ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πτυχιακή Εργασία

Τίτλος Πτυχιακής Εργασίας	(Ελληνικά) ***Αρχιτεκτονική και Μηχανισμοί Ασφάλειας του macOS*** (Αγγλικά) Architecture and Security Mechanisms of macOs
Ονοματεπώνυμο Φοιτητή	ΣΤΑΜΑΤΙΟΣ ΣΓΟΥΡΟΣ
Πατρώνυμο	ΚΩΝΣΤΑΝΤΙΝΟΣ-ΣΠΥΡΙΔΩΝ
Αριθμός Μητρώου	Π16124
Επιβλέπων	ΚΟΤΖΑΝΙΚΟΛΑΟΥ ΠΑΝΑΓΙΩΤΗΣ, Καθηγητής

Ημερομηνία Παράδοσης Ιούλιος 2026

Περίληψη

Η παρούσα εργασία πραγματεύεται τη συστηματική και θεωρητικά τεκμηριωμένη ανάλυση της αρχιτεκτονικής ασφάλειας του λειτουργικού συστήματος macOS, προσεγγίζοντάς το ως ένα συνεκτικό και πολυεπίπεδο σύστημα προστασίας. Στο θεωρητικό επίπεδο, η μελέτη εδράζεται στις θεμελιώδεις αρχές της ασφάλειας πληροφοριακών συστημάτων, με έμφαση στην τριάδα εμπιστευτικότητας, ακεραιότητας και διαθεσιμότητας, στο μοντέλο Authentication–Authorization–Accountability (AAA), καθώς και στα σύγχρονα μοντέλα ελέγχου πρόσβασης και στη στρατηγική Defense-in-Depth.

Σε επίπεδο αρχιτεκτονικής, αναλύεται η δομή του Darwin και του υβριδικού πυρήνα XNU, με ιδιαίτερη έμφαση στη διαχείριση εικονικής μνήμης, στην απομόνωση διεργασιών και στη διαβάθμιση προνομίων ως βασικών μηχανισμών επιβολής ασφάλειας. Η εμπιστευτικότητα δεδομένων εξετάζεται μέσω της αλληλεπίδρασης του APFS, της κρυπτογράφησης FileVault, των μηχανισμών Data Protection και του Secure Enclave, αναδεικνύοντας τη σημασία της ιεραρχικής διαχείρισης κλειδιών και της hardware-assisted ασφάλειας. Παράλληλα, το μοντέλο ασφάλειας εφαρμογών αποτιμάται υπό το πρίσμα της αλυσίδας εμπιστοσύνης που συγκροτούν το code signing, ο Gatekeeper, η notarization, το sandboxing και το Hardened Runtime.

Επιπροσθέτως, αναλύονται οι μηχανισμοί διασφάλισης ακεραιότητας του συστήματος, όπως το System Integrity Protection, το signed system volume και η αλυσίδα secure boot, καθώς και το πλαίσιο ελέγχου πρόσβασης σε ευαίσθητους πόρους μέσω του TCC. Η συγκριτική αποτίμηση με άλλα σύγχρονα λειτουργικά συστήματα καταδεικνύει ότι η καθετοποιημένη προσέγγιση του macOS επιτρέπει υψηλό βαθμό ενοποίησης μηχανισμών ασφάλειας, ενισχύοντας τη συνολική ανθεκτικότητα του συστήματος.

Το macOS αναδεικνύεται ως χαρακτηριστικό παράδειγμα σύγχρονης αρχιτεκτονικής ασφάλειας λειτουργικών συστημάτων, όπου η σύζευξη υλικού και λογισμικού, σε συνδυασμό με πολυεπίπεδες πολιτικές ελέγχου, οδηγεί σε αυξημένο επίπεδο προστασίας, χωρίς ωστόσο να εξαλείφει πλήρως τις εγγενείς προκλήσεις της ασφάλειας σε περιβάλλοντα υψηλής πολυπλοκότητας.

Λέξεις-κλειδιά: macOS, ασφάλεια λειτουργικών συστημάτων, εμπιστευτικότητα, ακεραιότητα, Defense-in-Depth, sandboxing, FileVault, Secure Enclave

Abstract

This thesis presents a rigorous and theoretically grounded analysis of the security architecture of macOS, treating it as a coherent, multi-layered protection system. The study is founded on core principles of information system security, including the confidentiality–integrity–availability (CIA) triad, the Authentication–Authorization–Accountability (AAA) model, modern access control paradigms, and the Defense-in-Depth strategy.

From an architectural perspective, the analysis focuses on the Darwin foundation and the hybrid XNU kernel, emphasizing virtual memory management, process isolation, and privilege separation as fundamental enforcement mechanisms. Data confidentiality is examined through the interaction of APFS, FileVault encryption, Data Protection classes, and the Secure Enclave, highlighting the critical role of hierarchical key management and hardware-assisted security. In parallel, the application security model is evaluated through the trust chain established by code signing, Gatekeeper, notarization, sandboxing, and the Hardened Runtime.

Furthermore, system integrity mechanisms are analyzed, including System Integrity Protection (SIP), the signed system volume, and the secure boot chain, along with the Transparency, Consent and Control (TCC) framework governing access to sensitive user resources. A comparative assessment with other contemporary operating systems demonstrates that the vertically integrated design of macOS enables a high degree of security cohesion and enforcement consistency.

In conclusion, macOS emerges as a representative case of modern operating system security architecture, where tight hardware–software integration and layered policy enforcement significantly enhance system resilience, while still being subject to inherent limitations associated with evolving threat models and system complexity.

Keywords: macOS, operating system security, confidentiality, integrity, Defense-in-Depth, sandboxing, FileVault, Secure Enclave

Πίνακας περιεχομένων

Λίστα πινάκων

Λίστα εικόνων

1.Εισαγωγή στην Ασφάλεια Λειτουργικών Συστημάτων

1.1 Η ασφάλεια ως θεμελιώδης απαίτηση των σύγχρονων λειτουργικών συστημάτων

Η ασφάλεια αποτελεί θεμελιώδη απαίτηση των σύγχρονων λειτουργικών συστημάτων, καθώς αυτά λειτουργούν ως ο κεντρικός μεσολαβητής μεταξύ υλικού, λογισμικού και χρήστη. Το λειτουργικό σύστημα διαχειρίζεται κρίσιμους πόρους, όπως τη μνήμη, τον επεξεργαστή, το σύστημα αρχείων και τις διεργασίες, και συνεπώς οποιαδήποτε παραβίαση της ακεραιότητάς του μπορεί να οδηγήσει σε πλήρη έλεγχο του συστήματος από κακόβουλους παράγοντες. Από τις αρχές του 21ου αιώνα, η εκθετική αύξηση της υπολογιστικής ισχύος, η μαζική διασύνδεση μέσω διαδικτύου και η ενσωμάτωση των υπολογιστικών συστημάτων σε κάθε πτυχή της κοινωνικής και οικονομικής δραστηριότητας έχουν αναβαθμίσει την ασφάλεια από δευτερεύον χαρακτηριστικό σε πρωταρχικό σχεδιαστικό στόχο.

Η θεωρητική βάση της ασφάλειας πληροφοριακών συστημάτων εδράζεται στην τριάδα εμπιστευτικότητας, ακεραιότητας και διαθεσιμότητας, η οποία παραμένει διαχρονικά επίκαιρη, παρά τις τεχνολογικές μεταβολές (Shannon, 1948). Στο πλαίσιο των λειτουργικών συστημάτων, οι έννοιες αυτές μεταφράζονται σε μηχανισμούς ελέγχου πρόσβασης, απομόνωσης διεργασιών, προστασίας μνήμης και αξιόπιστης διαχείρισης πόρων. Η ασφάλεια δεν αντιμετωπίζεται πλέον ως προσθήκη εκ των υστέρων, αλλά ως εγγενές χαρακτηριστικό της αρχιτεκτονικής του συστήματος, με έμφαση στην πρόληψη, στον περιορισμό των επιπτώσεων μιας επιτυχούς επίθεσης και στη δυσχέρανση της εκμετάλλευσης ευπαθειών.

Στα σύγχρονα λειτουργικά συστήματα, η ασφάλεια υλοποιείται μέσω πολυεπίπεδων μηχανισμών, οι οποίοι συνεργάζονται σύμφωνα με τη φιλοσοφία της άμυνας σε βάθος. Η απομόνωση της εικονικής μνήμης, η τυχαιοποίηση της διάταξης του χώρου διευθύνσεων (ASLR), η προστασία εκτέλεσης δεδομένων και η αυστηρή διαχείριση προνομίων αποτελούν βασικούς πυλώνες αυτής της προσέγγισης (Herlands et al., 2014). Η αποτελεσματικότητα των μηχανισμών αυτών εξαρτάται σε μεγάλο βαθμό από την ποιότητα της τυχαιότητας και την ορθή υλοποίησή τους, όπως έχει αναδειχθεί από πλήθος ερευνητικών μελετών που εξετάζουν την εντροπία και τη στατιστική ισχύ των τεχνικών τυχαιοποίησης (Acharya et al., 2017; Nemenman et al., 2001).

Η ασφάλεια, ωστόσο, δεν αποτελεί μόνο τεχνικό ζήτημα αλλά και δυναμικό πεδίο αντιπαράθεσης μεταξύ αμυντικών και επιθετικών τεχνικών. Καθώς οι μηχανισμοί προστασίας εξελίσσονται, αναπτύσσονται αντίστοιχα τεχνικές παράκαμψης που στοχεύουν στις αδυναμίες τους. Αυτό έχει οδηγήσει σε μια συνεχή διαδικασία αναθεώρησης και ενίσχυσης των λειτουργικών συστημάτων, με στόχο τη διατήρηση ενός αποδεκτού επιπέδου ασφάλειας σε περιβάλλοντα αυξανόμενης πολυπλοκότητας.

1.2 Σύγχρονες απειλές: malware, privilege escalation, supply chain attacks

Οι σύγχρονες απειλές που αντιμετωπίζουν τα λειτουργικά συστήματα χαρακτηρίζονται από υψηλό βαθμό τεχνικής εξειδίκευσης και συστηματική εκμετάλλευση ευπαθειών σε χαμηλό επίπεδο. Το κακόβουλο λογισμικό έχει εξελιχθεί από απλά αυτοαναπαραγόμενα προγράμματα σε σύνθετα οικοσυστήματα επιθέσεων, τα οποία συνδυάζουν τεχνικές κοινωνικής μηχανικής, εκμετάλλευση σφαλμάτων μνήμης και παράκαμψη μηχανισμών ασφαλείας. Οι επιθέσεις αυτές

συχνά στοχεύουν στην απόκτηση αυξημένων προνομίων, ώστε να αποκτήσουν μόνιμη παρουσία στο σύστημα.

