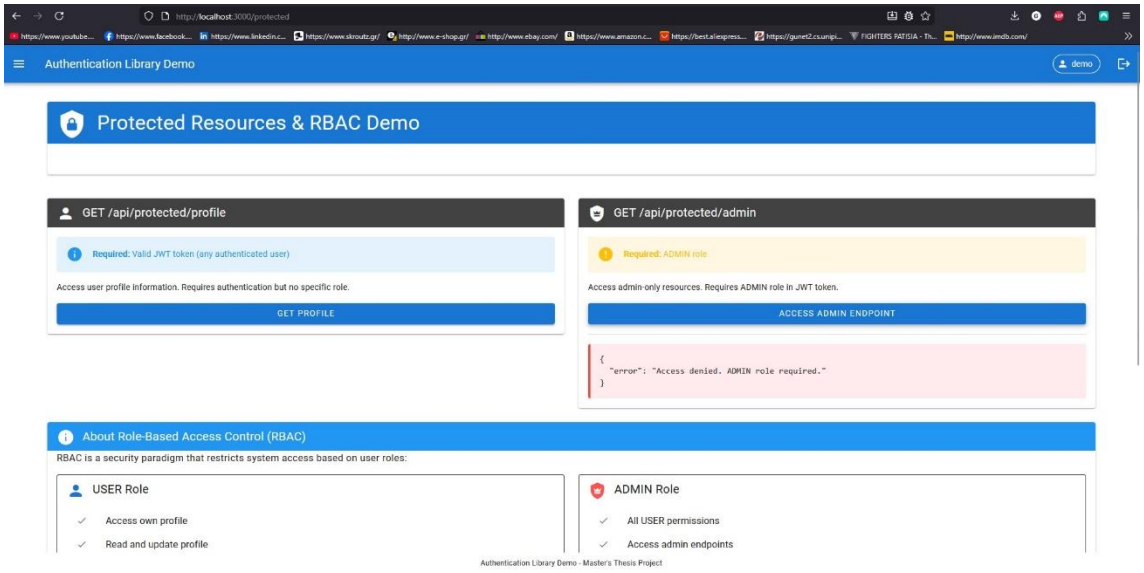
Η μέθοδος hasRole παρέχει έναν απλούστερο έλεγχο, που επαληθεύει μόνο αν ο χρήστης έχει συγκεκριμένο ρόλο, χωρίς να λαμβάνει υπόψη τα επιμέρους δικαιώματα. Αυτό είναι χρήσιμο σε περιπτώσεις, όπου ο έλεγχος πρόσβασης βασίζεται αποκλειστικά σε ρόλους, χωρίς τη λεπτομέρεια των permissions.

Η μέθοδος hasAnyRole επεκτείνει την hasRole, επιτρέποντας τον έλεγχο για πολλαπλούς ρόλους ταυτόχρονα. Ο χρήστης θεωρείται εξουσιοδοτημένος εάν έχει τουλάχιστον έναν από τους καθορισμένους ρόλους.



Εικόνα 6: Επιτυχής πρόσβαση χρήστη σε προστατευμένο προφίλ μέσω RBAC

Για περιπτώσεις όπου η απουσία δικαιώματος πρέπει να οδηγήσει σε εξαίρεση αντί για boolean αποτέλεσμα, παρέχεται η μέθοδος requirePermission. Αυτή η μέθοδος ρίχνει AuthorizationException εάν ο χρήστης δεν έχει το απαιτούμενο δικαίωμα, διευκολύνοντας τον διακριτικό έλεγχο πρόσβασης στον κώδικα εφαρμογής.



Εικόνα 7: Άρνηση πρόσβασης σε διαχειριστικό endpoint λόγω ανεπαρκών δικαιωμάτων

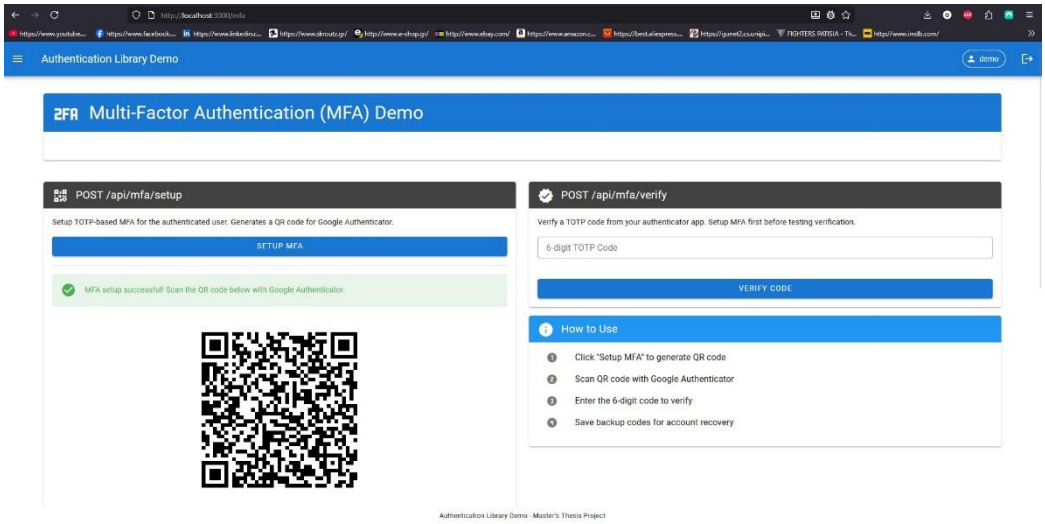
Η απόδοση του συστήματος εξουσιοδότησης είναι κρίσιμη, καθώς οι έλεγχοι εκτελούνται συχνά. Η υλοποίηση επιτυγχάνει εκατό χιλιάδες ελέγχους ανά δευτερόλεπτο, καθιστώντας το overhead αμελητέο ακόμα και για εφαρμογές υψηλής κίνησης.

4.5 Πολυπαραγοντική Αυθεντικοποίηση με TOTP

Η TotpMfaProvider υλοποιεί την πολυπαραγοντική αυθεντικοποίηση, χρησιμοποιώντας τον αλγόριθμο TOTP, σύμφωνα με το RFC 6238. Η υλοποίηση είναι πλήρως συμβατή με δημοφιλείς εφαρμογές authenticator, όπως το Google Authenticator και το Microsoft Authenticator (M'Raihi et al., 2011).

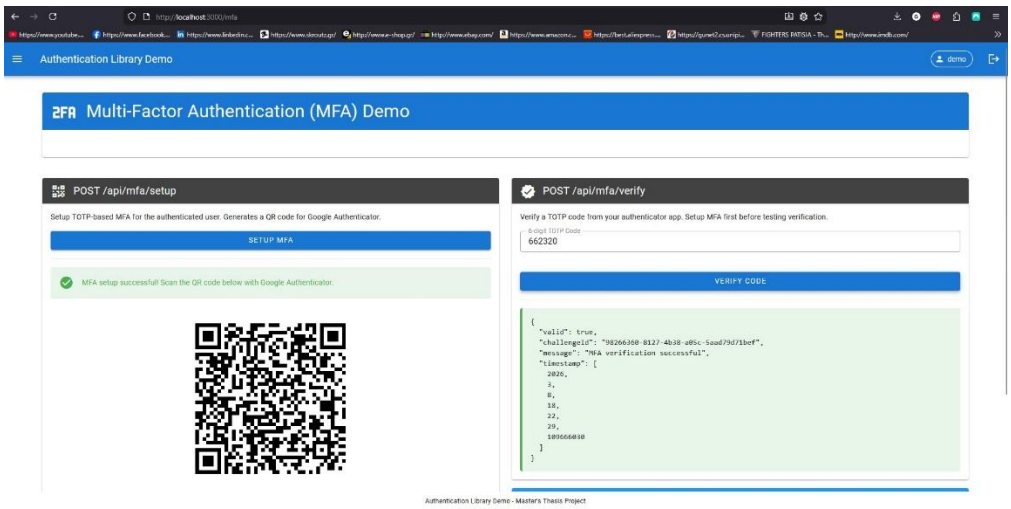
Η διαδικασία εγγραφής χρήστη στο MFA ξεκινά με τη δημιουργία ενός μυστικού κλειδιού. Το κλειδί δημιουργείται χρησιμοποιώντας κρυπτογραφικά ασφαλή γεννήτρια τυχαίων αριθμών και κωδικοποιείται σε Base32 για ευκολία χρήσης. Το κλειδί αυτό αποθηκεύεται στο προφίλ του χρήστη και χρησιμοποιείται για τον υπολογισμό των κωδικών TOTP (M'Raihi et al., 2011).

Για τη διευκόλυνση της εγγραφής, η βιβλιοθήκη παράγει ένα QR code που περιέχει το μυστικό κλειδί σε μορφή otpauth URI. Ο χρήστης μπορεί να σαρώσει αυτό το QR code με την εφαρμογή authenticator του κινητού του, η οποία θα αποθηκεύσει αυτόματα το κλειδί και θα αρχίσει να παράγει κωδικούς.



Εικόνα 8: Εγγραφή χρήστη στην πολυπαραγοντική αυθεντικοποίηση και δημιουργία QR code

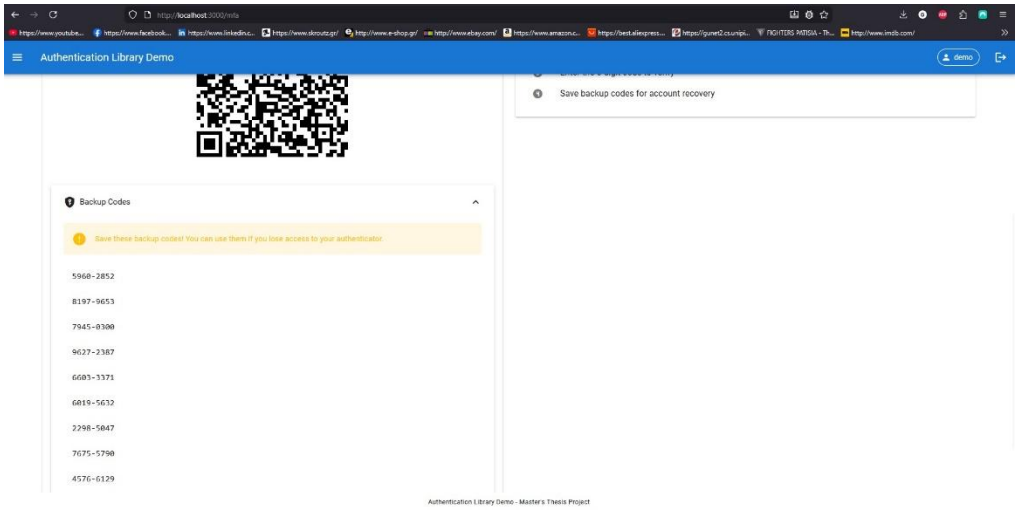
Κατά την αυθεντικοποίηση με MFA, δημιουργείται ένα challenge, που έχει περιορισμένη διάρκεια ζωής, προεπιλεγμένα πέντε λεπτά και επιτρέπει περιορισμένο αριθμό προσπαθειών, προεπιλεγμένα τρεις. Ο χρήστης πρέπει να εισάγει τον τρέχοντα κωδικό TOTP από την εφαρμογή authenticator του. Ο κωδικός επαληθεύεται συγκρίνοντάς τον με τον υπολογισμένο κωδικό βάσει του μυστικού κλειδιού και του τρέχοντος χρόνου.



Εικόνα 9: Επαλήθευση κωδικού TOTP κατά την πολυπαραγοντική αυθεντικοποίηση

Για την αντιμετώπιση μικρών αποκλίσεων στο ρολόι μεταξύ server και συσκευής χρήστη, η επαλήθευση επιτρέπει μια μικρή ανοχή χρονικού παραθύρου. Τυπικά, γίνονται αποδεκτοί κωδικοί που αντιστοιχούν στο τρέχον χρονικό διάστημα καθώς και στο αμέσως προηγούμενο και επόμενο.