Η κλιμάκωση προνομίων αποτελεί μία από τις πιο κρίσιμες κατηγορίες απειλών, καθώς επιτρέπει σε έναν επιτιθέμενο να μεταβεί από περιορισμένο επίπεδο χρήστη σε πλήρη έλεγχο του συστήματος. Χαρακτηριστικό παράδειγμα αποτελεί η ευπάθεια CVE-2023-4911 στη βιβλιοθήκη glibc, η οποία επέτρεπε τοπική κλιμάκωση προνομίων μέσω χειρισμού παραμέτρων φόρτωσης μνήμης (Qualys Security Advisory, 2023). Παρόμοια περιστατικά καταδεικνύουν ότι ακόμη και θεμελιώδη υποσυστήματα, όπως οι βιβλιοθήκες συστήματος και οι μηχανισμοί διαχείρισης μνήμης, μπορούν να αποτελέσουν σημεία επίθεσης.

Η αποτελεσματικότητα των επιθέσεων κλιμάκωσης προνομίων συχνά συνδέεται με αδυναμίες στους μηχανισμούς τυχαιοποίησης και εντροπίας. Έρευνες έχουν δείξει ότι χαμηλή εντροπία σε κρίσιμες δομές μνήμης διευκολύνει την πρόβλεψη διευθύνσεων και την επιτυχή εκμετάλλευση ευπαθειών (Ding et al., 2014; Kaplan et al., 2014). Παρά τις βελτιώσεις που έχουν εισαχθεί σε σύγχρονα λειτουργικά συστήματα, η πρακτική αποτελεσματικότητα της ASLR εξακολουθεί να εξαρτάται από την αρχιτεκτονική και την υλοποίηση κάθε πλατφόρμας (Marco-Gisbert & Ripoll, 2019).

Ιδιαίτερη σημασία έχουν αποκτήσει τα τελευταία χρόνια οι επιθέσεις εφοδιαστικής αλυσίδας (supply chain attacks), οι οποίες δεν στοχεύουν άμεσα τον τελικό χρήστη, αλλά τα στάδια ανάπτυξης, διανομής και ενημέρωσης λογισμικού. Τέτοιου είδους επιθέσεις αξιοποιούν την εμπιστοσύνη που αποδίδεται σε νόμιμους προμηθευτές λογισμικού, ενσωματώνοντας κακόβουλο κώδικα σε φαινομενικά αξιόπιστες ενημερώσεις ή βιβλιοθήκες. Η αυξανόμενη πολυπλοκότητα των οικοσυστημάτων λογισμικού και η εξάρτηση από τρίτες βιβλιοθήκες ενισχύουν την επιφάνεια επίθεσης, καθιστώντας αναγκαία την αυστηρή επιβολή πολιτικών ελέγχου και επικύρωσης κώδικα.

Η συχνότητα και η σοβαρότητα των ευπαθειών αποτυπώνεται και στα στατιστικά δεδομένα που καταγράφονται σε βάσεις δεδομένων ευπαθειών, τα οποία καταδεικνύουν ότι ακόμη και ώριμα και ευρέως χρησιμοποιούμενα προϊόντα εμφανίζουν σημαντικό αριθμό καταγεγραμμένων αδυναμιών (CVEdetails.com, 2024; SecurityScorecard, 2023). Τα δεδομένα αυτά υπογραμμίζουν ότι η ασφάλεια δεν αποτελεί στατική ιδιότητα, αλλά διαρκή διαδικασία προσαρμογής απέναντι σε ένα μεταβαλλόμενο τοπίο απειλών.

1.3 Ιστορική εξέλιξη της ασφάλειας στα desktop λειτουργικά συστήματα

Η ιστορική εξέλιξη της ασφάλειας στα desktop λειτουργικά συστήματα από τις αρχές της δεκαετίας του 2000 αντικατοπτρίζει τη σταδιακή μετάβαση από ένα μοντέλο περιορισμένης απειλής σε ένα περιβάλλον συνεχούς και συστηματικής επίθεσης. Στις αρχές της περιόδου αυτής, τα περισσότερα συστήματα σχεδιάζονταν με έμφαση στη λειτουργικότητα και την απόδοση, ενώ η ασφάλεια αντιμετωπιζόταν κυρίως μέσω βασικών μηχανισμών ελέγχου πρόσβασης και δικαιωμάτων χρηστών.

Η εισαγωγή της ASLR στα μέσα της δεκαετίας του 2000 αποτέλεσε σημείο καμπής στην προστασία μνήμης, καθώς στόχευε στη δυσχέρανση της εκμετάλλευσης σφαλμάτων τύπου buffer overflow. Οι πρώτες υλοποιήσεις, ωστόσο, παρουσίαζαν περιορισμένη εντροπία και προβλέψιμες δομές, γεγονός που επέτρεπε την ανάπτυξη τεχνικών παράκαμψης (Whitehouse, 2007; Li et al., 2006). Με την πάροδο του χρόνου, οι μηχανισμοί αυτοί ενισχύθηκαν, ιδιαίτερα σε

64-bit αρχιτεκτονικές, αν και έρευνες συνεχίζουν να αναδεικνύουν πρακτικούς περιορισμούς στην πλήρη αποτελεσματικότητά τους (Marco-Gisbert & Ripoll, 2014).

Παράλληλα, η ανάπτυξη πιο σύνθετων επιθέσεων οδήγησε στην ανάγκη για αυστηρότερη απομόνωση διεργασιών και καλύτερη διαχείριση μνήμης. Η εξέλιξη των υποσυστημάτων μνήμης σε λειτουργικά συστήματα Linux και Unix-like πλατφόρμες, καθώς και η εισαγωγή νέων αφαιρέσεων, όπως τα memory folios, αντανακλούν αυτή τη στροφή προς μεγαλύτερη ασφάλεια και αποδοτικότητα (Torvalds, 2021a; Wilcox, 2021).

Κατά τη δεκαετία του 2010, η έμφαση μετατοπίστηκε περαιτέρω προς την ποσοτική αξιολόγηση της ασφάλειας. Έννοιες όπως η «αποτελεσματική εντροπία» προτάθηκαν ως μετρικές για την αποτίμηση της πραγματικής ισχύος των μηχανισμών τυχαιοποίησης, πέρα από τη θεωρητική τους περιγραφή (Herlands et al., 2014). Η προσέγγιση αυτή συνδέει τη θεωρία της πληροφορίας με την πρακτική ασφάλεια, αναδεικνύοντας τη σημασία της στατιστικής ανάλυσης στην αξιολόγηση αμυντικών τεχνικών.

Σήμερα, η ασφάλεια των desktop λειτουργικών συστημάτων αποτελεί αποτέλεσμα μακρόχρονης εξέλιξης και συνεχούς αναπροσαρμογής. Παρά τη σημαντική πρόοδο, η ιστορική εμπειρία δείχνει ότι κάθε νέα τεχνική προστασίας συνοδεύεται από αντίστοιχες προσπάθειες παράκαμψης, καθιστώντας σαφές ότι η ασφάλεια παραμένει μια δυναμική και ανοιχτή διαδικασία, άρρηκτα συνδεδεμένη με την εξέλιξη των απειλών και της τεχνολογίας.

1.4 Γιατί το macOS αποτελεί ιδιαίτερη περίπτωση μελέτης

Το macOS αποτελεί ιδιαίτερη και επιστημονικά ενδιαφέρουσα περίπτωση μελέτης στον χώρο της ασφάλειας λειτουργικών συστημάτων, καθώς συνδυάζει χαρακτηριστικά τόσο των παραδοσιακών Unix-like συστημάτων όσο και ενός αυστηρά ελεγχόμενου, εμπορικού οικοσυστήματος. Σε αντίθεση με άλλα desktop λειτουργικά συστήματα, το macOS αναπτύσσεται και διανέμεται αποκλειστικά από έναν κατασκευαστή, ο οποίος ελέγχει σε μεγάλο βαθμό τόσο το λογισμικό όσο και το υλικό. Αυτή η καθετοποίηση επιτρέπει την υλοποίηση μηχανισμών ασφάλειας που εκτείνονται από το επίπεδο του πυρήνα έως το υλικολογισμικό, δημιουργώντας ένα συνεκτικό μοντέλο άμυνας που δύσκολα απαντάται σε πιο ανοικτά οικοσυστήματα.

Η αρχιτεκτονική του macOS βασίζεται σε Unix θεμέλια, γεγονός που το καθιστά συγκρίσιμο με συστήματα όπως το Linux, ιδιαίτερα ως προς τη διαχείριση μνήμης, τις διεργασίες και τα δικαιώματα χρηστών. Ωστόσο, σε αντίθεση με τις περισσότερες Linux διανομές, το macOS ενσωματώνει αυστηρούς μηχανισμούς ελέγχου εκτέλεσης κώδικα, πιστοποίησης εφαρμογών και επιβολής πολιτικών ασφάλειας, οι οποίοι εφαρμόζονται κεντρικά και ομοιόμορφα. Η ύπαρξη αυτών των μηχανισμών καθιστά το macOS ένα αντιπροσωπευτικό παράδειγμα λειτουργικού συστήματος που υιοθετεί πλήρως τη φιλοσοφία της άμυνας σε βάθος, με έμφαση στην πρόληψη και στον περιορισμό των επιπτώσεων επιτυχών επιθέσεων.

Ιδιαίτερη σημασία έχει και η μετάβαση του macOS σε αρχιτεκτονικές ARM, η οποία συνοδεύτηκε από αλλαγές στον τρόπο διαχείρισης της εικονικής μνήμης και της μετάφρασης διευθύνσεων. Η υλοποίηση της εικονικής διευθυνσιοδότησης και των μηχανισμών προστασίας μνήμης στο macOS συνδέεται άμεσα με τις δυνατότητες των σύγχρονων ARMv8-A αρχιτεκτονικών, γεγονός που επηρεάζει τόσο την αποτελεσματικότητα της ASLR όσο και την ανθεκτικότητα απέναντι σε επιθέσεις χαμηλού επιπέδου (ARM, 2019; Apple, 2024a). Η αλληλεπίδραση μεταξύ υλικού και λειτουργικού συστήματος καθιστά το macOS ιδιαίτερα

κατάλληλο για μελέτη προηγμένων τεχνικών ασφάλειας που βασίζονται σε hardware-assisted μηχανισμούς.

Παρά την εικόνα αυξημένης ασφάλειας που συνοδεύει το macOS στη δημόσια αντίληψη, η πλατφόρμα δεν είναι απαλλαγμένη από ευπάθειες. Στατιστικά στοιχεία από βάσεις δεδομένων ευπαθειών δείχνουν ότι προϊόντα με μεγάλο μερίδιο αγοράς και υψηλή πολυπλοκότητα εμφανίζουν σημαντικό αριθμό καταγεγραμμένων αδυναμιών, ανεξαρτήτως φιλοσοφίας σχεδίασης (CVEdetails.com, 2024; SecurityScorecard, 2023). Η συνεχής εμφάνιση ευπαθειών στο macOS αναδεικνύει τη σημασία της επιστημονικής ανάλυσης των μηχανισμών του, όχι μόνο ως προς τη θεωρητική τους σύλληψη αλλά και ως προς την πρακτική τους αποτελεσματικότητα.