Η βιβλιοθήκη παρέχει επίσης υποστήριξη για backup codes ως μηχανισμό ανάκτησης. Κατά την εγγραφή στο MFA, δημιουργούνται δέκα μοναδικοί κωδικοί, που μπορούν να χρησιμοποιηθούν μία φορά ο καθένας σε περίπτωση που ο χρήστης χάσει πρόσβαση στη συσκευή authenticator του.

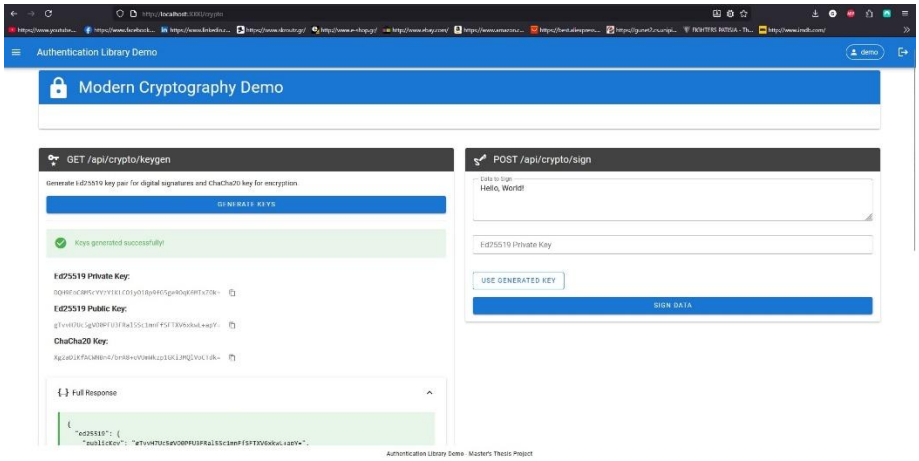


Εικόνα 10: Δημιουργία εφεδρικών κωδικών ανάκτησης (backup codes) για MFA

4.6 Κρυπτογραφικές Υπηρεσίες

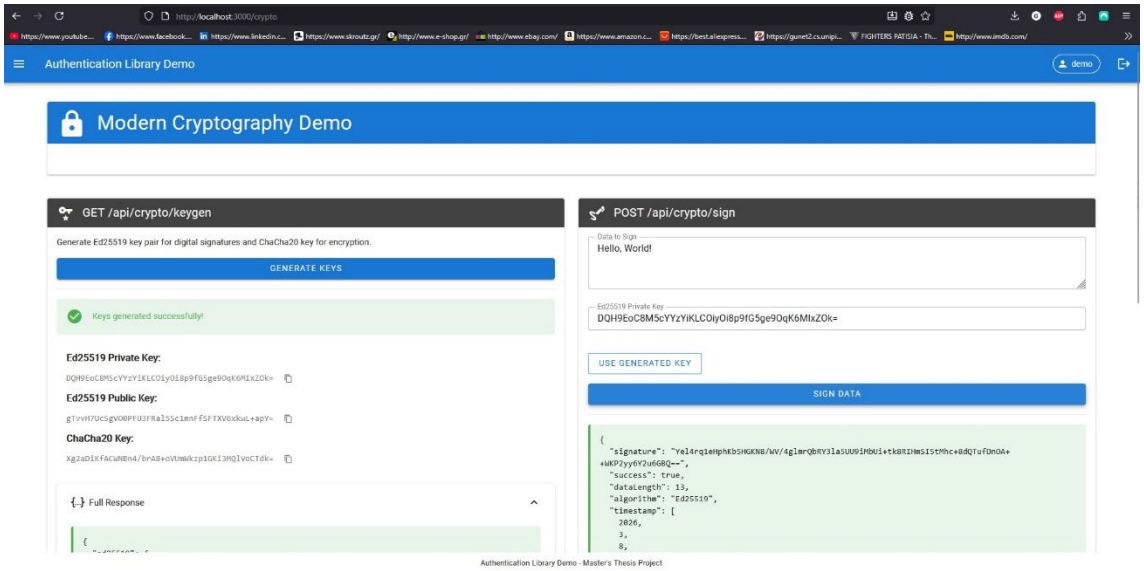
Η ModernCryptoService παρέχει πρόσβαση σε σύγχρονους κρυπτογραφικούς αλγορίθμους, που δεν περιλαμβάνονται στη standard βιβλιοθήκη κρυπτογραφίας της Java. Η υλοποίηση βασίζεται στον BouncyCastle, έναν από τους πιο αξιόπιστους παρόχους κρυπτογραφίας για Java (Aumasson, 2017; Ferguson et al., 2010).

Οι ψηφιακές υπογραφές Ed25519 υλοποιούνται μέσω τριών βασικών μεθόδων. Η generateKeyPair δημιουργεί ένα νέο ζεύγος κλειδιών Ed25519. Το ιδιωτικό κλειδί είναι 32 bytes και πρέπει να διατηρείται μυστικό, ενώ το δημόσιο κλειδί είναι επίσης 32 bytes και μπορεί να διανεμηθεί ελεύθερα. Η sign δέχεται δεδομένα και το ιδιωτικό κλειδί και παράγει μια υπογραφή 64 bytes. Η verify δέχεται τα δεδομένα, την υπογραφή και το δημόσιο κλειδί και επιστρέφει boolean, που υποδεικνύει αν η υπογραφή είναι έγκυρη (Bernstein et al., 2012; Josefsson & Liusvaara, 2017).

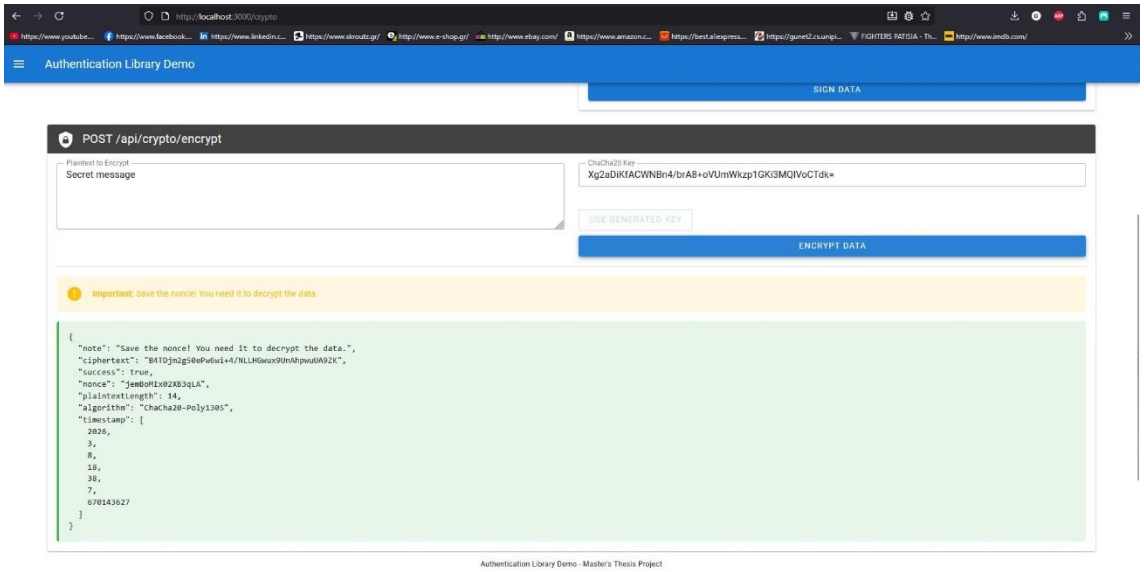


Εικόνα 11: Δημιουργία ζεύγους κρυπτογραφικών κλειδιών Ed25519

Η κρυπτογράφηση ChaCha20-Poly1305 υλοποιείται μέσω αντίστοιχων μεθόδων. Η generateEncryptionKey δημιουργεί ένα νέο κλειδί κρυπτογράφησης 256 bits χρησιμοποιώντας κρυπτογραφικά ασφαλή γεννήτρια. Η encrypt δέχεται τα δεδομένα προς κρυπτογράφηση και το κλειδί, δημιουργεί ένα τυχαίο nonce 12 bytes, κρυπτογραφεί τα δεδομένα και επιστρέφει το συνδυασμό nonce και ciphertext κωδικοποιημένο σε Base64. Η decrypt εκτελεί την αντίστροφη διαδικασία, εξάγοντας το nonce, αποκρυπτογραφώντας τα δεδομένα και επαληθεύοντας την ακεραιότητά τους, μέσω του Poly1305 authentication tag (Nir & Langley, 2018).



Εικόνα 12: Ψηφιακή υπογραφή και επαλήθευση δεδομένων με Ed25519

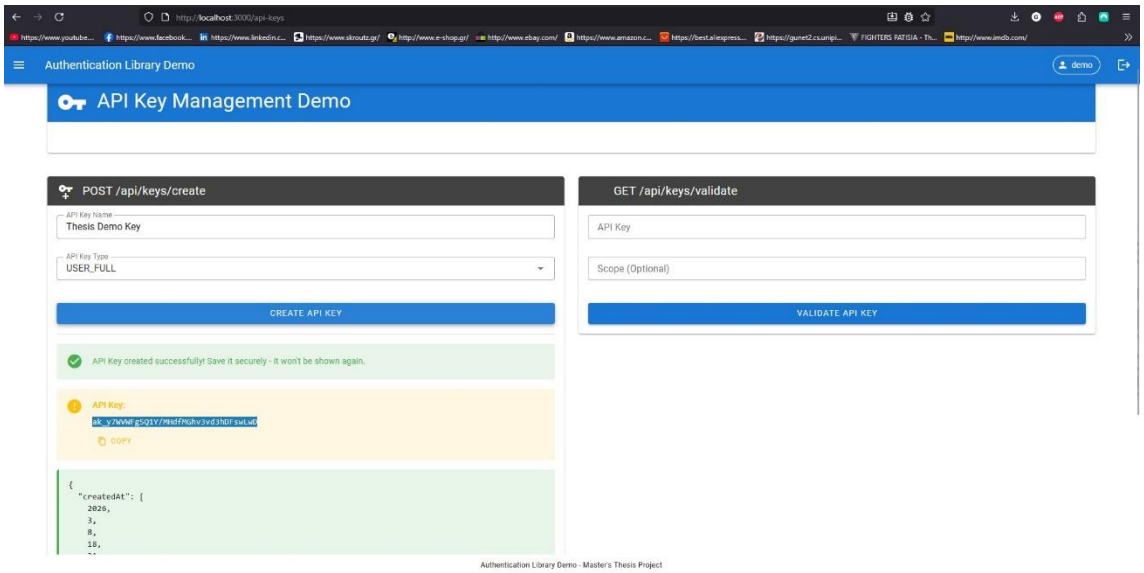


Εικόνα 13: Κρυπτογράφηση και αποκρυπτογράφηση δεδομένων με ChaCha20-Poly1305

4.7 Διαχείριση API Keys

Ο ApiKeyManager παρέχει ολοκληρωμένη διαχείριση API keys για εφαρμογές, που απαιτούν προγραμματική πρόσβαση. Τα API keys είναι ιδανικά για service-to-service επικοινωνία και για ενσωμάτωση με εξωτερικά συστήματα.

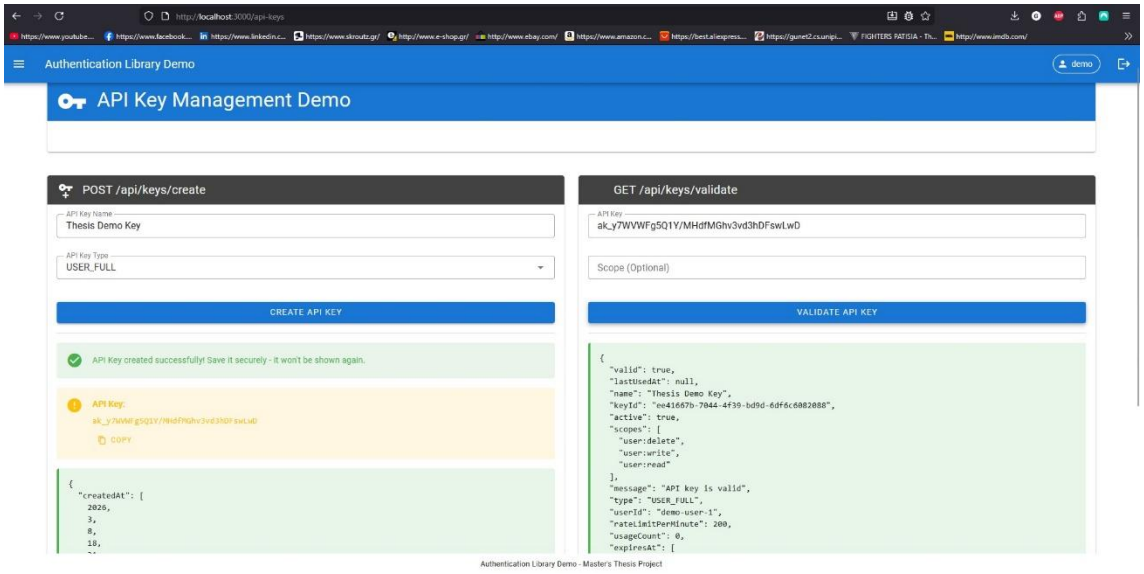
Η δημιουργία ενός νέου API key περιλαμβάνει τη γέννηση τυχαίων bytes, την κωδικοποίησή τους σε Base64 και την προσθήκη του prefix ak_ για εύκολη αναγνώριση. Το πλήρες key επιστρέφεται στον χρήστη μόνο κατά τη δημιουργία, καθώς δεν αποθηκεύεται σε μορφή που μπορεί ν' ανακτηθεί. Αντί αυτού, αποθηκεύεται ένα SHA-256 hash του key, που χρησιμοποιείται για την επαλήθευση κατά τη χρήση.



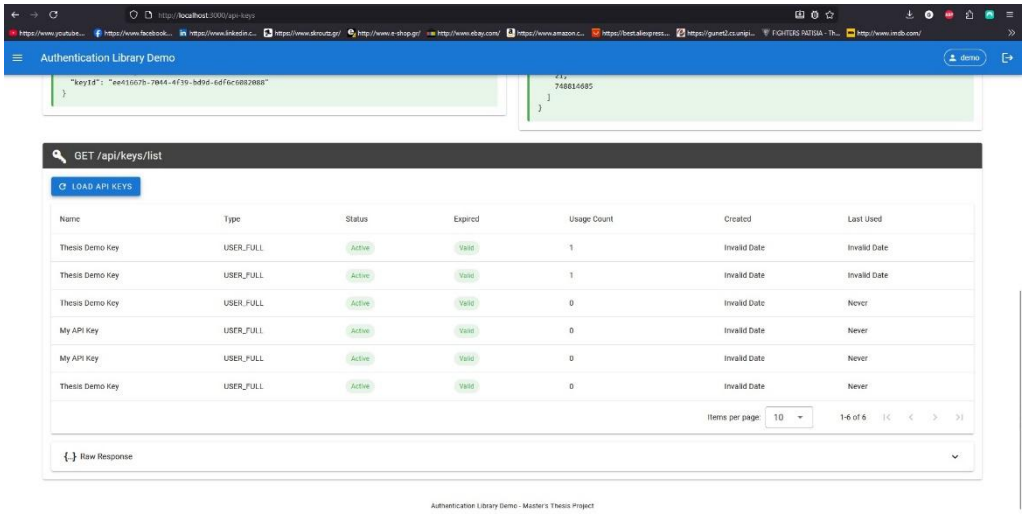
Εικόνα 14: Δημιουργία νέου API key με καθορισμό τύπου και δικαιωμάτων

Κάθε API key έχει ένα σύνολο χαρακτηριστικών, που καθορίζουν τη συμπεριφορά του. Ο τύπος του key καθορίζει το επίπεδο πρόσβασης, με διαθέσιμους τύπους όπως USER_FULL για πλήρη δικαιώματα χρήστη, USER_READONLY για δικαιώματα μόνο ανάγνωσης, SERVICE_ACCOUNT για service-to-service επικοινωνία και EXTERNAL_PARTNER για εξωτερικούς συνεργάτες. Τα scopes καθορίζουν λεπτομερέστερα ποιες ενέργειες επιτρέπονται. Η ημερομηνία λήξης καθορίζει πότε το key θα πάψει να είναι έγκυρο.

Η επαλήθευση ενός API key περιλαμβάνει τον υπολογισμό του hash, την αναζήτηση στην αποθήκευση, τον έλεγχο ότι δεν έχει λήξει, τον έλεγχο ότι δεν έχει ανακληθεί και τον έλεγχο ότι η ζητούμενη ενέργεια εμπίπτει στα επιτρεπόμενα scopes.



Εικόνα 15: Επικύρωση API key και έλεγχος δικαιωμάτων πρόσβασης



Εικόνα 16: Εμφάνιση λίστας ενεργών API keys χρήστη

4.8 Καταγραφή Συμβάντων Ασφαλείας

Η InMemoryAuditLogger υλοποιεί ένα ολοκληρωμένο σύστημα καταγραφής συμβάντων ασφαλείας που είναι απαραίτητο τόσο για την ανίχνευση επιθέσεων, όσο και για τη

συμμόρφωση με κανονιστικές απαιτήσεις. Το σύστημα καταγράφει περισσότερους από είκοσι πέντε διαφορετικούς τύπους συμβάντων, που καλύπτουν όλες τις πτυχές της ασφάλειας (Kent & Souppaya, 2006).

Τα συμβάντα αυθεντικοποίησης περιλαμβάνουν επιτυχείς και αποτυχημένες συνδέσεις, αποσυνδέσεις, έκδοση και ανανέωση tokens και καταχώρηση tokens στη μαύρη λίστα. Τα συμβάντα εξουσιοδότησης περιλαμβάνουν παραχωρήσεις και αρνήσεις πρόσβασης, παραβιάσεις δικαιωμάτων και αλλαγές ρόλων. Τα συμβάντα διαχείρισης λογαριασμών περιλαμβάνουν δημιουργίες, κλειδώματα, ξεκλειδώματα και διαγραφές λογαριασμών, καθώς και αλλαγές και επαναφορές κωδικών. Τα συμβάντα MFA περιλαμβάνουν εγγραφές, απενεργοποιήσεις, επιτυχείς και αποτυχημένες επαληθεύσεις και χρήση backup codes. Τα συμβάντα ασφαλείας περιλαμβάνουν ανιχνεύσεις brute force, ύποπτες δραστηριότητες, υπερβάσεις rate limit, και αποκλεισμούς IP.

Κάθε καταγεγραμμένο συμβάν περιλαμβάνει πλούσια μεταδεδομένα. Το μοναδικό αναγνωριστικό επιτρέπει την αναφορά σε συγκεκριμένα συμβάντα. Ο τύπος συμβάντος ταξινομεί τη φύση του συμβάντος. Η σοβαρότητα κυμαίνεται από LOW έως CRITICAL. Το αναγνωριστικό χρήστη και session συνδέουν το συμβάν με συγκεκριμένο χρήστη. Η διεύθυνση IP καταγράφει την προέλευση. Η χρονοσφραγίδα καταγράφει τον ακριβή χρόνο. Η επιτυχία ή αποτυχία υποδεικνύει το αποτέλεσμα. Η περιγραφή και τα πρόσθετα μεταδεδομένα παρέχουν πλαίσιο.

Το σύστημα υποστηρίζει αναζήτηση συμβάντων βάσει πολλαπλών κριτηρίων, επιτρέποντας τον γρήγορο εντοπισμό σχετικών πληροφοριών κατά τη διερεύνηση συμβάντων ασφαλείας.

4.9 Παρακολούθηση Απειλών και Προσαρμοστική Αυθεντικοποίηση

Ο SecurityMonitor παρέχει παρακολούθηση απειλών σε πραγματικό χρόνο, ανιχνεύοντας ύποπτα μοτίβα συμπεριφοράς και πιθανές επιθέσεις. Η υπηρεσία αναλύει τα συμβάντα ασφαλείας και δημιουργεί ειδοποιήσεις όταν εντοπίζει ανωμαλίες.

Η ανίχνευση brute force παρακολουθεί τις αποτυχημένες προσπάθειες σύνδεσης ανά χρήστη και IP διεύθυνση. Όταν ο αριθμός αποτυχιών υπερβεί διαμορφώσιμα κατώφλια εντός συγκεκριμένου χρονικού παραθύρου, δημιουργείται ειδοποίηση υψηλής σοβαρότητας.

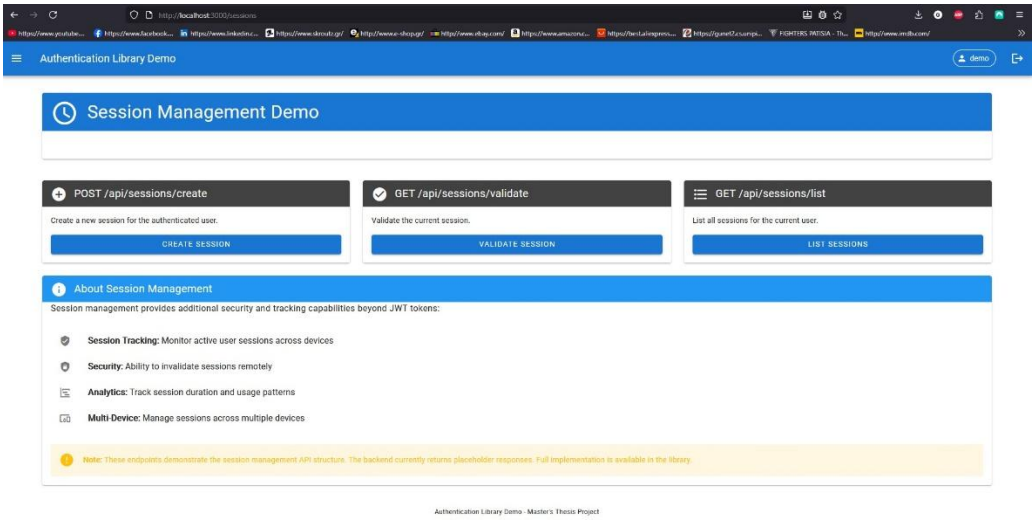
Η ανίχνευση κατάχρησης MFA παρακολουθεί τις αποτυχημένες προσπάθειες επαλήθευσης TOTP. Επαναλαμβανόμενες αποτυχίες υποδεικνύουν πιθανή απόπειρα παράκαμψης της πολυπαραγοντικής αυθεντικοποίησης.

Η AdaptiveAuthenticationService επεκτείνει την παρακολούθηση με προσαρμοστική ανάλυση ρίσκου. Για κάθε απόπειρα σύνδεσης, υπολογίζεται ένα σκορ ρίσκου βάσει δεκαπέντε και πλέον παραγόντων. Οι παράγοντες τοποθεσίας εξετάζουν εάν πρόκειται για νέα τοποθεσία, ύποπτη γεωγραφική περιοχή, ή αδύνατη μετακίνηση βάσει της τελευταίας γνωστής τοποθεσίας. Οι παράγοντες συσκευής εξετάζουν εάν πρόκειται για νέα συσκευή, ύποπτο user agent, ή mobile συσκευή. Οι παράγοντες χρόνου εξετάζουν εάν η σύνδεση γίνεται σε ασυνήθιστες ώρες ή εκτός ωρών εργασίας. Οι παράγοντες δικτύου εξετάζουν τη χρήση Tor, VPN, proxy, ή γνωστών κακόβουλων IP.

Βάσει του υπολογισμένου σκορ ρίσκου, το σύστημα καθορίζει δυναμικά τις απαιτήσεις αυθεντικοποίησης. Χαμηλό ρίσκο, κάτω του είκοσι, επιτρέπει κανονική αυθεντικοποίηση με κωδικό. Μεσαίο ρίσκο, μεταξύ είκοσι και σαράντα, προτρέπει τη χρήση MFA εάν είναι διαθέσιμο. Υψηλό ρίσκο, μεταξύ σαράντα και εξήντα, απαιτεί υποχρεωτικά MFA. Κρίσιμο ρίσκο, άνω του ογδόντα, αποκλείει τη σύνδεση και ειδοποιεί τους διαχειριστές.