Επιπλέον, το macOS αποτελεί ελκυστικό στόχο για ερευνητές ασφάλειας λόγω της αυξανόμενης χρήσης του σε επαγγελματικά περιβάλλοντα και της διείσδυσής του σε τομείς υψηλής αξίας, όπως η ανάπτυξη λογισμικού και η διαχείριση ευαίσθητων δεδομένων. Τα δεδομένα χρήσης λειτουργικών συστημάτων καταδεικνύουν ότι, αν και το macOS δεν κατέχει την πλειοψηφία της αγοράς desktop, διατηρεί σταθερό και υπολογίσιμο μερίδιο, γεγονός που ενισχύει το κίνητρο για στοχευμένες επιθέσεις (StatCounter Global Stats, 2023a). Η συνύπαρξη αυξημένης ασφάλειας και πραγματικών επιθέσεων καθιστά το macOS ένα ιδανικό πεδίο μελέτης για την αξιολόγηση της αποτελεσματικότητας σύγχρονων αμυντικών τεχνικών.

1.5 Σκοπός, ερευνητικά ερωτήματα και συμβολή της εργασίας

Σκοπός της παρούσας εργασίας είναι η συστηματική και βιβλιογραφικά τεκμηριωμένη ανάλυση της αρχιτεκτονικής ασφάλειας του macOS ως ενιαίου τεχνολογικού οικοσυστήματος. Η προσέγγιση της μελέτης δεν περιορίζεται στον έλεγχο πρόσβασης ή στη διαχείριση μνήμης, αλλά επεκτείνεται στις διακριτές υπηρεσίες εμπιστευτικότητας δεδομένων, ακεραιότητας συστήματος και ασφάλειας εφαρμογών. Με άλλα λόγια, το macOS εξετάζεται ως πλατφόρμα στην οποία υλικό, πυρήνας, σύστημα αρχείων, υπηρεσίες κρυπτογράφησης, πολιτικές εκτέλεσης κώδικα και μηχανισμοί ιδιωτικότητας συνεργάζονται στο πλαίσιο μιας πολυεπίπεδης στρατηγικής άμυνας σε βάθος (Apple, 2026d; Apple, 2026c).

Το πρώτο ερευνητικό ερώτημα αφορά το κατά πόσο η αρχιτεκτονική του macOS πετυχαίνει να μετασχηματίσει κλασικές αρχές ασφάλειας λειτουργικών συστημάτων, όπως το least privilege, η απομόνωση διεργασιών και η διαβάθμιση προνομίων, σε πρακτικά αποτελεσματικούς μηχανισμούς προστασίας. Το δεύτερο ερευνητικό ερώτημα εστιάζει στην εμπιστευτικότητα: με ποιον τρόπο το APFS, το FileVault, οι κλάσεις Data Protection, η Keychain και το Secure Enclave συγκροτούν μια συνεκτική αλυσίδα κρυπτογράφησης και διαχείρισης κλειδιών για την προστασία των δεδομένων σε κατάσταση ηρεμίας και κατά τη χρήση τους (Barker, 2020; Apple, 2024b; Apple, 2026a).

Το τρίτο ερευνητικό ερώτημα εξετάζει εάν η ασφάλεια εφαρμογών και η ακεραιότητα συστήματος στο macOS αποτελούν απλώς μεμονωμένους μηχανισμούς ή εάν συγκροτούν ένα ώριμο μοντέλο εμπιστοσύνης που εκτείνεται από την υπογραφή κώδικα και τη notarization έως το secure boot, το signed system volume, το SIP και το TCC. Η συμβολή της εργασίας έγκειται ακριβώς σε αυτήν τη συνθετική χαρτογράφηση: η θεωρία του Κεφαλαίου 2 συνδέεται άμεσα με τις υλοποιήσεις των Κεφαλαίων 4 έως 7, ώστε να προκύπτει μια ολοκληρωμένη και κριτική εικόνα της θέσης του macOS στο σύγχρονο τοπίο ασφάλειας desktop λειτουργικών συστημάτων (Apple, 2024f; Apple, 2024g; Microsoft, 2025a; Ubuntu, 2025a).

1.6 Δομή της εργασίας

Η εργασία οργανώνεται σε οκτώ κεφάλαια με λογική πορεία από τη θεωρία προς την τεχνολογική υλοποίηση και, τελικά, προς την κριτική αποτίμηση. Το πρώτο κεφάλαιο παρουσιάζει το πρόβλημα, τη σημασία της ασφάλειας λειτουργικών συστημάτων και το ειδικό ενδιαφέρον που παρουσιάζει το macOS ως καθετοποιημένο οικοσύστημα. Το τρίτο κεφάλαιο εστιάζει στη θεμελιώδη αρχιτεκτονική του συστήματος, δηλαδή στο Darwin, στον πυρήνα XNU, στη διαχείριση μνήμης, στην απομόνωση διεργασιών και στο μοντέλο χρηστών και προνομίων.

Το δεύτερο κεφάλαιο λειτουργεί ως θεωρητικός κόμβος της μελέτης. Πέρα από την τριάδα CIA, το μοντέλο AAA και τα μοντέλα ελέγχου πρόσβασης, ενσωματώνει τρεις επιπλέον υπηρεσίες ασφάλειας που είναι κρίσιμες για τη συνέχεια της εργασίας: την εμπιστευτικότητα δεδομένων μέσω κρυπτογράφησης και διαχείρισης κλειδιών, την ακεραιότητα δεδομένων και συστήματος μέσω μηχανισμών επαλήθευσης και trusted boot, και την ασφάλεια εφαρμογών μέσω αλυσίδων εμπιστοσύνης, sandboxing και runtime περιορισμών. Με τον τρόπο αυτό, τα θεωρητικά σχήματα δεν παραμένουν αφηρημένα, αλλά προετοιμάζουν άμεσα την ανάλυση των επόμενων κεφαλαίων.

Τα Κεφάλαια 4, 5 και 6 αποτελούν τον πυρήνα της ειδικής ανάλυσης. Στο Κεφάλαιο 4 εξετάζεται η εμπιστευτικότητα δεδομένων στο macOS, με αναφορά στο APFS, στις κλάσεις προστασίας, στο FileVault, στο Secure Enclave και στη διαχείριση κλειδιών. Στο Κεφάλαιο 5 αναλύεται το μοντέλο ασφάλειας εφαρμογών μέσω code signing, Gatekeeper, notarization, sandboxing και hardened runtime. Στο Κεφάλαιο 6 μελετώνται οι μηχανισμοί ακεραιότητας και προστασίας πόρων, όπως το secure boot, το signed system volume, το SIP, οι ενημερώσεις λογισμικού και το TCC. Το Κεφάλαιο 7 συγκρίνει το macOS με Windows και Linux, ενώ το Κεφάλαιο 8 συνοψίζει τα ευρήματα, απαντά στα ερευνητικά ερωτήματα και προτείνει κατευθύνσεις για μελλοντική έρευνα.

2.Θεωρητικό Πλαίσιο Ασφάλειας Πληροφοριακών Συστημάτων

2.1 Η τριάδα CIA (Confidentiality, Integrity, Availability)

Η τριάδα CIA (Confidentiality, Integrity, Availability) αποτελεί το θεμελιώδες εννοιολογικό πλαίσιο πάνω στο οποίο στηρίζεται η ασφάλεια πληροφοριακών συστημάτων και, κατ' επέκταση, η ασφάλεια των λειτουργικών συστημάτων. Παρά την απλότητά της, η τριάδα αυτή παραμένει εξαιρετικά επίκαιρη, καθώς συμπυκνώνει τους βασικούς στόχους που οφείλει να ικανοποιεί κάθε μηχανισμός προστασίας ανεξαρτήτως τεχνολογικής πλατφόρμας. Οι τρεις συνιστώσες της δεν λειτουργούν απομονωμένα, αλλά βρίσκονται σε συνεχή αλληλεπίδραση, δημιουργώντας συχνά σχέσεις ισορροπίας και συγκρούσεων που επηρεάζουν τον συνολικό σχεδιασμό της ασφάλειας.

Η έννοια της εμπιστευτικότητας (confidentiality) αναφέρεται στη διασφάλιση ότι η πρόσβαση στις πληροφορίες επιτρέπεται μόνο σε εξουσιοδοτημένα υποκείμενα. Στο επίπεδο των λειτουργικών συστημάτων, η εμπιστευτικότητα υλοποιείται μέσω μηχανισμών ελέγχου πρόσβασης, απομόνωσης διεργασιών και προστασίας μνήμης. Η διαχείριση των δικαιωμάτων χρηστών, η επιβολή ορίων μεταξύ διεργασιών και η αποτροπή μη εξουσιοδοτημένης ανάγνωσης δεδομένων αποτελούν βασικά εργαλεία προς αυτή την κατεύθυνση. Η θεωρητική θεμελίωση της εμπιστευτικότητας συνδέεται άμεσα με την έννοια της πληροφορίας και της αβεβαιότητας, όπως αυτή διατυπώθηκε στη θεωρία πληροφορίας, όπου η αποκάλυψη πληροφορίας ισοδυναμεί με μείωση της εντροπίας από την πλευρά του μη εξουσιοδοτημένου παρατηρητή (Shannon, 1948).

Στη σύγχρονη πραγματικότητα, η εμπιστευτικότητα απειλείται κυρίως μέσω επιθέσεων που εκμεταλλεύονται σφάλματα μνήμης, πλευρικά κανάλια και αδυναμίες τυχαιοποίησης. Έρευνες έχουν δείξει ότι ανεπαρκής τυχαιοποίηση του χώρου διευθύνσεων ή χαμηλή εντροπία σε κρίσιμες δομές μνήμης διευκολύνουν την ανάκτηση ευαίσθητων πληροφοριών, ακόμη και χωρίς άμεση παραβίαση των μηχανισμών ελέγχου πρόσβασης (Herlands et al., 2014). Συνεπώς, η εμπιστευτικότητα δεν εξαρτάται μόνο από την πολιτική δικαιωμάτων, αλλά και από την ποιότητα των μηχανισμών τυχαιοποίησης και απομόνωσης που εφαρμόζονται στο λειτουργικό σύστημα.

Η ακεραιότητα (integrity) αφορά τη διασφάλιση ότι τα δεδομένα και οι υπολογιστικές διεργασίες δεν τροποποιούνται με μη εξουσιοδοτημένο ή μη ανιχνεύσιμο τρόπο. Σε επίπεδο λειτουργικού συστήματος, η ακεραιότητα σχετίζεται άμεσα με την προστασία της μνήμης, την επαλήθευση της αυθεντικότητας του κώδικα και τη διατήρηση της συνέπειας των κρίσιμων δομών δεδομένων. Η παραβίαση της ακεραιότητας μπορεί να οδηγήσει σε αλλοίωση δεδομένων, εισαγωγή κακόβουλου κώδικα ή υπονόμευση της ορθής λειτουργίας του συστήματος.

Οι σύγχρονες επιθέσεις συχνά στοχεύουν στην υπονόμευση της ακεραιότητας μέσω εκμετάλλευσης ευπαθειών που επιτρέπουν αυθαίρετη εγγραφή στη μνήμη. Τεχνικές όπως η παράκαμψη της ASLR και η εκμετάλλευση αδυναμιών στην τυχαιοποίηση διευθύνσεων έχουν αποδειχθεί αποτελεσματικές σε πολλές περιπτώσεις, ιδιαίτερα όταν η πραγματική εντροπία του συστήματος είναι χαμηλότερη από τη θεωρητικά αναμενόμενη (Marco-Gisbert & Ripoll, 2019). Η ποσοτική αποτίμηση της ακεραιότητας, επομένως, δεν μπορεί να περιορίζεται σε θεωρητικά μοντέλα, αλλά απαιτεί εμπειρική αξιολόγηση της ανθεκτικότητας των μηχανισμών προστασίας.

Η διαθεσιμότητα (availability) αναφέρεται στην ικανότητα του συστήματος να παρέχει τις υπηρεσίες του στους εξουσιοδοτημένους χρήστες όταν αυτές απαιτούνται. Σε αντίθεση με την εμπιστευτικότητα και την ακεραιότητα, οι οποίες εστιάζουν κυρίως στην αποτροπή μη εξουσιοδοτημένων ενεργειών, η διαθεσιμότητα σχετίζεται με την ανθεκτικότητα του συστήματος απέναντι σε αποτυχίες, σφάλματα και επιθέσεις άρνησης υπηρεσίας. Στο πλαίσιο των λειτουργικών συστημάτων, η διαθεσιμότητα επηρεάζεται από τον σχεδιασμό του υποσυστήματος μνήμης, τη διαχείριση πόρων και την απομόνωση σφαλμάτων.

Η επιδίωξη υψηλής διαθεσιμότητας συχνά έρχεται σε σύγκρουση με αυστηρά μέτρα ασφάλειας. Για παράδειγμα, μηχανισμοί που περιορίζουν επιθετικά τη συμπεριφορά διεργασιών για λόγους ασφάλειας μπορεί να επηρεάσουν την απόδοση ή να οδηγήσουν σε ανεπιθύμητους τερματισμούς εφαρμογών. Η ισορροπία μεταξύ ασφάλειας και διαθεσιμότητας αποτελεί διαχρονικό πρόβλημα στον σχεδιασμό λειτουργικών συστημάτων και αντικατοπτρίζεται στις επιλογές που γίνονται σε επίπεδο αρχιτεκτονικής μνήμης και διαχείρισης διεργασιών (Wilcox, 2021; Torvalds, 2021a).

Σημαντικό στοιχείο της σύγχρονης προσέγγισης στην τριάδα CIA είναι η αναγνώριση ότι οι τρεις συνιστώσες δεν μπορούν να μεγιστοποιηθούν ταυτόχρονα χωρίς συμβιβασμούς. Η ενίσχυση της εμπιστευτικότητας και της ακεραιότητας μέσω πολύπλοκων μηχανισμών τυχαιοποίησης και ελέγχου μπορεί να επιβαρύνει τη διαθεσιμότητα, ενώ η υπερβολική έμφαση στη διαθεσιμότητα μπορεί να οδηγήσει σε χαλαρότερες πολιτικές ασφάλειας. Η κατανόηση αυτών των αλληλεπιδράσεων είναι κρίσιμη για την αξιολόγηση της αποτελεσματικότητας των μηχανισμών ασφάλειας σε πραγματικά λειτουργικά συστήματα.

Η τριάδα CIA, παρά τη μακρόχρονη ιστορία της, εξακολουθεί να αποτελεί χρήσιμο αναλυτικό εργαλείο για τη μελέτη της ασφάλειας των λειτουργικών συστημάτων. Λειτουργεί ως εννοιολογικός άξονας που επιτρέπει τη συστηματική αξιολόγηση διαφορετικών μηχανισμών προστασίας και τη σύγκριση εναλλακτικών σχεδιαστικών επιλογών. Στο πλαίσιο της παρούσας εργασίας, η τριάδα CIA χρησιμοποιείται ως θεωρητικό υπόβαθρο για την ανάλυση των μηχανισμών ασφάλειας του macOS, επιτρέποντας τη σύνδεση αφηρημένων εννοιών ασφάλειας με συγκεκριμένες τεχνικές υλοποιήσεις.

2.2 Authentication, Authorization, Accountability (AAA)

Το μοντέλο Authentication, Authorization, Accountability (AAA) αποτελεί κεντρικό θεωρητικό άξονα της ασφάλειας πληροφοριακών συστημάτων και λειτουργεί συμπληρωματικά προς την τριάδα CIA, εξειδικεύοντας τον τρόπο με τον οποίο οι αρχές της εμπιστευτικότητας, της ακεραιότητας και της διαθεσιμότητας υλοποιούνται σε πρακτικό επίπεδο. Ενώ η CIA περιγράφει τους στόχους της ασφάλειας, το AAA επικεντρώνεται στις διαδικασίες και στους μηχανισμούς μέσω των οποίων επιτυγχάνονται οι στόχοι αυτοί, ιδίως στο πλαίσιο πολυχρηστικών και πολυδιεργασιακών λειτουργικών συστημάτων.

Η επαλήθευση ταυτότητας (authentication) αφορά τη διαδικασία με την οποία ένα σύστημα επιβεβαιώνει την ταυτότητα ενός χρήστη, μιας διεργασίας ή μιας οντότητας πριν της επιτραπεί οποιαδήποτε μορφή πρόσβασης. Σε επίπεδο λειτουργικού συστήματος, η authentication δεν περιορίζεται μόνο στους ανθρώπινους χρήστες, αλλά επεκτείνεται και σε υπηρεσίες, διεργασίες και συστατικά λογισμικού. Η αξιοπιστία της διαδικασίας αυτής αποτελεί προϋπόθεση για κάθε επόμενη απόφαση ασφάλειας, καθώς η αποτυχία στην επαλήθευση ταυτότητας οδηγεί αναπόφευκτα σε κατάρρευση ολόκληρου του μοντέλου προστασίας.

Η authentication συνδέεται άμεσα με τη διαχείριση μνήμης και διεργασιών, καθώς πολλές επιθέσεις στοχεύουν στην πλαστοπροσωπία διεργασιών ή στην απόκτηση αυξημένων προνομίων μέσω εκμετάλλευσης σφαλμάτων χαμηλού επιπέδου. Έρευνες έχουν δείξει ότι αδυναμίες στην τυχαιοποίηση μνήμης και στη διαχείριση εντροπίας μπορούν να διευκολύνουν επιθέσεις που επιτρέπουν την παράκαμψη μηχανισμών επαλήθευσης, ιδίως όταν η ταυτότητα μιας διεργασίας βασίζεται σε προβλέψιμες δομές δεδομένων (Herlands et al., 2014; Acharya et al., 2017). Συνεπώς, η authentication δεν αποτελεί απομονωμένη διαδικασία, αλλά εξαρτάται από την ορθότητα και την ανθεκτικότητα του συνολικού αρχιτεκτονικού σχεδιασμού.

Η εξουσιοδότηση (authorization) αφορά τον καθορισμό και την επιβολή των ενεργειών που επιτρέπεται να εκτελέσει μια αυθεντικοποιημένη οντότητα. Αφού ολοκληρωθεί επιτυχώς η authentication, το λειτουργικό σύστημα καλείται να αποφασίσει σε ποιους πόρους μπορεί να έχει πρόσβαση ο χρήστης ή η διεργασία και με ποιον τρόπο. Οι αποφάσεις αυτές βασίζονται σε πολιτικές ελέγχου πρόσβασης, οι οποίες υλοποιούνται μέσω μηχανισμών όπως δικαιώματα αρχείων, επίπεδα προνομίων και περιορισμοί μνήμης.

Η authorization αποτελεί κρίσιμο σημείο άμυνας έναντι επιθέσεων κλιμάκωσης προνομίων, οι οποίες επιδιώκουν τη μετατροπή περιορισμένων δικαιωμάτων σε πλήρη έλεγχο του συστήματος. Πλήθος καταγεγραμμένων ευπαθειών δείχνει ότι ακόμη και μικρά σφάλματα στην εφαρμογή των πολιτικών εξουσιοδότησης μπορούν να οδηγήσουν σε σοβαρές παραβιάσεις ασφάλειας (Qualys Security Advisory, 2023; CVEdetails.com, 2024). Η αποτελεσματικότητα της authorization εξαρτάται, επομένως, από την αυστηρότητα των πολιτικών και από τη δυσκολία παράκαμψής τους μέσω τεχνικών εκμετάλλευσης μνήμης.

Ιδιαίτερη σημασία έχει το γεγονός ότι η authorization δεν περιορίζεται σε στατικές αποφάσεις, αλλά συχνά εξαρτάται από το εκτελεστικό περιβάλλον και τη χρονική στιγμή. Η δυναμική φύση των σύγχρονων λειτουργικών συστημάτων καθιστά αναγκαία την επαναξιολόγηση των δικαιωμάτων κατά τη διάρκεια εκτέλεσης, γεγονός που αυξάνει την πολυπλοκότητα και δημιουργεί νέα πιθανά σημεία αποτυχίας. Μελέτες που εξετάζουν την αποτελεσματικότητα των μηχανισμών ASLR και απομόνωσης διεργασιών αναδεικνύουν ότι η εξουσιοδότηση μπορεί να υπονομευθεί έμμεσα, όταν ένας επιτιθέμενος αποκτήσει τη δυνατότητα αυθαίρετης εκτέλεσης κώδικα εντός μιας διεργασίας με αυξημένα δικαιώματα (Marco-Gisbert & Ripoll, 2019; Díaz et al., 2021).

Η λογοδοσία (accountability) συμπληρώνει το μοντέλο AAA, διασφαλίζοντας ότι κάθε ενέργεια εντός του συστήματος μπορεί να αποδοθεί σε μια συγκεκριμένη οντότητα. Η accountability επιτυγχάνεται μέσω μηχανισμών καταγραφής, παρακολούθησης και ανίχνευσης γεγονότων, οι οποίοι επιτρέπουν την αναδρομική ανάλυση συμβάντων ασφάλειας. Σε αντίθεση με την authentication και την authorization, που στοχεύουν πρωτίστως στην πρόληψη, η accountability επικεντρώνεται στον εντοπισμό και στην απόδοση ευθύνης μετά από ένα συμβάν.

Η σημασία της accountability έχει αυξηθεί σημαντικά τα τελευταία χρόνια, καθώς η πλήρης αποτροπή επιθέσεων θεωρείται πλέον ανέφικτη. Σε ένα περιβάλλον συνεχώς εξελισσόμενων απειλών, η δυνατότητα ανίχνευσης, ανάλυσης και τεκμηρίωσης παραβιάσεων αποτελεί βασικό στοιχείο της συνολικής στρατηγικής ασφάλειας. Ωστόσο, η αποτελεσματικότητα της accountability εξαρτάται άμεσα από την ακεραιότητα των μηχανισμών καταγραφής και από τη δυσκολία παραποίησής τους, ιδιαίτερα σε περιπτώσεις όπου ο επιτιθέμενος αποκτά αυξημένα προνόμια.

Έρευνες που εξετάζουν πλευρικά κανάλια και χαμηλού επιπέδου επιθέσεις σε σύγχρονες αρχιτεκτονικές έχουν δείξει ότι ακόμη και μηχανισμοί παρακολούθησης μπορούν να παρακαμφθούν ή να χειραγωγηθούν, εάν η υποκείμενη αρχιτεκτονική παρουσιάζει αδυναμίες (Yu et al., 2023). Το γεγονός αυτό υπογραμμίζει ότι η accountability δεν μπορεί να θεωρηθεί ανεξάρτητη από τη συνολική ασφάλεια του συστήματος, αλλά απαιτεί ισχυρή υποστήριξη από μηχανισμούς προστασίας μνήμης και απομόνωσης.