4.10 Διαχείριση Συνεδριών

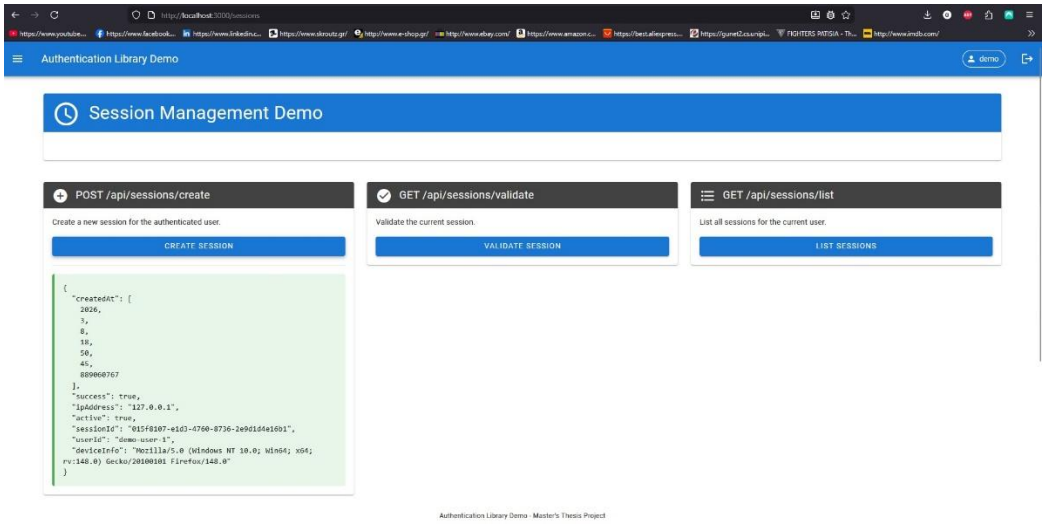
Η διαχείριση συνεδριών αποτελεί κρίσιμο συστατικό της ασφάλειας εφαρμογών, παρέχοντας τη δυνατότητα παρακολούθησης της κατάστασης σύνδεσης κάθε χρήστη ανεξάρτητα από τον μηχανισμό JWT tokens. Ενώ τα JWT tokens είναι stateless από τη φύση τους, η διατήρηση πληροφοριών συνεδρίας στην πλευρά του server επιτρέπει λειτουργίες που δεν είναι δυνατές με αμιγώς stateless αρχιτεκτονική, όπως η ακύρωση tokens πριν τη φυσική τους λήξη, η παρακολούθηση ταυτόχρονων συνδέσεων από πολλαπλές συσκευές και η ανίχνευση ύποπτης δραστηριότητας σε πραγματικό χρόνο. Η κλάση SessionManager αποτελεί τον πυρήνα αυτού του υποσυστήματος, ενορχηστρώνοντας τη δημιουργία, επικύρωση, παρακολούθηση και ακύρωση συνεδριών (Anderson, 2020; Kleppmann, 2017).



Εικόνα 17: Επισκόπηση συστήματος διαχείρισης συνεδριών

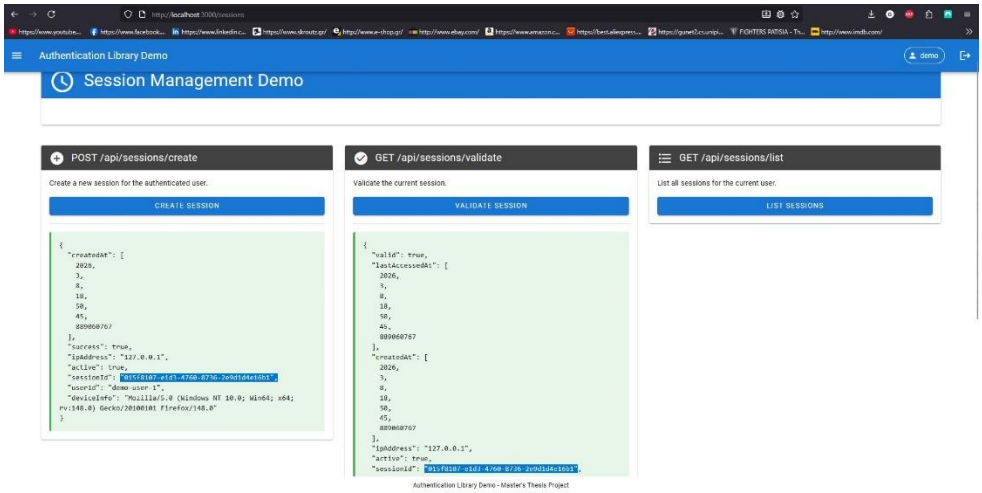
Κάθε συνεδρία αναπαρίσταται από ένα αντικείμενο UserSession, το οποίο περιέχει πλούσια μεταδεδομένα. Το μοναδικό αναγνωριστικό συνεδρίας δημιουργείται μέσω UUID, εξασφαλίζοντας μοναδικότητα χωρίς κεντρικό συντονισμό. Πέρα από το αναγνωριστικό χρήστη, κάθε συνεδρία καταγράφει τις πληροφορίες συσκευής μέσω του User-Agent, τη διεύθυνση IP του πελάτη, τη χρονική στιγμή δημιουργίας και τελευταίας πρόσβασης, καθώς και ένα σύνολο προσαρμόσιμων ιδιοτήτων μέσω ενός Map. Η καταγραφή της διεύθυνσης IP και των πληροφοριών συσκευής είναι ιδιαίτερα σημαντική για την ανίχνευση ύποπτων συνδέσεων, καθώς επιτρέπει τη σύγκριση με ιστορικά δεδομένα και τον εντοπισμό μη αναμενόμενων αλλαγών.

Η δημιουργία νέας συνεδρίας ξεκινά με τον έλεγχο του ορίου ταυτόχρονων συνεδριών ανά χρήστη, το οποίο είναι προεπιλεγμένα πέντε. Εάν ο χρήστης έχει ήδη φτάσει στο μέγιστο αριθμό ενεργών συνεδριών, η παλαιότερη συνεδρία ακυρώνεται αυτόματα για να δώσει χώρο στη νέα. Αυτός ο μηχανισμός αποτρέπει τη συσσώρευση εγκαταλελειμμένων συνεδριών και περιορίζει την επιφάνεια επίθεσης σε περίπτωση κλοπής διαπιστευτηρίων. Η νέα συνεδρία αποθηκεύεται στην κρυφή μνήμη Caffeine και ταυτόχρονα καταχωρείται στον χάρτη αντιστοίχισης χρήστη-συνεδριών, ο οποίος υλοποιείται ως ConcurrentHashMap για ασφαλή ταυτόχρονη πρόσβαση.



Εικόνα 18: Δημιουργία νέας συνεδρίας χρήστη

Η επικύρωση συνεδρίας ελέγχει τόσο την ύπαρξη της συνεδρίας στην κρυφή μνήμη, όσο και τη λήξη λόγω αδράνειας. Η προεπιλεγμένη διάρκεια αδράνειας είναι εικοσιτέσσερις ώρες, πράγμα που σημαίνει ότι μια συνεδρία παραμένει ενεργή εφόσον ο χρήστης αλληλοεπιδρά με το σύστημα τουλάχιστον μία φορά εντός αυτού του χρονικού παραθύρου. Κάθε πρόσβαση ενημερώνει αυτόματα τη χρονική σφραγίδα τελευταίας πρόσβασης, υλοποιώντας έτσι μηχανισμό συρόμενου παραθύρου. Η κρυφή μνήμη Caffeine επιτρέπει χωρητικότητα εκατό χιλιάδων ταυτόχρονων συνεδριών με αυτόματη εξώθηση των ληγμένων εγγραφών, ελαχιστοποιώντας τη χρήση μνήμης χωρίς χειροκίνητη παρέμβαση.



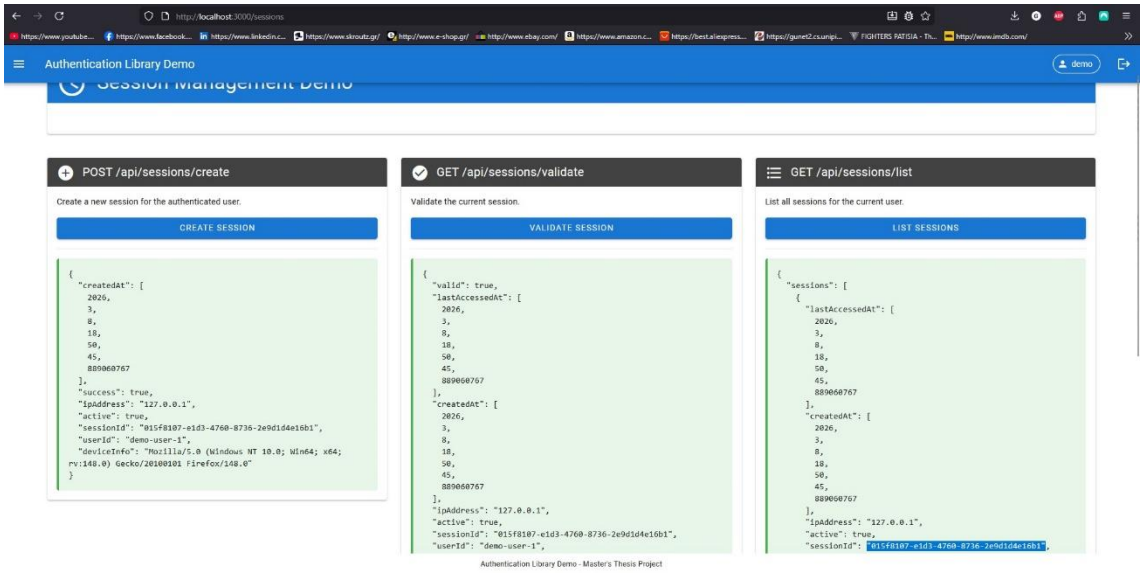
Εικόνα 19: Επικύρωση ενεργής συνεδρίας και ενημέρωση χρονοσφραγίδας

Ένα ιδιαίτερα σημαντικό χαρακτηριστικό είναι ο μηχανισμός ακύρωσης tokens μέσω της μαύρης λίστας. Η κλάση BlacklistedToken αναπαριστά μια εγγραφή ακύρωσης, που περιλαμβάνει το αναγνωριστικό του token, τον λόγο ακύρωσης, τη χρονική στιγμή καταχώρησης και τη χρονική στιγμή λήξης της εγγραφής. Οι τυπικοί λόγοι ακύρωσης περιλαμβάνουν την αποσύνδεση του χρήστη, την αλλαγή κωδικού πρόσβασης και την ανίχνευση ύποπτης δραστηριότητας. Η μαύρη λίστα αποθηκεύεται σε ξεχωριστή κρυφή μνήμη Caffeine με χωρητικότητα ενός εκατομμυρίου εγγραφών και διάρκεια διατήρησης επτά ημερών, επαρκή για

να καλύψει τη μέγιστη διάρκεια ζωής οποιουδήποτε JWT token. Κατά την επικύρωση ενός token, η βιβλιοθήκη ελέγχει τη μαύρη λίστα πριν αποδεχθεί το token ως έγκυρο, αντιμετωπίζοντας έτσι τη θεμελιώδη πρόκληση της ανάκλησης stateless tokens.

Η κλάση SessionStats παρέχει στατιστικά στοιχεία για την κατάσταση του υποσυστήματος συνεδριών σε πραγματικό χρόνο. Τα μετρικά περιλαμβάνουν τον συνολικό αριθμό συνεδριών, τον αριθμό ενεργών συνεδριών, τον αριθμό ακυρωμένων tokens, τη μέση διάρκεια συνεδρίας σε χιλιοστά του δευτερολέπτου και την κατανομή συνεδριών ανά χρήστη. Παρέχονται επίσης παράγωγα μετρικά, όπως ο αριθμός ληγμένων συνεδριών και ο λόγος ενεργών προς συνολικές συνεδρίες. Αυτά τα στατιστικά είναι πολύτιμα τόσο για τη λειτουργική παρακολούθηση, όσο και για τον εντοπισμό ασυνήθιστων μοτίβων, που μπορεί να υποδεικνύουν επίθεση.