Το μοντέλο AAA προσφέρει ένα λειτουργικό και αναλυτικό πλαίσιο για την κατανόηση της ασφάλειας των λειτουργικών συστημάτων, γεφυρώνοντας το χάσμα μεταξύ αφηρημένων αρχών και πρακτικών υλοποιήσεων. Η επιτυχής εφαρμογή του προϋποθέτει την ταυτόχρονη ενίσχυση και των τριών συνιστωσών, καθώς η αποδυνάμωση οποιασδήποτε από αυτές μπορεί να οδηγήσει σε συστημική αποτυχία. Στο πλαίσιο της παρούσας εργασίας, το μοντέλο AAA λειτουργεί ως βασικό εργαλείο ανάλυσης για την αξιολόγηση των μηχανισμών ασφάλειας του macOS, επιτρέποντας τη συστηματική εξέταση του τρόπου με τον οποίο η πλατφόρμα υλοποιεί την επαλήθευση ταυτότητας, την εξουσιοδότηση και τη λογοδοσία σε ένα σύγχρονο desktop περιβάλλον.

2.3 Μοντέλα ελέγχου πρόσβασης

Τα μοντέλα ελέγχου πρόσβασης αποτελούν τον βασικό μηχανισμό μέσω του οποίου οι αρχές της εμπιστευτικότητας, της ακεραιότητας και της λογοδοσίας μεταφράζονται σε συγκεκριμένες πολιτικές και τεχνικές υλοποιήσεις στα λειτουργικά συστήματα. Ο έλεγχος πρόσβασης ρυθμίζει ποιος μπορεί να προσπελάσει ποιον πόρο, υπό ποιες συνθήκες και με ποιο επίπεδο δικαιωμάτων, λειτουργώντας ως το κύριο φίλτρο μεταξύ νόμιμης χρήσης και κακόβουλης εκμετάλλευσης. Στα σύγχρονα desktop λειτουργικά συστήματα, τα μοντέλα ελέγχου πρόσβασης δεν υφίστανται ως απομονωμένες θεωρητικές κατασκευές, αλλά ενσωματώνονται σε πολύπλοκες αρχιτεκτονικές που αλληλεπιδρούν με τη διαχείριση μνήμης, την απομόνωση διεργασιών και τους μηχανισμούς ασφαλείας χαμηλού επιπέδου.

Η επιλογή και ο συνδυασμός μοντέλων ελέγχου πρόσβασης επηρεάζει άμεσα την ανθεκτικότητα του συστήματος απέναντι σε επιθέσεις κλιμάκωσης προνομίων. Πλήθος καταγεγραμμένων ευπαθειών δείχνει ότι σφάλματα ή ασάφειες στην εφαρμογή πολιτικών πρόσβασης μπορούν να οδηγήσουν σε σοβαρές παραβιάσεις, ακόμη και όταν οι μηχανισμοί authentication λειτουργούν ορθά (Qualys Security Advisory, 2023; CVEdetails.com, 2024). Η κατανόηση των βασικών μοντέλων Discretionary Access Control, Mandatory Access Control και Role-Based Access Control είναι επομένως κρίσιμη για την αξιολόγηση της ασφάλειας ενός λειτουργικού συστήματος.

2.3.1 Discretionary Access Control (DAC)

Το Discretionary Access Control αποτελεί το παραδοσιακότερο και πιο διαδεδομένο μοντέλο ελέγχου πρόσβασης στα Unix-like λειτουργικά συστήματα. Στο DAC, η απόφαση για την πρόσβαση σε έναν πόρο βασίζεται στην ταυτότητα του χρήστη και στην ιδιοκτησία του αντικειμένου, επιτρέποντας στον ιδιοκτήτη ενός πόρου να εκχωρεί ή να ανακαλεί δικαιώματα κατά βούληση. Το μοντέλο αυτό προσφέρει ευελιξία και απλότητα, καθιστώντας το ιδιαίτερα ελκυστικό για γενικής χρήσης περιβάλλοντα.

Ωστόσο, η διακριτική φύση του DAC εισάγει εγγενείς αδυναμίες ασφάλειας. Εφόσον οι χρήστες έχουν τη δυνατότητα να μεταβιβάζουν δικαιώματα, ένα κακόβουλο λογισμικό που εκτελείται με τα δικαιώματα ενός χρήστη μπορεί να αποκτήσει πρόσβαση σε πόρους πέραν του

αρχικού σκοπού του. Το πρόβλημα αυτό εντείνεται σε περιπτώσεις όπου μια διεργασία αποκτά αυξημένα δικαιώματα μέσω εκμετάλλευσης σφαλμάτων μνήμης ή ανεπαρκούς τυχαιοποίησης διευθύνσεων (Marco-Gisbert & Ripoll, 2019).

Έρευνες που εξετάζουν την αποτελεσματικότητα των μηχανισμών ASLR και προστασίας μνήμης έχουν δείξει ότι η παράκαμψη αυτών των μηχανισμών μπορεί να οδηγήσει άμεσα σε καταστρατήγηση του DAC, καθώς η αυθαίρετη εκτέλεση κώδικα επιτρέπει στον επιτιθέμενο να ενεργεί με τα πλήρη δικαιώματα της διεργασίας-στόχου (Herlands et al., 2014). Συνεπώς, το DAC θεωρείται επαρκές μόνο όταν συνδυάζεται με πρόσθετα μέτρα απομόνωσης και περιορισμού προνομίων.

2.3.2 Mandatory Access Control (MAC)

Το Mandatory Access Control αναπτύχθηκε ως απάντηση στις αδυναμίες του DAC, εισάγοντας ένα αυστηρότερο και κεντρικά ελεγχόμενο μοντέλο ελέγχου πρόσβασης. Στο MAC, οι αποφάσεις πρόσβασης δεν εξαρτώνται από τη διακριτική ευχέρεια των χρηστών, αλλά από προκαθορισμένες πολιτικές που επιβάλλονται από το σύστημα. Οι πόροι και τα υποκείμενα χαρακτηρίζονται από ετικέτες ασφάλειας, και η πρόσβαση επιτρέπεται ή απορρίπτεται βάσει αυτών των ετικετών.

Το πλεονέκτημα του MAC έγκειται στη δυνατότητά του να περιορίζει δραστικά την επιφάνεια επίθεσης, ακόμη και σε περιπτώσεις όπου μια διεργασία έχει παραβιαστεί. Ακόμη και αν ένας επιτιθέμενος αποκτήσει έλεγχο μιας διεργασίας, οι υποχρεωτικές πολιτικές πρόσβασης μπορούν να αποτρέψουν την περαιτέρω εξάπλωση της επίθεσης. Αυτό το χαρακτηριστικό καθιστά το MAC ιδιαίτερα αποτελεσματικό απέναντι σε επιθέσεις κλιμάκωσης προνομίων, οι οποίες αποτελούν μία από τις πιο κρίσιμες κατηγορίες απειλών στα σύγχρονα λειτουργικά συστήματα (Qualys Security Advisory, 2023).

Ωστόσο, η αυστηρότητα του MAC συνοδεύεται από αυξημένη πολυπλοκότητα στη διαχείριση και από πιθανές επιπτώσεις στη χρηστικότητα και στη διαθεσιμότητα του συστήματος. Η ανάγκη για λεπτομερή ορισμό πολιτικών μπορεί να οδηγήσει σε σφάλματα διαμόρφωσης, τα οποία, με τη σειρά τους, δημιουργούν νέα σημεία αποτυχίας. Επιπλέον, η αποτελεσματικότητα του MAC εξαρτάται από τη σωστή υλοποίησή του σε επίπεδο πυρήνα και από την ανθεκτικότητα των υποκείμενων μηχανισμών μνήμης και απομόνωσης (Torvalds, 2021a; Wilcox, 2021).

2.3.3 Role-Based Access Control (RBAC)

Το Role-Based Access Control εισάγει μια ενδιάμεση προσέγγιση μεταξύ της ευελιξίας του DAC και της αυστηρότητας του MAC, βασίζοντας τις αποφάσεις πρόσβασης σε ρόλους αντί για μεμονωμένους χρήστες. Στο RBAC, οι χρήστες αντιστοιχίζονται σε ρόλους που ενσωματώνουν συγκεκριμένα σύνολα δικαιωμάτων, επιτρέποντας πιο συστηματική και διαχειρίσιμη πολιτική ελέγχου πρόσβασης.

Η βασική συνεισφορά του RBAC έγκειται στη μείωση της πολυπλοκότητας και στη βελτίωση της λογοδοσίας. Δεδομένου ότι οι ενέργειες εκτελούνται στο πλαίσιο ρόλων, καθίσταται ευκολότερη η παρακολούθηση και η απόδοση ευθύνης σε περίπτωση παραβίασης. Παράλληλα, το RBAC διευκολύνει την εφαρμογή της αρχής του ελάχιστου προνομίου, καθώς οι ρόλοι μπορούν να σχεδιαστούν ώστε να περιλαμβάνουν μόνο τα απολύτως απαραίτητα δικαιώματα.

Παρά τα πλεονεκτήματά του, το RBAC δεν αποτελεί πανάκεια απέναντι σε επιθέσεις χαμηλού επιπέδου. Όπως και στα άλλα μοντέλα, η παράκαμψη των μηχανισμών προστασίας μνήμης μπορεί να οδηγήσει σε αυθαίρετη εκτέλεση κώδικα εντός ενός ρόλου με αυξημένα δικαιώματα, ακυρώνοντας στην πράξη τις θεωρητικές εγγυήσεις του μοντέλου (Marco-Gisbert & Ripoll, 2019; Díaz et al., 2021). Η αποτελεσματικότητα του RBAC εξαρτάται, συνεπώς, από τον συνδυασμό του με ισχυρούς μηχανισμούς απομόνωσης και τυχαιοποίησης.

Τα μοντέλα DAC, MAC και RBAC δεν πρέπει να αντιμετωπίζονται ως αμοιβαία αποκλειόμενες επιλογές, αλλά ως συμπληρωματικά εργαλεία που μπορούν να συνυπάρχουν σε ένα σύγχρονο λειτουργικό σύστημα. Η πρακτική ασφάλεια προκύπτει από τον συνδυασμό τους με μηχανισμούς προστασίας μνήμης, τυχαιοποίησης και αυστηρής επιβολής πολιτικών, στοιχείο που αναδεικνύεται επανειλημμένα στη σύγχρονη βιβλιογραφία ασφάλειας λειτουργικών συστημάτων.

2.4 Εμπιστευτικότητα δεδομένων, κρυπτογράφηση και διαχείριση κλειδιών

Η εμπιστευτικότητα δεδομένων αποτελεί διακριτή υπηρεσία ασφάλειας και δεν ταυτίζεται με τον έλεγχο πρόσβασης. Ο έλεγχος πρόσβασης ορίζει ποιος επιτρέπεται να ζητήσει έναν πόρο, ενώ η εμπιστευτικότητα εξετάζει εάν τα δεδομένα παραμένουν ακατανόητα ή μη ανακτήσιμα όταν ο δίσκος αφαιρεθεί, όταν ένα αντίγραφο αποθηκευτικού μέσου υποκλαπεί ή όταν ένας επιτιθέμενος βρεθεί εκτός του αναμενόμενου μονοπατιού χρήσης του λειτουργικού συστήματος. Για τον λόγο αυτό, στα σύγχρονα λειτουργικά συστήματα η εμπιστευτικότητα απαιτεί συνδυασμό πολιτικών, κρυπτογραφικών μηχανισμών και ασφαλούς αποθήκευσης κλειδιών, ώστε να μην αρκεί η φυσική πρόσβαση ή μια απλή παράκαμψη του γραφικού περιβάλλοντος για την ανάγνωση ευαίσθητων πληροφοριών (Shannon, 1948; Barker, 2020).

Η υλοποίηση της εμπιστευτικότητας στηρίζεται κυρίως σε τρεις τεχνικούς άξονες: κρυπτογράφηση δεδομένων, διαχείριση κλειδιών και αξιόπιστη σύνδεση των κλειδιών με κάποια ρίζα εμπιστοσύνης. Η κρυπτογράφηση χωρίς ορθή ιεραρχία κλειδιών παραμένει ατελής, ενώ η διαχείριση κλειδιών χωρίς ασφαλές υλικό ή ασφαλή αποθήκευση είναι ευάλωτη σε εξαγωγή μυστικού υλικού. Γι' αυτό η θεωρία της κρυπτογραφικής διαχείρισης δίνει έμφαση στον κύκλο ζωής του κλειδιού - δημιουργία, αποθήκευση, χρήση, περιστροφή και ανάκληση - και όχι μόνο στον αλγόριθμο κρυπτογράφησης που χρησιμοποιείται (Barker, 2020).

Στο macOS η υπηρεσία αυτή αποκτά ιδιαίτερη σημασία, επειδή το λειτουργικό σύστημα δεν αρκείται σε ένα μόνο επίπεδο προστασίας. Η εμπιστευτικότητα διαχέεται στο σύστημα αρχείων, στις κλάσεις Data Protection, στο FileVault, στην Keychain και στο Secure Enclave, ενώ η πρόσβαση στα κλειδιά επηρεάζεται από τη συνθηματική κατάσταση του χρήστη, από την πολιτική εκκίνησης και από το hardware root of trust. Για τον λόγο αυτό, το Κεφάλαιο 4 αναλύει την εμπιστευτικότητα όχι ως μεμονωμένο χαρακτηριστικό, αλλά ως αλληλουχία μηχανισμών που καλύπτουν διαφορετικά σενάρια απειλής - από την απώλεια συσκευής έως την κατάχρηση διαπιστευτηρίων εφαρμογής (Apple, 2024b; Apple, 2024d; Apple, 2025a; Apple, 2026a).

2.5 Ακεραιότητα δεδομένων και συστήματος

Η ακεραιότητα δεδομένων και συστήματος αναφέρεται στη διασφάλιση ότι κώδικας, ρυθμίσεις και πληροφορίες δεν μεταβάλλονται με μη εξουσιοδοτημένο ή μη ανιχνεύσιμο τρόπο. Ενώ η εμπιστευτικότητα εστιάζει στην απόκρυψη της πληροφορίας, η ακεραιότητα εστιάζει

στην αυθεντικότητα και στην αμεταβλητότητά της. Σε επίπεδο λειτουργικού συστήματος αυτό αφορά τόσο αρχεία και βιβλιοθήκες όσο και την ίδια την αλυσίδα εκκίνησης, τις δομές του πυρήνα, τις ενημερώσεις λογισμικού και τα δικαιώματα που επιτρέπουν ή αποτρέπουν κρίσιμες τροποποιήσεις.

Η πρακτική υλοποίηση της ακεραιότητας στηρίζεται σε hashes, ψηφιακές υπογραφές, verified boot, read-only ή sealed περιοχές και μηχανισμούς που περιορίζουν ακόμη και τον χρήστη root όταν επιχειρείται μεταβολή κρίσιμου κώδικα. Η υπηρεσία αυτή συνδέεται στενά με τη διαδικασία εκκίνησης: εάν το σύστημα ξεκινήσει από μη αξιόπιστο firmware ή μη επαληθευμένο kernel, τότε όλες οι μεταγενέστερες πολιτικές προστασίας εφαρμόζονται σε ήδη υπονομευμένη βάση. Για τον ίδιο λόγο, η ακεραιότητα περιλαμβάνει και ασφαλείς ενημερώσεις, αντι-rollback προστασίες και πολιτικές που δυσκολεύουν την επαναφορά παλαιότερων ή παραποιημένων images (Apple, 2021c; Apple, 2022b; Apple, 2026c).

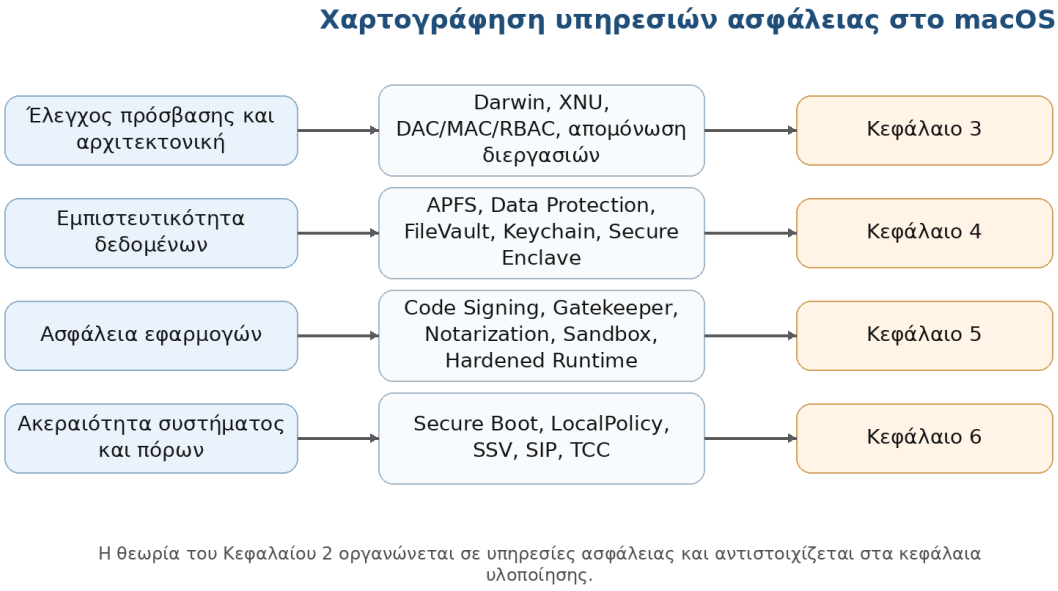
Στο macOS η ακεραιότητα δεν περιορίζεται σε μια υπογραφή αρχείου, αλλά υλοποιείται ως αλυσίδα εμπιστοσύνης από το hardware root of trust μέχρι το signed system volume, το SIP και τους μηχανισμούς μεσολάβησης πρόσβασης σε προστατευμένους πόρους. Αυτό σημαίνει ότι το Κεφάλαιο 6 δεν θα εξετάσει την ακεραιότητα μόνο ως τεχνική επαλήθευση κώδικα, αλλά ως συνολικό μοντέλο που προστατεύει την εκκίνηση, το runtime και την αλληλεπίδραση εφαρμογών με κρίσιμους καταλόγους και δεδομένα χρήστη (Apple, 2021b; Apple, 2022b; Apple, 2026c).

2.6 Ασφάλεια εφαρμογών ως διακριτή υπηρεσία

Η ασφάλεια εφαρμογών συνιστά ξεχωριστή υπηρεσία ασφάλειας, επειδή αντιμετωπίζει ένα διαφορετικό πρόβλημα από εκείνο του ελέγχου πρόσβασης ή της κρυπτογράφησης: το εάν μια εφαρμογή είναι αξιόπιστη να εκτελεστεί, με ποιον τρόπο επιτρέπεται να αλληλεπιδρά με το σύστημα και ποια μέσα διαθέτει για να περιορίσει τις επιπτώσεις μιας πιθανής παραβίασής της. Στα σύγχρονα οικοσυστήματα λογισμικού, η ασφάλεια εφαρμογών δεν αρχίζει τη στιγμή της εκτέλεσης αλλά από τον κύκλο ζωής ανάπτυξης, υπογραφής, διανομής, πρώτης εκκίνησης, runtime επιβολής και ενδεχόμενης ανάκλησης ή απομάκρυνσης ενός πακέτου λογισμικού.

Από θεωρητική σκοπιά, η υπηρεσία αυτή βασίζεται στην αλυσίδα εμπιστοσύνης του κώδικα: ποιος παρήγαγε το λογισμικό, αν το πακέτο παραδόθηκε αναλλοίωτο, αν ελέγχθηκε πριν από τη διανομή του και τι επιτρέπεται να κάνει αφού ξεκινήσει. Στην πράξη, αυτό μεταφράζεται σε code signing, συστήματα reputation και notarization, sandboxing, entitlements, runtime περιορισμούς, revocation και μηχανισμούς ανίχνευσης malware. Όσο μεγαλύτερη είναι η εξάρτηση μιας πλατφόρμας από τρίτο λογισμικό και από αλυσίδες εφοδιασμού, τόσο κρισιμότερη γίνεται η υπηρεσία της ασφάλειας εφαρμογών (Apple, 2022a; Apple, 2024f; Apple, 2024g).

Το macOS παρουσιάζει ιδιαίτερο ενδιαφέρον στη συγκεκριμένη υπηρεσία, επειδή διαθέτει σχετικά ώριμο και πολυεπίπεδο μοντέλο εμπιστοσύνης εφαρμογών, στο οποίο συνυπάρχουν η υπογραφή κώδικα, ο έλεγχος προέλευσης μέσω Gatekeeper, η notarization, το App Sandbox, το Hardened Runtime και η μεταγενέστερη άμυνα μέσω XProtect. Η εικόνα 1 αποτυπώνει αυτήν τη χαρτογράφηση και λειτουργεί ως γέφυρα προς το Κεφάλαιο 5, όπου θα εξεταστεί αναλυτικά πώς η θεωρητική υπηρεσία της ασφάλειας εφαρμογών μεταφράζεται σε συγκεκριμένο μηχανισμό πολιτικής και εκτέλεσης στο macOS (Apple, 2026b; Apple Developer, n.d.-a; Apple Developer, n.d.-b; Apple Developer, n.d.-c).



Εικόνα 1 Θεωρητική χαρτογράφηση υπηρεσιών ασφάλειας και βασικών μηχανισμών του macOS.