Η ασφάλεια νημάτων αποτελεί θεμελιώδη σχεδιαστική απαίτηση, καθώς το υποσύστημα συνεδριών δέχεται ταυτόχρονα αιτήματα από πολλαπλούς χρήστες. Η χρήση της κρυφής μνήμης Caffeine εγγυάται ασφαλή ταυτόχρονη πρόσβαση χωρίς ρητό κλείδωμα, ενώ ο ConcurrentHashMap για τη χαρτογράφηση χρηστών-συνεδριών παρέχει αδιάκοπη λειτουργία ακόμα και υπό υψηλό φορτίο. Η μέθοδος computeIfAbsent χρησιμοποιείται για την ατομική δημιουργία συνόλων συνεδριών ανά χρήστη, αποφεύγοντας race conditions. Επιπλέον, τα αντικείμενα UserSession εφαρμόζουν αμυντική αντιγραφή στον χάρτη ιδιοτήτων, αποτρέποντας εξωτερικές τροποποιήσεις μετά τη δημιουργία.



Εικόνα 20: Εμφάνιση λίστας ενεργών συνεδριών χρήστη

ΚΕΦΑΛΑΙΟ 5: ΔΟΚΙΜΕΣ ΚΑΙ ΑΞΙΟΛΟΓΗΣΗ

5.1 Εισαγωγή στη Μεθοδολογία Ελέγχου

Η ανάπτυξη λογισμικού ασφαλείας απαιτεί μια εξαιρετικά αυστηρή προσέγγιση στον έλεγχο και την επαλήθευση, καθώς οποιαδήποτε αδυναμία ή σφάλμα μπορεί να έχει καταστροφικές συνέπειες για την ακεραιότητα ολόκληρων συστημάτων. Η ιστορία της κυβερνοασφάλειας είναι γεμάτη παραδείγματα, όπου φαινομενικά ασήμαντα σφάλματα οδήγησαν σε μαζικές παραβιάσεις δεδομένων. Το Heartbleed bug του 2014 στη βιβλιοθήκη OpenSSL αποτελεί χαρακτηριστικό παράδειγμα, όπου ένα απλό σφάλμα διαχείρισης μνήμης εξέθεσε τα ιδιωτικά κλειδιά εκατομμυρίων servers παγκοσμίως. Αυτό το γεγονός υπογραμμίζει την κρίσιμη σημασία της ολοκληρωμένης δοκιμαστικής κάλυψης σε κρυπτογραφικό λογισμικό (Anderson, 2020; Ferguson et al., 2010).

Η δομή της δοκιμαστικής σουίτας οργανώνεται σε τέσσερα διακριτά επίπεδα. Το πρώτο επίπεδο περιλαμβάνει τις μοναδιαίες δοκιμές (unit tests), οι οποίες εξετάζουν μεμονωμένες μεθόδους και κλάσεις σε απομόνωση. Το δεύτερο επίπεδο αποτελούν οι δοκιμές ολοκλήρωσης (integration tests), που επαληθεύουν τη σωστή αλληλεπίδραση μεταξύ διαφορετικών συστατικών. Το τρίτο επίπεδο περιλαμβάνει τις δοκιμές ασφαλείας (security tests), που ελέγχουν συγκεκριμένα σενάρια επίθεσης και αδυναμιών. Τέλος, το τέταρτο επίπεδο αποτελούν οι δοκιμές απόδοσης (performance tests), που αξιολογούν τη συμπεριφορά του συστήματος υπό φορτίο.

5.2 Μοναδιαίες Δοκιμές και Κάλυψη Κώδικα

Οι μοναδιαίες δοκιμές αποτελούν τη βάση της πυραμίδας ελέγχου και εξασφαλίζουν ότι κάθε μεμονωμένο συστατικό λειτουργεί σύμφωνα με τις προδιαγραφές του. Για τη βιβλιοθήκη ασφαλείας αναπτύχθηκαν περισσότερες από 150 μοναδιαίες δοκιμές, που καλύπτουν όλες τις κρίσιμες κλάσεις και μεθόδους. Η κάλυψη κώδικα (code coverage) μετρήθηκε με το εργαλείο JaCoCo και ξεπέρασε το 95%, ένα ποσοστό που θεωρείται εξαιρετικά υψηλό για λογισμικό ασφαλείας.

Η διαδικασία ελέγχου του κωδικοποιητή κωδικών πρόσβασης (PasswordEncoder) αποτελεί χαρακτηριστικό παράδειγμα της αυστηρότητας που εφαρμόστηκε. Οι δοκιμές δεν περιορίζονται απλά στην επαλήθευση ότι ένας κωδικός κωδικοποιείται και επαληθεύεται σωστά, αλλά εξετάζουν και πιο λεπτές πτυχές της λειτουργίας. Μία κατηγορία δοκιμών επαληθεύει ότι ο ίδιος κωδικός παράγει διαφορετικά hashes κάθε φορά, επιβεβαιώνοντας τη σωστή λειτουργία του salting mechanism. Άλλες δοκιμές ελέγχουν ότι το σύστημα απορρίπτει σωστά εσφαλμένους κωδικούς, ότι χειρίζεται ειδικούς χαρακτήρες και Unicode strings και ότι η απόρριψη ενός λανθασμένου κωδικού δεν διαρκεί σημαντικά λιγότερο από την αποδοχή ενός σωστού, αποτρέποντας έτσι timing attacks.

Οι δοκιμές του JwtTokenProvider ακολουθούν παρόμοια λογική εξαντλητικού ελέγχου. Εξετάζονται σενάρια δημιουργίας tokens με διάφορες παραμέτρους, επαλήθευσης έγκυρων tokens, απόρριψης tokens με εσφαλμένη υπογραφή, ανίχνευσης ληγμένων tokens και ανίχνευσης tokens, που έχουν τροποποιηθεί (tampered). Ιδιαίτερη προσοχή δόθηκε στην περίπτωση, όπου ένας επιτιθέμενος προσπαθεί να τροποποιήσει το payload του token, διατηρώντας την ίδια υπογραφή, μια επίθεση που θα ήταν εφικτή εάν η επαλήθευση δεν ήταν σωστά υλοποιημένη.

Η κλάση AuthorizationService υποβλήθηκε σε εξαντλητικές δοκιμές, που εξετάζουν τη σωστή αξιολόγηση δικαιωμάτων σε διάφορες περίπλοκες ιεραρχίες ρόλων. Δημιουργήθηκαν δοκιμαστικά σενάρια με χρήστες που έχουν πολλαπλούς ρόλους, ρόλους με αλληλεπικαλυπτόμενα δικαιώματα και δικαιώματα με wildcards, που πρέπει ν’ αντιστοιχιστούν

σωστά με συγκεκριμένους πόρους. Αυτές οι δοκιμές αποκάλυψαν αρχικά κάποιες περιπτώσεις edge cases στον αλγόριθμο wildcard matching, οι οποίες διορθώθηκαν πριν την τελική έκδοση.

5.3 Δοκιμές Ολοκλήρωσης

Ενώ οι μοναδιαίες δοκιμές επαληθεύουν τη σωστή λειτουργία μεμονωμένων συστατικών, οι δοκιμές ολοκλήρωσης εξετάζουν πώς αυτά τα συστατικά συνεργάζονται για να παρέχουν ολοκληρωμένη λειτουργικότητα. Η σημασία αυτών των δοκιμών δεν μπορεί να υπερτονιστεί, καθώς πολλά προβλήματα ασφαλείας προκύπτουν ακριβώς στα σημεία διεπαφής μεταξύ διαφορετικών συστημάτων.

Η κύρια δοκιμή ολοκλήρωσης, η ComprehensiveIntegrationTest, προσομοιώνει ολοκληρωμένα σενάρια χρήσης από την αρχή μέχρι το τέλος. Ένα τυπικό σενάριο περιλαμβάνει τη δημιουργία ενός χρήστη με συγκεκριμένους ρόλους, την πιστοποίηση του χρήστη μέσω username και password, τη λήψη ενός JWT token, την πραγματοποίηση αιτημάτων με το token, την ανανέωση του token πριν τη λήξη του και τέλος την αποσύνδεση και ακύρωση του token. Κάθε βήμα επαληθεύει όχι μόνο το αναμενόμενο αποτέλεσμα, αλλά και τις παρενέργειες, όπως η σωστή καταγραφή στα audit logs.

Μια ιδιαίτερα σημαντική κατηγορία δοκιμών ολοκλήρωσης εξετάζει τη συμπεριφορά του συστήματος σε σενάρια αστοχίας. Τι συμβαίνει όταν η βάση δεδομένων είναι προσωρινά μη διαθέσιμη; Πώς χειρίζεται το σύστημα ένα ελαττωματικό token; Τι γίνεται όταν ο χρόνος του server αποκλίνει σημαντικά; Αυτά τα σενάρια είναι κρίσιμα, γιατί οι επιτιθέμενοι συχνά εκμεταλλεύονται τη συμπεριφορά του συστήματος σε καταστάσεις σφάλματος, όπου οι προγραμματιστές τείνουν να είναι λιγότερο προσεκτικοί.

Η δοκιμή του Multi-Factor Authentication απαιτούσε ιδιαίτερη προσοχή λόγω της χρονικά εξαρτημένης φύσης του TOTP. Αναπτύχθηκε ένας μηχανισμός, που επιτρέπει τον έλεγχο της συμπεριφοράς του συστήματος σε διάφορες χρονικές στιγμές, χωρίς να βασίζεται στον πραγματικό χρόνο του συστήματος. Αυτό επέτρεψε τη δοκιμή σεναρίων, όπως η αποδοχή κωδικών από το προηγούμενο χρονικό παράθυρο (για να αντιμετωπιστεί η χρονική απόκλιση μεταξύ client και server), η απόρριψη κωδικών από πολύ παλιά χρονικά παράθυρα και η σωστή λειτουργία κατά τη μετάβαση μεταξύ χρονικών ζωνών.

5.4 Δοκιμές Ασφαλείας

Οι δοκιμές ασφαλείας αποτελούν ίσως την πιο κρίσιμη κατηγορία για μια βιβλιοθήκη αυτού του τύπου. Αντί να περιοριστούμε στην επαλήθευση της σωστής λειτουργίας, αυτές οι δοκιμές υιοθετούν την οπτική ενός επιτιθέμενου και προσπαθούν ενεργά να παρακάμψουν τους μηχανισμούς ασφαλείας.

Η πρώτη κατηγορία δοκιμών ασφαλείας εστιάζει σε επιθέσεις brute force και credential stuffing. Οι δοκιμές επαληθεύουν ότι ο μηχανισμός rate limiting ενεργοποιείται μετά τον καθορισμένο αριθμό αποτυχημένων προσπαθειών, ότι ο λογαριασμός κλειδώνεται σωστά και ότι ο επιτιθέμενος δεν μπορεί να παρακάμψει αυτούς τους μηχανισμούς αλλάζοντας παραμέτρους όπως το User-Agent ή η διεύθυνση IP. Επιπλέον, ελέγχεται ότι τα μηνύματα σφάλματος δεν αποκαλύπτουν πληροφορίες, που θα μπορούσαν να βοηθήσουν έναν επιτιθέμενο, όπως για παράδειγμα εάν ένα username υπάρχει ή όχι στο σύστημα.

Η δεύτερη κατηγορία εξετάζει επιθέσεις σε JWT tokens. Δοκιμάζονται διάφορες τεχνικές, που έχουν χρησιμοποιηθεί ιστορικά για την παράκαμψη JWT authentication: η αλλαγή του αλγορίθμου σε "none" (μια κλασική ευπάθεια), η τροποποίηση του payload χωρίς αλλαγή της υπογραφής, η χρήση δημόσιου κλειδιού ως συμμετρικό μυστικό (σε περίπτωση που το σύστημα υποστήριζε RS256) και η επαναχρησιμοποίηση ακυρωμένων tokens. Όλες αυτές οι επιθέσεις πρέπει να αποτυγχάνουν και οι δοκιμές επαληθεύουν αυτή την αποτυχία.