2.7 Αρχές Least Privilege και Separation of Duties

Οι αρχές του ελάχιστου προνομίου (Least Privilege) και του διαχωρισμού καθηκόντων (Separation of Duties) αποτελούν θεμελιώδεις σχεδιαστικές αρχές της ασφάλειας πληροφοριακών συστημάτων και λειτουργούν συμπληρωματικά προς τα μοντέλα ελέγχου πρόσβασης. Ενώ τα μοντέλα DAC, MAC και RBAC ορίζουν το πλαίσιο λήψης αποφάσεων πρόσβασης, οι δύο αυτές αρχές καθορίζουν τον τρόπο με τον οποίο τα προνόμια κατανέμονται και χρησιμοποιούνται στην πράξη, με στόχο τη μείωση της επιφάνειας επίθεσης και τον περιορισμό των επιπτώσεων μιας πιθανής παραβίασης.

Η αρχή του ελάχιστου προνομίου υπαγορεύει ότι κάθε χρήστης, διεργασία ή υποσύστημα πρέπει να διαθέτει μόνο τα απολύτως απαραίτητα δικαιώματα για την εκτέλεση της λειτουργίας του και τίποτε περισσότερο. Η εφαρμογή της αρχής αυτής μειώνει δραστικά τις δυνατότητες ενός επιτιθέμενου σε περίπτωση επιτυχούς εκμετάλλευσης μιας ευπάθειας, καθώς τα δικαιώματα που μπορεί να καταχραστεί περιορίζονται εκ των προτέρων. Στο πλαίσιο των λειτουργικών συστημάτων, το Least Privilege συνδέεται άμεσα με τη διαχείριση διεργασιών, την απομόνωση μνήμης και τη διάκριση μεταξύ προνομιούχου και μη προνομιούχου κώδικα.

Η σημασία του ελάχιστου προνομίου αναδεικνύεται ιδιαίτερα από τη μελέτη περιστατικών κλιμάκωσης προνομίων, όπου αρχικά περιορισμένη πρόσβαση μετατρέπεται σε πλήρη έλεγχο του συστήματος. Καταγεγραμμένες ευπάθειες σε βασικά υποσυστήματα, όπως βιβλιοθήκες συστήματος και μηχανισμοί φόρτωσης διεργασιών, δείχνουν ότι ακόμη και μικρές αποκλίσεις από την αρχή του ελάχιστου προνομίου μπορούν να έχουν δυσανάλογα μεγάλες επιπτώσεις (Qualys Security Advisory, 2023). Τα στατιστικά δεδομένα ευπαθειών επιβεβαιώνουν ότι η κλιμάκωση προνομίων παραμένει μία από τις πιο συχνές και επικίνδυνες κατηγορίες επιθέσεων σε σύγχρονα λειτουργικά συστήματα (CVEdetails.com, 2024; SecurityScorecard, 2023).

Η αποτελεσματική εφαρμογή του Least Privilege δεν περιορίζεται σε επίπεδο πολιτικών πρόσβασης, αλλά εξαρτάται και από τη σωστή υλοποίηση μηχανισμών προστασίας μνήμης. Έρευνες που εξετάζουν την πρακτική αποτελεσματικότητα της ASLR και συναφών τεχνικών

δείχνουν ότι, όταν η τυχαιοποίηση είναι ανεπαρκής ή προβλέψιμη, ένας επιτιθέμενος μπορεί να αποκτήσει αυθαίρετη εκτέλεση κώδικα εντός μιας διεργασίας, παρακάμπτοντας στην πράξη τα περιορισμένα προνόμια που της έχουν αποδοθεί (Marco-Gisbert & Ripoll, 2019; Díaz et al., 2021). Το γεγονός αυτό υπογραμμίζει ότι το ελάχιστο προνόμιο πρέπει να συνδυάζεται με ισχυρή απομόνωση και υψηλή εντροπία στη διάταξη κρίσιμων δομών μνήμης (Herlands et al., 2014).

Η αρχή του διαχωρισμού καθηκόντων συμπληρώνει το Least Privilege, επιβάλλοντας την κατανομή κρίσιμων λειτουργιών σε διαφορετικές οντότητες, ώστε καμία μεμονωμένη οντότητα να μην έχει τη δυνατότητα ολοκληρωμένου ελέγχου ενός συστήματος ή μιας κρίσιμης διαδικασίας. Στο επίπεδο των λειτουργικών συστημάτων, ο Separation of Duties υλοποιείται μέσω της διάκρισης ρόλων, της απομόνωσης υπηρεσιών και της κατανομής λειτουργιών μεταξύ πυρήνα, συστημικών υπηρεσιών και εφαρμογών χρήστη.

Ο διαχωρισμός καθηκόντων συμβάλλει στη μείωση του κινδύνου τόσο κακόβουλων ενεργειών όσο και σφαλμάτων, καθώς απαιτείται η σύμπραξη πολλαπλών μηχανισμών ή ρόλων για την ολοκλήρωση ευαίσθητων ενεργειών. Η αρχή αυτή συνδέεται στενά με το RBAC, όπου τα δικαιώματα αποδίδονται σε ρόλους αντί για μεμονωμένους χρήστες, επιτρέποντας πιο συστηματική και ελεγχόμενη διαχείριση προνομίων. Παράλληλα, ενισχύει τη λογοδοσία, καθώς οι ενέργειες μπορούν να αποδοθούν με μεγαλύτερη σαφήνεια σε συγκεκριμένες λειτουργικές οντότητες.

Η πρακτική αξία του διαχωρισμού καθηκόντων γίνεται ιδιαίτερα εμφανής σε περιπτώσεις όπου μια διεργασία ή υπηρεσία παραβιάζεται. Εάν το σύστημα έχει σχεδιαστεί έτσι ώστε κρίσιμες λειτουργίες να είναι κατανεμημένες, η παραβίαση μιας μόνο συνιστώσας δεν αρκεί για την πλήρη υπονόμευση της ασφάλειας. Ωστόσο, όπως και στην περίπτωση του Least Privilege, η αποτελεσματικότητα του Separation of Duties εξαρτάται από την ανθεκτικότητα των υποκείμενων μηχανισμών. Επιθέσεις χαμηλού επιπέδου που εκμεταλλεύονται αδυναμίες στη διαχείριση μνήμης ή πλευρικά κανάλια μπορούν να οδηγήσουν σε παραβίαση των ορίων μεταξύ διακριτών καθηκόντων, υπονομεύοντας τον αρχικό σχεδιασμό (Yu et al., 2023).

Η ιστορική εξέλιξη των λειτουργικών συστημάτων δείχνει ότι οι αρχές του ελάχιστου προνομίου και του διαχωρισμού καθηκόντων δεν υιοθετήθηκαν εξαρχής με συστηματικό τρόπο, αλλά ενισχύθηκαν σταδιακά ως απάντηση στην αύξηση της πολυπλοκότητας και των απειλών. Η σύγχρονη έρευνα στην ασφάλεια μνήμης και τυχαιοποίησης έχει καταδείξει ότι οι αρχές αυτές παραμένουν αναγκαίες αλλά όχι επαρκείς από μόνες τους, απαιτώντας συνεχή επανεξέταση και προσαρμογή στις νέες τεχνικές επίθεσης (Marco-Gisbert & Ripoll, 2019; Herlands et al., 2014).

Οι αρχές Least Privilege και Separation of Duties αποτελούν κρίσιμα δομικά στοιχεία της ασφάλειας πληροφοριακών συστημάτων, καθώς περιορίζουν τόσο τις δυνατότητες εκμετάλλευσης ευπαθειών όσο και τις επιπτώσεις μιας επιτυχούς επίθεσης. Στο πλαίσιο της παρούσας εργασίας, οι αρχές αυτές λειτουργούν ως θεωρητική βάση για την ανάλυση των μηχανισμών ασφάλειας του macOS, επιτρέποντας την αξιολόγηση του βαθμού στον οποίο η πλατφόρμα ενσωματώνει και εφαρμόζει στην πράξη τις θεμελιώδεις αρχές της σύγχρονης ασφάλειας λειτουργικών συστημάτων.

2.8 Defense-in-Depth ως στρατηγική ασφάλειας

Η στρατηγική της άμυνας σε βάθος (Defense-in-Depth) αποτελεί μία από τις πιο ανθεκτικές και διαχρονικά αποτελεσματικές προσεγγίσεις στην ασφάλεια πληροφοριακών συστημάτων. Σε

αντίθεση με μονοεπίπεδες στρατηγικές που βασίζονται στην αποτροπή μέσω ενός και μόνο μηχανισμού, η άμυνα σε βάθος στηρίζεται στη συνδυαστική χρήση πολλαπλών, ετερογενών επιπέδων προστασίας, με στόχο να αποτρέψει, να καθυστερήσει ή να περιορίσει τις επιπτώσεις μιας επιτυχούς επίθεσης. Η βασική της παραδοχή είναι ότι κανένας μεμονωμένος μηχανισμός δεν μπορεί να θεωρηθεί απολύτως ασφαλής και ότι η αποτυχία ενός επιπέδου δεν πρέπει να συνεπάγεται την πλήρη κατάρρευση του συστήματος.

Η θεωρητική θεμελίωση της Defense-in-Depth συνδέεται άμεσα με τις αρχές της τριάδας CIA και του μοντέλου AAA, καθώς και με τις αρχές του ελάχιστου προνομίου και του διαχωρισμού καθηκόντων. Η πολυεπίπεδη προστασία επιδιώκει να διασφαλίσει ότι ακόμη και αν παραβιαστεί η εμπιστευτικότητα ή η ακεραιότητα σε ένα επίπεδο, θα υπάρχουν πρόσθετοι μηχανισμοί που θα αποτρέψουν την περαιτέρω εκμετάλλευση ή θα διατηρήσουν τη διαθεσιμότητα του συστήματος. Η λογική αυτή αντικατοπτρίζει τη μετάβαση από την απόλυτη πρόληψη προς την ανθεκτικότητα και τη διαχείριση κινδύνου, αναγνωρίζοντας ότι η πλήρης αποτροπή όλων των επιθέσεων είναι πρακτικά ανέφικτη.

Στο πλαίσιο των λειτουργικών συστημάτων, η Defense-in-Depth υλοποιείται μέσω της ταυτόχρονης εφαρμογής μηχανισμών σε διαφορετικά επίπεδα της αρχιτεκτονικής, όπως η διαχείριση μνήμης, η απομόνωση διεργασιών, ο έλεγχος πρόσβασης και η επιβολή πολιτικών εκτέλεσης κώδικα. Μηχανισμοί όπως η τυχαιοποίηση της διάταξης του χώρου διευθύνσεων, η προστασία εγγραφής σε κρίσιμες περιοχές μνήμης και ο διαχωρισμός προνομιούχου και μη προνομιούχου κώδικα λειτουργούν συμπληρωματικά, αυξάνοντας το κόστος και την πολυπλοκότητα μιας επίθεσης (Herlands et al., 2014; Marco-Gisbert & Ripoll, 2019).