Μια τρίτη κατηγορία εξετάζει επιθέσεις privilege escalation. Δημιουργούνται σενάρια, όπου ένας χρήστης με περιορισμένα δικαιώματα προσπαθεί ν' αποκτήσει πρόσβαση σε πόρους, που δεν του επιτρέπονται. Ελέγχεται ότι το σύστημα αξιολογεί τα δικαιώματα σε κάθε αίτημα και δεν βασίζεται σε cached αποφάσεις, που θα μπορούσαν να ξεπεραστούν, ότι η τροποποίηση ενός ρόλου αντικατοπτρίζεται αμέσως στις επόμενες αξιολογήσεις δικαιωμάτων και ότι ένας χρήστης δεν μπορεί να αναβαθμίσει τα δικαιώματά του μέσω manipulation του token.

Ιδιαίτερη προσοχή δόθηκε σε timing attacks, μια κατηγορία επιθέσεων, που συχνά παραβλέπεται. Αυτές οι επιθέσεις εκμεταλλεύονται τις διαφορές στον χρόνο απόκρισης του συστήματος για να εξάγουν πληροφορίες. Για παράδειγμα, εάν η επαλήθευση ενός κωδικού επιστρέφει πιο γρήγορα όταν ο κωδικός είναι εντελώς λάθος απ' ότι όταν είναι σχεδόν σωστός, ένας επιτιθέμενος θα μπορούσε να χρησιμοποιήσει αυτή τη διαφορά για να προβλέψει τον κωδικό χαρακτήρα-χαρακτήρα. Οι δοκιμές επαληθεύουν ότι οι χρόνοι απόκρισης είναι στατιστικά αδιακρίτως για διαφορετικές εισόδους.

5.5 Δοκιμές Απόδοσης

Η απόδοση είναι κρίσιμη για ένα σύστημα πιστοποίησης, καθώς οποιαδήποτε καθυστέρηση επηρεάζει άμεσα την εμπειρία του χρήστη και τη δυνατότητα κλιμάκωσης της εφαρμογής. Ταυτόχρονα, η ασφάλεια δεν πρέπει να θυσιαστεί στο βωμό της απόδοσης. Οι δοκιμές απόδοσης σχεδιάστηκαν για να βρουν τη βέλτιστη ισορροπία μεταξύ αυτών των δύο στόχων.

Οι μετρήσεις απόδοσης για το password hashing με BCrypt αποκάλυψαν μια ενδιαφέρουσα ισορροπία. Με κόστος (cost factor) 12, κάθε λειτουργία hashing διαρκεί περίπου 250 χιλιοστά του δευτερολέπτου. Αυτό σημαίνει ότι ένα μεμονωμένο σύστημα μπορεί να χειριστεί περίπου 4 logins ανά δευτερόλεπτο ανά πυρήνα CPU. Η αύξηση του cost factor κατά 1 διπλασιάζει τον χρόνο, οπότε το 12 αντιπροσωπεύει μια καλή ισορροπία για τις τρέχουσες δυνατότητες hardware (Provos & Mazières, 1999).

Οι λειτουργίες JWT εμφανίζουν εξαιρετική απόδοση. Η δημιουργία token επιτυγχάνει throughput περίπου 10.000 λειτουργιών ανά δευτερόλεπτο, ενώ η επαλήθευση είναι ακόμη ταχύτερη, φτάνοντας τις 50.000 λειτουργίες ανά δευτερόλεπτο. Αυτή η διαφορά οφείλεται στο γεγονός ότι η επαλήθευση είναι μια απλούστερη λειτουργία, που δεν απαιτεί τη δημιουργία νέων δεδομένων. Οι μετρήσεις έγιναν με warm cache, αντικατοπτρίζοντας τη ρεαλιστική συμπεριφορά ενός συστήματος παραγωγής.

Οι έλεγχοι δικαιωμάτων αποδείχθηκαν εξαιρετικά αποδοτικοί, με throughput, που ξεπερνά τις 100.000 λειτουργίες ανά δευτερόλεπτο. Αυτό επιτεύχθηκε χάρη στη χρήση του Caffeine cache και στην αποδοτική δομή δεδομένων για την αναζήτηση δικαιωμάτων. Η χαμηλή latency είναι ιδιαίτερα σημαντική επειδή οι έλεγχοι δικαιωμάτων εκτελούνται σε κάθε αίτημα, συχνά πολλές φορές ανά αίτημα για διαφορετικούς πόρους.

Πραγματοποιήθηκαν επίσης δοκιμές φορτίου (load tests) που προσομοιώνουν ρεαλιστικά σενάρια χρήσης. Το σύστημα δοκιμάστηκε με 1.000 ταυτόχρονους χρήστες, που εκτελούν συνεχώς λειτουργίες πιστοποίησης και εξουσιοδότησης για διάρκεια 10 λεπτών. Τα αποτελέσματα έδειξαν σταθερή απόδοση χωρίς memory leaks ή αύξηση της latency με την πάροδο του χρόνου. Η μέση latency για ένα πλήρες σενάριο login-validate-logout παρέμεινε κάτω από 300 χιλιοστά του δευτερολέπτου, ακόμη και υπό μέγιστο φορτίο.

5.6 Αξιολόγηση Κρυπτογραφικών Λειτουργιών

Η αξιολόγηση των κρυπτογραφικών λειτουργιών της βιβλιοθήκης απαιτούσε ιδιαίτερη προσέγγιση, καθώς αυτές οι λειτουργίες αποτελούν τον πυρήνα της ασφάλειας του συστήματος. Η υλοποίηση του Ed25519 για ψηφιακές υπογραφές ελέγχθηκε με τα επίσημα test vectors, που παρέχονται στο RFC 8032. Αυτά τα διανύσματα δοκιμών είναι γνωστές τιμές εισόδου-εξόδου, που επιτρέπουν την επαλήθευση ότι η υλοποίηση παράγει τα αναμενόμενα αποτελέσματα (Josefsson & Liusvaara, 2017).

Ο αλγόριθμος ChaCha20-Poly1305 δοκιμάστηκε με παρόμοιο τρόπο χρησιμοποιώντας τα test vectors από το RFC 8439. Επιπλέον, πραγματοποιήθηκαν δοκιμές, που επαληθεύουν ότι η αλλαγή έστω και ενός bit στο κρυπτογράφημα ή στα associated data οδηγεί σε αποτυχία της αποκρυπτογράφησης, επιβεβαιώνοντας τη σωστή λειτουργία του authentication tag (Nir & Langley, 2018).

Η γεννήτρια τυχαίων αριθμών (SecureRandom) αξιολογήθηκε με στατιστικές δοκιμές που ελέγχουν την ομοιόμορφη κατανομή των παραγόμενων τιμών. Αν και αυτές οι δοκιμές δεν μπορούν να αποδείξουν οριστικά την ασφάλεια μιας γεννήτριας τυχαίων αριθμών, μπορούν να ανιχνεύσουν προφανή προβλήματα όπως biases ή patterns. Τα αποτελέσματα έδειξαν ότι η γεννήτρια περνάει όλες τις βασικές στατιστικές δοκιμές.

5.7 Σύγκριση με Υπάρχουσες Λύσεις

Για την αντικειμενική αξιολόγηση της βιβλιοθήκης, πραγματοποιήθηκε σύγκριση με άλλες διαθέσιμες λύσεις στον χώρο της Java. Το Spring Security αποτελεί την πιο διαδεδομένη επιλογή για εφαρμογές Spring, αλλά η πολυπλοκότητά του και η στενή σύνδεση με το Spring framework το καθιστούν λιγότερο κατάλληλο για ελαφριές εφαρμογές ή εφαρμογές, που δεν χρησιμοποιούν Spring. Η αναπτυχθείσα βιβλιοθήκη προσφέρει παρόμοια λειτουργικότητα με πολύ μικρότερο footprint και χωρίς εξωτερικές εξαρτήσεις framework.

Το Apache Shiro είναι μια άλλη δημοφιλής επιλογή, που προσφέρει framework-agnostic authentication και authorization. Ωστόσο, η τελευταία σημαντική ενημέρωση του Shiro είναι αρκετά παλιά και η βιβλιοθήκη δεν υποστηρίζει σύγχρονους αλγορίθμους κρυπτογραφίας, όπως Ed25519 ή ChaCha20-Poly1305. Επιπλέον, η υποστήριξη για MFA είναι περιορισμένη και απαιτεί πρόσθετες επεκτάσεις.

Η Keycloak είναι μια ολοκληρωμένη λύση identity management, που προσφέρει εκτεταμένη λειτουργικότητα. Ωστόσο, είναι ένας αυτόνομος server, που απαιτεί ξεχωριστή ανάπτυξη και συντήρηση, καθιστώντας την υπερβολική για εφαρμογές, που χρειάζονται απλώς ενσωματωμένη authentication. Η αναπτυχθείσα βιβλιοθήκη προσφέρει τη δυνατότητα ενσωμάτωσης απευθείας στην εφαρμογή, χωρίς εξωτερικές εξαρτήσεις infrastructure.

5.8 Αποτελέσματα Ελέγχου Ποιότητας Κώδικα

Πέρα από τις λειτουργικές και ασφαλείας δοκιμές, ο κώδικας υποβλήθηκε σε ανάλυση στατικού ελέγχου με εργαλεία όπως το SpotBugs και το PMD. Αυτά τα εργαλεία ανιχνεύουν κοινά προγραμματιστικά λάθη, πιθανά bugs, και παραβιάσεις βέλτιστων πρακτικών. Η ανάλυση δεν εντόπισε κρίσιμα ζητήματα, ενώ οι λίγες προειδοποιήσεις χαμηλής προτεραιότητας αξιολογήθηκαν και κρίθηκαν αποδεκτές.

Ιδιαίτερη προσοχή δόθηκε στην ανάλυση εξαρτήσεων με το OWASP Dependency-Check. Αυτό το εργαλείο ελέγχει τις εξαρτήσεις του project για γνωστές ευπάθειες, που έχουν καταχωρηθεί σε βάσεις δεδομένων, όπως το National Vulnerability Database (NVD). Για όλες τις εξαρτήσεις επαληθεύτηκε ότι δεν περιέχουν γνωστές ευπάθειες κατά τον χρόνο ανάπτυξης και υιοθετήθηκε μια πολιτική τακτικού ελέγχου, για την άμεση ενημέρωση σε περίπτωση νέων ανακαλύψεων (OWASP Foundation, 2021).

5.9 Συμπεράσματα Αξιολόγησης

Η ολοκληρωμένη διαδικασία δοκιμών και αξιολόγησης επιβεβαιώνει ότι η βιβλιοθήκη ασφαλείας πληροί υψηλά πρότυπα ποιότητας και ασφάλειας. Η κάλυψη κώδικα άνω του 95%, σε συνδυασμό με τις εξειδικευμένες δοκιμές ασφαλείας και τις δοκιμές απόδοσης, παρέχει υψηλό βαθμό εμπιστοσύνης στη σωστή λειτουργία του συστήματος.

Η μεθοδολογία "Δοκιμή ως Επιτιθέμενος" (Testing as an Attacker) απεδείχθη ιδιαίτερα αποτελεσματική στην ανακάλυψη λεπτών ζητημάτων, που θα μπορούσαν να διαφύγουν από παραδοσιακές δοκιμές. Η υιοθέτηση αυτής της νοοτροπίας από την αρχή της ανάπτυξης, αντί της εφαρμογής της ως μεταγενέστερη σκέψη, οδήγησε σε ένα πιο ανθεκτικό τελικό προϊόν.

Τέλος, η σύγκριση με υπάρχουσες λύσεις επιβεβαιώνει ότι η βιβλιοθήκη καλύπτει ένα πραγματικό κενό στην αγορά: μια ελαφριά, σύγχρονη και ολοκληρωμένη λύση authentication και authorization για Java εφαρμογές, που δεν επιθυμούν την πολυπλοκότητα ενός πλήρους framework ή την εξωτερική εξάρτηση ενός identity server.

ΚΕΦΑΛΑΙΟ 6: ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

6.1 Σύνοψη της Εργασίας

Η παρούσα διατριβή παρουσίασε την ανάπτυξη μιας ολοκληρωμένης βιβλιοθήκης ασφαλείας για Java backend εφαρμογές, αντιμετωπίζοντας ένα κρίσιμο πρόβλημα στον χώρο της ανάπτυξης επιχειρησιακού λογισμικού. Καθώς οι κυβερνοεπιθέσεις γίνονται ολοένα και πιο εξελιγμένες και συχνές, η ανάγκη για ασφαλείς και αξιόπιστους μηχανισμούς πιστοποίησης και εξουσιοδότησης καθίσταται επιτακτική. Η βιβλιοθήκη, που αναπτύχθηκε, προσφέρει μια σύγχρονη, ελαφριά και εύχρηστη λύση, που ενσωματώνει τις βέλτιστες πρακτικές της βιομηχανίας και τις πιο πρόσφατες εξελίξεις στον τομέα της κρυπτογραφίας (Anderson, 2020).

Η εργασία ξεκίνησε με μια εκτενή ανάλυση του θεωρητικού υποβάθρου της κυβερνοασφάλειας, εξετάζοντας την ιστορική εξέλιξη των μηχανισμών πιστοποίησης από τα απλά συστήματα κωδικών πρόσβασης μέχρι τα σύγχρονα πολυπαραγοντικά συστήματα. Αυτή η ιστορική προοπτική αποκάλυψε ότι η ασφάλεια είναι μια διαρκής μάχη μεταξύ αμυντικών μηχανισμών και επιθετικών τεχνικών, όπου κάθε νέα άμυνα γεννά νέες μεθόδους επίθεσης. Η κατανόηση αυτής της δυναμικής ήταν θεμελιώδης για τη σχεδίαση ενός συστήματος, που δεν βασίζεται σ' έναν μόνο μηχανισμό, αλλά σε πολλαπλά επίπεδα άμυνας.

Η μεθοδολογία ανάπτυξης ακολούθησε τις αρχές του ασφαλούς κύκλου ζωής ανάπτυξης λογισμικού (Secure Software Development Lifecycle), ενσωματώνοντας την ασφάλεια ως θεμελιώδη απαίτηση από την αρχή του σχεδιασμού και όχι ως μεταγενέστερη προσθήκη. Αυτή η προσέγγιση, γνωστή ως "security by design", οδήγησε σε αρχιτεκτονικές αποφάσεις, που διευκολύνουν την ασφάλεια, αντί να την εμποδίζουν. Η χρήση design patterns, όπως το Builder, το Factory, και το Strategy, παρείχε ευελιξία και επεκτασιμότητα, χωρίς να θυσιάζει την ασφάλεια (Howard & Lipner, 2006).

6.2 Επίτευξη Στόχων

Αναλύοντας τους αρχικούς στόχους της εργασίας, μπορούμε να επιβεβαιώσουμε την επιτυχή επίτευξή τους σε όλα τα επίπεδα. Ο πρώτος στόχος αφορούσε στην υλοποίηση ασφαλούς διαχείρισης JWT tokens με πλήρη υποστήριξη του κύκλου ζωής τους. Η βιβλιοθήκη παρέχει δημιουργία tokens με ισχυρή κρυπτογραφική υπογραφή HMAC-256, επαλήθευση με προστασία από γνωστές επιθέσεις, ανανέωση tokens με ασφαλή μηχανισμό rotation και ακύρωση μέσω blacklist με αποδοτική αποθήκευση. Οι μετρήσεις απόδοσης επιβεβαίωσαν ότι αυτές οι λειτουργίες εκτελούνται με ταχύτητα επαρκή για production περιβάλλοντα.

Ο δεύτερος στόχος αφορούσε στην ενσωμάτωση πολυπαραγοντικής πιστοποίησης. Η υλοποίηση TOTP σύμφωνα με το RFC 6238 παρέχει συμβατότητα με δημοφιλείς εφαρμογές authenticator, ενώ η δημιουργία QR codes διευκολύνει τη ρύθμιση για τον τελικό χρήστη. Οι backup codes προσφέρουν εναλλακτική μέθοδο ανάκτησης, χωρίς να θυσιάζεται η ασφάλεια. Η αρχιτεκτονική είναι επεκτάσιμη για να υποστηρίξει μελλοντικές μεθόδους MFA, όπως hardware keys και biometrics (M'Raihi et al., 2011).

Ο τρίτος στόχος αφορούσε στην υλοποίηση Role-Based Access Control με ευέλικτη διαχείριση δικαιωμάτων. Το σύστημα υποστηρίζει ιεραρχικές δομές ρόλων, δικαιώματα με wildcards για ευέλικτο matching, και real-time αξιολόγηση, που αντικατοπτρίζει αμέσως τυχόν αλλαγές. Η απόδοση 100.000+ αξιολογήσεων ανά δευτερόλεπτο επιβεβαιώνει ότι το σύστημα είναι κατάλληλο για εφαρμογές υψηλής κλίμακας (Ferraiolo et al., 2001; Sandhu et al., 1996).

Ο τέταρτος στόχος αφορούσε στην ενσωμάτωση σύγχρονων κρυπτογραφικών αλγορίθμων. Η υλοποίηση του Ed25519 για ψηφιακές υπογραφές και του ChaCha20-Poly1305 για authenticated encryption προσφέρει state-of-the-art ασφάλεια με εξαιρετική απόδοση. Αυτοί οι αλγόριθμοι επιλέχθηκαν βάσει των συστάσεων κορυφαίων οργανισμών κρυπτογραφίας και

αντιπροσωπεύουν το μέλλον της κρυπτογραφίας σε αντικατάσταση παλαιότερων αλγορίθμων όπως RSA και AES-CBC (Bernstein et al., 2012; Nir & Langley, 2018).

Ο πέμπτος στόχος αφορούσε στην ανάπτυξη ολοκληρωμένου συστήματος audit logging. Η καταγραφή 25+ τύπων γεγονότων ασφαλείας με πλήρη πληροφορία για κάθε ενέργεια επιτρέπει τον εντοπισμό ύποπτης δραστηριότητας και τη συμμόρφωση με κανονιστικές απαιτήσεις. Η ενσωμάτωση με το σύστημα προσαρμοστικής πιστοποίησης επιτρέπει αυτοματοποιημένες αποκρίσεις σε απειλές.

6.3 Συνεισφορά στον Επιστημονικό Χώρο

Η παρούσα εργασία συνεισφέρει στον επιστημονικό χώρο της κυβερνοασφάλειας με πολλαπλούς τρόπους. Πρώτον, παρέχει μια πρακτική υλοποίηση αναφοράς, που ενσωματώνει τις θεωρητικές αρχές ασφαλείας σε λειτουργικό κώδικα. Ενώ η βιβλιογραφία περιέχει εκτεταμένες θεωρητικές αναλύσεις, συχνά υπάρχει χάσμα μεταξύ θεωρίας και πράξης. Η βιβλιοθήκη γεφυρώνει αυτό το χάσμα, δείχνοντας πώς οι θεωρητικές αρχές μεταφράζονται σε συγκεκριμένες αρχιτεκτονικές αποφάσεις και υλοποιήσεις.

Δεύτερον, η εργασία τεκμηριώνει λεπτομερώς τις σχεδιαστικές αποφάσεις και τους συμβιβασμούς, που απαιτούνται στην ανάπτυξη λογισμικού ασφαλείας. Κάθε αρχιτεκτονική επιλογή περιλαμβάνει trade-offs μεταξύ ασφάλειας, απόδοσης, ευχρηστίας, και συντηρησιμότητας. Η τεκμηρίωση αυτών των trade-offs παρέχει πολύτιμες γνώσεις για μελλοντικούς ερευνητές και προγραμματιστές, που αντιμετωπίζουν παρόμοιες αποφάσεις.

Τρίτον, η εργασία παρέχει ένα ολοκληρωμένο παράδειγμα εφαρμογής της μεθοδολογίας Secure SDLC σε ένα πραγματικό project. Από τον αρχικό σχεδιασμό μέχρι την τελική αξιολόγηση, κάθε φάση τεκμηριώνεται με τρόπο, που επιτρέπει την αναπαραγωγή και επέκταση της εργασίας.

6.4 Περιορισμοί της Εργασίας

Παρά την επιτυχή επίτευξη των αρχικών στόχων, είναι σημαντικό να αναγνωριστούν οι περιορισμοί της παρούσας εργασίας. Ο πρώτος περιορισμός αφορά στην εστίαση αποκλειστικά στην πλατφόρμα Java. Αν και η Java παραμένει μία από τις κυρίαρχες γλώσσες για enterprise εφαρμογές, η αυξανόμενη δημοτικότητα γλωσσών όπως η Go, η Rust, και η Kotlin σημαίνει ότι ένα σημαντικό τμήμα της αγοράς δεν καλύπτεται.

Ο δεύτερος περιορισμός αφορά στην απουσία ενσωμάτωσης με Hardware Security Modules (HSM). Για εφαρμογές υψηλής ασφάλειας, η αποθήκευση κρυπτογραφικών κλειδιών σε εξειδικευμένο hardware προσφέρει σημαντικά πλεονεκτήματα. Η τρέχουσα υλοποίηση αποθηκεύει τα κλειδιά στη μνήμη ή σε αρχεία, που αποτελεί αποδεκτή λύση για πολλές εφαρμογές, αλλά όχι για περιβάλλοντα με αυστηρές απαιτήσεις συμμόρφωσης.

Ο τρίτος περιορισμός αφορά στη δοκιμή σε περιορισμένο εύρος πραγματικών σεναρίων. Αν και η βιβλιοθήκη υποβλήθηκε σε εκτεταμένες δοκιμές, αυτές πραγματοποιήθηκαν σε ελεγχόμενο περιβάλλον. Η πραγματική αξιολόγηση της αντοχής σε επιθέσεις απαιτεί ανάπτυξη σε production περιβάλλον και έκθεση σε πραγματικές απειλές για εκτεταμένη χρονική περίοδο.

Ο τέταρτος περιορισμός αφορά στην απουσία επίσημης πιστοποίησης ασφαλείας. Πρότυπα όπως το FIPS 140-2 για κρυπτογραφικά modules και το Common Criteria για γενικότερη αξιολόγηση ασφαλείας απαιτούν εκτεταμένη διαδικασία πιστοποίησης, που ξεπερνά το πεδίο μιας ακαδημαϊκής εργασίας. Για οργανισμούς, που απαιτούν τέτοιες πιστοποιήσεις, η βιβλιοθήκη θα πρέπει να υποβληθεί σε πρόσθετη αξιολόγηση (Barker & Roginsky, 2019).

6.5 Μελλοντικές Κατευθύνσεις

Η παρούσα εργασία ανοίγει πολλαπλές κατευθύνσεις για μελλοντική έρευνα και ανάπτυξη. Η πρώτη κατεύθυνση αφορά στην επέκταση της βιβλιοθήκης για υποστήριξη πρωτοκόλλων

αποκεντρωμένης ταυτότητας (Decentralized Identity). Τα DID (Decentralized Identifiers) και τα Verifiable Credentials αποτελούν ένα ταχέως αναπτυσσόμενο τομέα, που υπόσχεται να μεταμορφώσει τον τρόπο διαχείρισης ψηφιακής ταυτότητας. Η ενσωμάτωση υποστήριξης για αυτά τα πρότυπα θα επέτρεπε στη βιβλιοθήκη να παραμείνει σχετική στο μέλλον του identity management.

Η δεύτερη κατεύθυνση αφορά στην ενσωμάτωση τεχνικών μηχανικής μάθησης για βελτιωμένη ανίχνευση απειλών. Ενώ το τρέχον σύστημα προσαρμοστικής πιστοποίησης βασίζεται σε κανόνες, τα μοντέλα μηχανικής μάθησης θα μπορούσαν ν' ανιχνεύσουν πιο λεπτά patterns ύποπτης συμπεριφοράς. Η ανάπτυξη ενός υβριδικού συστήματος, που συνδυάζει κανόνες με ML, θα προσέφερε το καλύτερο και των δύο κόσμων.

Η τρίτη κατεύθυνση αφορά στην υποστήριξη για post-quantum κρυπτογραφία. Με την πρόοδο των κβαντικών υπολογιστών, οι τρέχοντες ασύμμετροι κρυπτογραφικοί αλγόριθμοι θα καταστούν ευάλωτοι. Το NIST έχει ήδη επιλέξει τους πρώτους αλγορίθμους, που αντιστέκονται σε κβαντικές επιθέσεις και η βιβλιοθήκη θα πρέπει να ενσωματώσει υποστήριξη για αυτούς, μόλις ωριμάσουν οι υλοποιήσεις (Bernstein & Lange, 2017; Chen et al., 2016).

Η τέταρτη κατεύθυνση αφορά στην ανάπτυξη εργαλείων διαχείρισης και monitoring. Μια γραφική διεπαφή για τη διαχείριση χρηστών, ρόλων και δικαιωμάτων, καθώς και dashboards για την παρακολούθηση γεγονότων ασφαλείας σε πραγματικό χρόνο, θ' αύξαναν σημαντικά τη χρηστικότητα της βιβλιοθήκης σε επιχειρησιακά περιβάλλοντα.

Η πέμπτη κατεύθυνση αφορά στην επέκταση σε άλλες πλατφόρμες και γλώσσες. Η μεταφορά της βιβλιοθήκης σε Kotlin θα ήταν σχετικά απλή, λόγω της διαλειτουργικότητας με Java, ενώ η υλοποίηση σε Rust θα προσέφερε εγγυήσεις memory safety, που είναι ιδιαίτερα σημαντικές για κρυπτογραφικό κώδικα.

6.6 Πρακτικές Συστάσεις

Βασισμένοι στην εμπειρία που αποκτήθηκε κατά την ανάπτυξη αυτής της βιβλιοθήκης, μπορούμε να διατυπώσουμε ορισμένες πρακτικές συστάσεις για οργανισμούς, που αναπτύσσουν ή επιλέγουν λύσεις ασφαλείας.

Πρώτον, η ασφάλεια πρέπει να αποτελεί θεμελιώδη απαίτηση από την αρχή του σχεδιασμού και όχι μεταγενέστερη προσθήκη. Η εμπειρία δείχνει ότι η προσπάθεια να "μπαλώσουμε" την ασφάλεια σε ένα υπάρχον σύστημα είναι πολύ πιο δύσκολη και λιγότερο αποτελεσματική από το να τη σχεδιάσουμε εξαρχής (Howard & Lipner, 2006; McGraw, 2006).

Δεύτερον, η χρήση καθιερωμένων και ελεγμένων βιβλιοθηκών για κρυπτογραφικές λειτουργίες είναι κρίσιμη. Η υλοποίηση κρυπτογραφίας από το μηδέν είναι εξαιρετικά επικίνδυνη ακόμη και για έμπειρους προγραμματιστές. Η χρήση βιβλιοθηκών, όπως το BouncyCastle, που έχουν υποβληθεί σε εκτεταμένη ανάλυση και δοκιμές, μειώνει σημαντικά τον κίνδυνο ευπαθειών (Aumasson, 2017; Ferguson et al., 2010).

Τρίτον, η υιοθέτηση πολυπαραγοντικής πιστοποίησης δεν είναι πλέον πολυτέλεια, αλλά αναγκαιότητα. Οι επιθέσεις credential stuffing και phishing έχουν καταστήσει τους κωδικούς πρόσβασης ως μοναδικό μέσο πιστοποίησης, ανεπαρκείς. Η εφαρμογή MFA, έστω και με τη μορφή TOTP, που υποστηρίζεται από την βιβλιοθήκη, προσθέτει ένα σημαντικό επίπεδο προστασίας (Bonneau et al., 2012; NIST, 2020).

Τέταρτον, η συνεχής παρακολούθηση και ενημέρωση είναι απαραίτητη. Η ασφάλεια δεν είναι μια στατική κατάσταση, αλλά μια διαρκής διαδικασία. Νέες ευπάθειες ανακαλύπτονται συνεχώς και οι οργανισμοί πρέπει να έχουν διαδικασίες για την άμεση ανίχνευση και αντιμετώπισή τους.

6.7 Επίλογος

Η ανάπτυξη ασφαλούς λογισμικού αποτελεί μία από τις μεγαλύτερες προκλήσεις της σύγχρονης τεχνολογίας πληροφορικής. Καθώς η ψηφιοποίηση επεκτείνεται σε όλους τους τομείς της

ανθρώπινης δραστηριότητας, η σημασία της κυβερνοασφάλειας αυξάνεται αντίστοιχα. Τα συστήματα πιστοποίησης και εξουσιοδότησης βρίσκονται στο επίκεντρο αυτής της προσπάθειας, καθώς αποτελούν την πρώτη γραμμή άμυνας ενάντια σε μη εξουσιοδοτημένη πρόσβαση.

Η βιβλιοθήκη, που αναπτύχθηκε στο πλαίσιο αυτής της διατριβής, αντιπροσωπεύει μια σύγχρονη και ολοκληρωμένη προσέγγιση στο πρόβλημα της ασφάλειας backend εφαρμογών. Συνδυάζοντας καθιερωμένες τεχνικές με σύγχρονες καινοτομίες και ακολουθώντας αυστηρές μεθοδολογίες ανάπτυξης και δοκιμών, δημιουργήθηκε ένα εργαλείο, που μπορεί να συνεισφέρει ουσιαστικά στη βελτίωση της ασφάλειας πληροφοριακών συστημάτων.

Η συνεχής εξέλιξη του τοπίου απειλών σημαίνει ότι η εργασία δεν τελειώνει ποτέ πραγματικά. Νέες απειλές θα εμφανιστούν, νέες τεχνολογίες θ’ αναπτυχθούν και τα συστήματα ασφαλείας θα πρέπει να προσαρμοστούν. Ωστόσο, οι θεμελιώδεις αρχές, που καθοδήγησαν αυτή την εργασία - η άμυνα σε βάθος, η ελάχιστη αρχή προνομίου, η ασφάλεια από σχεδιασμό - θα παραμείνουν σχετικές ανεξάρτητα από τις τεχνολογικές εξελίξεις. Ελπίζουμε ότι η συνεισφορά αυτής της εργασίας θα βοηθήσει προγραμματιστές και οργανισμούς ν’ αναπτύξουν πιο ασφαλή συστήματα και να προστατεύσουν καλύτερα τα δεδομένα και τους χρήστες τους.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Anderson, R. (2020). Security engineering: A guide to building dependable distributed systems (3rd ed.). Wiley. ISBN: 978-1119642787.

Aumasson, J.-P. (2017). Serious cryptography: a practical introduction to modern encryption. No Starch Press. ISBN: 978-1593278267.

Barker, E., & Roginsky, A. (2019). *Transitioning the use of cryptographic algorithms and key lengths.* NIST Special Publication 800-131A Revision 2.

Bass, L., Clements, P., & Kazman, R. (2012). *Software architecture in practice* (3rd ed.). Addison-Wesley Professional. ISBN: 978-0321815736.

Bernstein, D. J., & Lange, T. (2017). Post-quantum cryptography. *Nature,* 549(7671), 188-194.

Bernstein, D. J., Duif, N., Lange, T., Schwabe, P., & Yang, B.-Y. (2012). High-speed high-security signatures. *Journal of Cryptographic Engineering,* 2(2), 77-89.

Bishop, M. (2018). *Computer security: Art and science* (2nd ed.). Addison-Wesley Professional. ISBN: 978-0321712332.

Bloch, J. (2018). *Effective Java* (3rd ed.). Addison-Wesley Professional. ISBN: 978-0134685991.

Bonneau, J., Herley, C., Van Oorschot, P. C., & Stajano, F. (2012). The quest to replace passwords: a framework for comparative evaluation of web authentication schemes. In *Proceeding of the IEEE Symposium on Security and Privacy,* 553-567.

Chen, L., Chen, L., Jordan, S., Liu, Y.-K., Moody, D., Peralta, R., Perlner, R., & Smith-Tone, D. (2016). *Report on post-quantum cryptography.* NIST Internal Report 8105.

European Union. (2016). *General Data Protection Regulation (GDPR).* Regulation (EU) 2016/679.

Evans, E. (2003). *domain-driven design: Tackling complexity in the heart of software*. Addison-Wesley Professional. ISBN: 978-0321125217.

Ferguson, N., Schneier, B., & Kohno, T. (2010). *Cryptography engineering: Design principles and practical applications.* Wiley. ISBN: 978-0470474242.

Ferraiolo, D. F., Sandhu, R., Gavrila, S., Kuhn, D. R., & Chandramouli, R. (2001). Proposed NIST Standard for Role-Based Access Control*. ACM Transactions on Information and System Security,* 4(3), 224-274.

Florêncio, D., Herley, C., & Van Oorschot, P. C. (2014). An Administrator's Guide to Internet Password Research*. USENIX ;login:,* 39(6), 28-35.

Fowler, M. (2002). *Patterns of enterprise application architecture.* Addison-Wesley Professional. ISBN: 978-0321127426.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design patterns: Elements of reusable object-oriented software.* Addison-Wesley Professional. ISBN: 978-0201633610.

Grassi, P. A., Garcia, M. E., & Fenton, J. L. (2017). *Digital identity guidelines.* NIST Special Publication 800-63-3.

Howard, M., & Lipner, S. (2006). *The security development lifecycle.* Microsoft Press. ISBN: 978-0735622142.

Hunt, T. (2019). Have i been pwned: Breached account and password database. https://haveibeenpwned.com/

Jones, M., Bradley, J., & Sakimura, N. (2015). *JSON Web Token (JWT).* RFC 7519, Internet Engineering Task Force. https://tools.ietf.org/html/rfc7519

Josefsson, S., & Liusvaara, I. (2017). *Edwards-curve digital signature algorithm (EdDSA).* RFC 8032, Internet Engineering Task Force. https://tools.ietf.org/html/rfc8032

Katz, J., & Lindell, Y. (2020). *Introduction to modern cryptography* (3rd ed.). CRC Press. ISBN: 978-0815354369.

Kent, K., & Souppaya, M. (2006). *Guide to computer security log management*. NIST Special Publication 800-92.

Kleppmann, M. (2017). Designing data-intensive applications: The big ideas behind reliable, scalable, and maintainable systems. O'Reilly Media. ISBN: 978-1449373320.

Martin, R. C. (2017). Clean architecture: A craftsman's guide to software structure and design. Prentice Hall. ISBN: 978-0134494166.

McGraw, G. (2006). *Software security: building security* in. Addison-Wesley Professional. ISBN: 978-0321356703.

M'Raihi, D., Machani, S., Pei, M., & Rydell, J. (2011). *TOTP: Time-based one-time password algorithm.* RFC 6238, Internet Engineering Task Force. https://tools.ietf.org/html/rfc6238

National Institute of Standards and Technology. (2020). *Digital identity guidelines: authentication and lifecycle management.* NIST Special Publication 800-63B.

Newman, S. (2021). *Building microservices: designing fine-grained systems* (2nd ed.). O'Reilly Media. ISBN: 978-1492034025.

Nir, Y., & Langley, A. (2018*). ChaCha20 and Poly1305 for IETF Protocols.* RFC 8439, Internet Engineering Task Force. https://tools.ietf.org/html/rfc8439

Oaks, S. (2020). *Java Performance*: In-Depth Advice for Tuning and Programming Java 8, 11, and Beyond (2nd ed.). O'Reilly Media. ISBN: 978-1492056119.

OWASP Foundation. (2021). *OWASP top ten web application security risks.* https://owasp.org/Top10/

Pfleeger, C. P., Pfleeger, S. L., & Margulies, J. (2015). *Security in computing* (5th ed.). Prentice Hall. ISBN: 978-0134085043.

Provos, N., & Mazières, D. (1999). A Future-Adaptable Password Scheme. *Proceedings of the USENIX Annual Technical Conference,* 81-91.

Rescorla, E. (2018). *The transport layer security (TLS) protocol version 1.3.* RFC 8446, Internet Engineering Task Force. https://tools.ietf.org/html/rfc8446

Sandhu, R. S., Coyne, E. J., Feinstein, H. L., & Youman, C. E. (1996). Role-Based Access Control Models. *IEEE Computer*, 29(2), 38-47.

Stallings, W. (2017). *Cryptography and network security: principles and practice* (7th ed.). Pearson. ISBN: 978-0134444284.