Η σημασία της άμυνας σε βάθος αναδεικνύεται ιδιαίτερα από μελέτες που εξετάζουν την πρακτική αποτελεσματικότητα μεμονωμένων μηχανισμών ασφάλειας. Έρευνες πάνω στην ASLR έχουν δείξει ότι, αν και η τυχαιοποίηση μνήμης μπορεί να δυσχεράνει σημαντικά την εκμετάλλευση ευπαθειών, σπάνια είναι από μόνη της επαρκής για την πλήρη αποτροπή επιθέσεων, ιδίως όταν η πραγματική εντροπία είναι χαμηλότερη από τη θεωρητικά αναμενόμενη (Acharya et al., 2017; Díaz et al., 2021). Σε τέτοιες περιπτώσεις, η ύπαρξη πρόσθετων επιπέδων, όπως αυστηρότερος έλεγχος προνομίων και απομόνωση διεργασιών, καθίσταται κρίσιμη για τη συνολική ασφάλεια.

Η Defense-in-Depth αποκτά ιδιαίτερη σημασία και σε σχέση με τις επιθέσεις κλιμάκωσης προνομίων. Πολλά καταγεγραμμένα περιστατικά δείχνουν ότι οι επιτιθέμενοι σπάνια επιτυγχάνουν πλήρη έλεγχο του συστήματος με μία μόνο ευπάθεια, αλλά ακολουθούν αλυσιδωτές επιθέσεις που εκμεταλλεύονται διαδοχικά κενά ασφάλειας σε διαφορετικά επίπεδα (Qualys Security Advisory, 2023). Η πολυεπίπεδη άμυνα στοχεύει ακριβώς στη διάσπαση αυτών των αλυσίδων, εισάγοντας εμπόδια που περιορίζουν την κλιμάκωση και καθιστούν την επίθεση λιγότερο αξιόπιστη ή πιο εύκολα ανιχνεύσιμη.

Η αποτελεσματικότητα της Defense-in-Depth εξαρτάται, ωστόσο, όχι μόνο από τον αριθμό των επιπέδων προστασίας, αλλά και από την ετερογένεια και την ανεξαρτησία τους. Εάν πολλαπλοί μηχανισμοί βασίζονται σε κοινές παραδοχές ή κοινές υλοποιήσεις, τότε μια θεμελιώδης αδυναμία μπορεί να τους υπονομεύσει ταυτόχρονα. Μελέτες σε χαμηλού επιπέδου επιθέσεις και πλευρικά κανάλια έχουν δείξει ότι η κοινή εξάρτηση από συγκεκριμένα χαρακτηριστικά της αρχιτεκτονικής μπορεί να οδηγήσει σε παράκαμψη πολλαπλών επιπέδων άμυνας ταυτόχρονα (Yu et al., 2023). Το γεγονός αυτό υπογραμμίζει την ανάγκη για διαφοροποίηση των μηχανισμών και για συνεχή αξιολόγηση των υποκείμενων παραδοχών ασφαλείας.

Σημαντική διάσταση της άμυνας σε βάθος είναι και η δυνατότητα ανίχνευσης και απόκρισης. Ακόμη και όταν μια επίθεση καταφέρει να παρακάμψει αρχικά επίπεδα προστασίας, η ύπαρξη μηχανισμών καταγραφής και παρακολούθησης μπορεί να επιτρέψει την έγκαιρη αναγνώριση και τον περιορισμό της. Η λογοδοσία, ως συνιστώσα του μοντέλου AAA, ενσωματώνεται έτσι στη στρατηγική Defense-in-Depth, μετατρέποντας την ασφάλεια από στατική άμυνα σε δυναμική διαδικασία προσαρμογής.

Η ιστορική εξέλιξη της ασφάλειας λειτουργικών συστημάτων καταδεικνύει ότι η Defense-in-Depth δεν αποτελεί απλώς μια θεωρητική σύσταση, αλλά αναγκαία απάντηση στην αυξανόμενη πολυπλοκότητα και στην επαγγελματικοποίηση των επιθέσεων. Η συσσώρευση ευπαθειών σε δημοφιλείς πλατφόρμες και η συνεχής δημοσίευση νέων τεχνικών εκμετάλλευσης επιβεβαιώνουν ότι η μονοεπίπεδη άμυνα είναι ανεπαρκής σε σύγχρονα περιβάλλοντα (CVEdetails.com, 2024; SecurityScorecard, 2023). Αντιθέτως, η συνδυαστική εφαρμογή πολλαπλών επιπέδων προστασίας αυξάνει σημαντικά την ανθεκτικότητα του συστήματος, ακόμη και υπό συνθήκες μερικής αποτυχίας.

Η Defense-in-Depth αποτελεί θεμελιώδη στρατηγική ασφάλειας που ενσωματώνει και συνθέτει τις βασικές αρχές της ασφάλειας πληροφοριακών συστημάτων. Δεν υπόσχεται απόλυτη ασφάλεια, αλλά επιδιώκει τη συστηματική μείωση του κινδύνου και τον περιορισμό των επιπτώσεων επιθέσεων, καθιστώντας την ιδιαίτερα κατάλληλη για τη μελέτη και αξιολόγηση σύγχρονων λειτουργικών συστημάτων. Στο πλαίσιο της παρούσας εργασίας, η στρατηγική αυτή λειτουργεί ως θεωρητικό υπόβαθρο για την ανάλυση των μηχανισμών ασφάλειας του macOS, επιτρέποντας την αποτίμηση του βαθμού στον οποίο η πλατφόρμα υιοθετεί μια πραγματικά πολυεπίπεδη προσέγγιση άμυνας.

2.9 Θεωρητική τοποθέτηση της ασφάλειας του macOS στο ευρύτερο πλαίσιο OS Security

Η ασφάλεια του macOS μπορεί να τοποθετηθεί θεωρητικά στο ευρύτερο πλαίσιο της ασφάλειας λειτουργικών συστημάτων ως ένα χαρακτηριστικό παράδειγμα σύγχρονης, πολυεπίπεδης αρχιτεκτονικής που επιδιώκει να συνδυάσει κλασικές αρχές ασφάλειας με προηγμένους μηχανισμούς προστασίας χαμηλού επιπέδου. Σε αντίθεση με προσεγγίσεις που βασίζονται κυρίως στην ευελιξία και στην παραμετροποίηση από τον χρήστη ή τον διαχειριστή, το macOS υιοθετεί μια περισσότερο κατευθυνόμενη και κεντρικά ελεγχόμενη φιλοσοφία, η οποία εστιάζει στην πρόληψη και στον περιορισμό της επιφάνειας επίθεσης μέσω αυστηρών σχεδιαστικών επιλογών.

Στο επίπεδο της θεωρίας ασφάλειας, το macOS ενσωματώνει με συστηματικό τρόπο τις αρχές της τριάδας CIA και του μοντέλου AAA, όχι ως αφηρημένες έννοιες, αλλά ως λειτουργικές απαιτήσεις που διατρέχουν ολόκληρη την αρχιτεκτονική του συστήματος. Η εμπιστευτικότητα και η ακεραιότητα διασφαλίζονται μέσω εκτεταμένης απομόνωσης μνήμης και ελέγχου εκτέλεσης κώδικα, ενώ η διαθεσιμότητα αντιμετωπίζεται ως ζήτημα ανθεκτικότητας και ελεγχόμενης αποτυχίας. Η προσέγγιση αυτή ευθυγραμμίζεται με σύγχρονες θεωρητικές αντιλήψεις της ασφάλειας λειτουργικών συστημάτων, σύμφωνα με τις οποίες η πλήρης αποτροπή επιθέσεων θεωρείται ανέφικτη και αντικαθίσταται από τη στρατηγική της μείωσης κινδύνου και του περιορισμού συνεπειών (Herlands et al., 2014).

Σε σύγκριση με άλλα desktop λειτουργικά συστήματα, η θεωρητική τοποθέτηση του macOS διακρίνεται από την έντονη έμφαση στη διαχείριση μνήμης ως βασικό άξονα ασφάλειας. Η

αποτελεσματικότητα μηχανισμών όπως η ASLR εξαρτάται σε μεγάλο βαθμό από την ποιότητα της τυχαιοποίησης και την πραγματική εντροπία που επιτυγχάνεται στην πράξη. Μελέτες που εξετάζουν διαφορετικές υλοποιήσεις ASLR σε σύγχρονα λειτουργικά συστήματα δείχνουν ότι η θεωρητική ισχύς των μηχανισμών αυτών συχνά αποκλίνει από την πρακτική τους ανθεκτικότητα, ιδίως όταν η αρχιτεκτονική ή η υλοποίηση επιβάλλει προβλέψιμους περιορισμούς (Díaz et al., 2021; Marco-Gisbert & Ripoll, 2019). Στο πλαίσιο αυτό, το macOS μπορεί να θεωρηθεί ως ένα σύστημα που επενδύει σημαντικά στην αύξηση της πολυπλοκότητας των επιθέσεων, χωρίς ωστόσο να εξαλείφει πλήρως τις θεωρητικές αδυναμίες των μηχανισμών τυχαιοποίησης.

Η θεωρητική διαφοροποίηση του macOS γίνεται ακόμη πιο εμφανής όταν εξεταστεί σε σχέση με την έννοια της εντροπίας ως μετρικής ασφάλειας. Η σύγχρονη βιβλιογραφία έχει αναδείξει ότι η ασφάλεια πολλών μηχανισμών χαμηλού επιπέδου μπορεί να αποτιμηθεί ποσοτικά μέσω της έννοιας της αποτελεσματικής εντροπίας, η οποία λαμβάνει υπόψη όχι μόνο το θεωρητικό εύρος τυχαιοποίησης αλλά και τους πρακτικούς περιορισμούς υλοποίησης (Acharya et al., 2017; Herlands et al., 2014). Υπό αυτό το πρίσμα, η ασφάλεια του macOS τοποθετείται σε ένα ενδιάμεσο σημείο μεταξύ θεωρητικής επάρκειας και πρακτικής ανθεκτικότητας, καθώς επιδιώκει να μεγιστοποιήσει την εντροπία εντός των ορίων που θέτουν η απόδοση και η συμβατότητα.

Η αυστηρή επιβολή πολιτικών ασφάλειας στο macOS συνδέεται άμεσα με τη θεωρητική αρχή του Defense-in-Depth. Αντί να βασίζεται αποκλειστικά σε έναν μηχανισμό, το σύστημα υιοθετεί πολλαπλά, επικαλυπτόμενα επίπεδα προστασίας, τα οποία καλύπτουν διαφορετικά στάδια της αλυσίδας επίθεσης. Η θεωρητική αξία της προσέγγισης αυτής επιβεβαιώνεται από εμπειρικές μελέτες που δείχνουν ότι οι επιθέσεις χαμηλού επιπέδου σπάνια επιτυγχάνουν πλήρη έλεγχο του συστήματος χωρίς τη διαδοχική παράκαμψη πολλών μηχανισμών (Qualys Security Advisory, 2023). Η άμυνα σε βάθος, επομένως, λειτουργεί ως ενοποιητικός θεωρητικός μηχανισμός που συνδέει την αρχιτεκτονική του macOS με τις σύγχρονες αντιλήψεις της ασφάλειας λειτουργικών συστημάτων.

Παράλληλα, η θεωρητική τοποθέτηση του macOS δεν μπορεί να αποσυνδεθεί από το ευρύτερο οικοσύστημα απειλών και ευπαθειών που χαρακτηρίζει τα σύγχρονα λειτουργικά συστήματα. Στατιστικά δεδομένα από βάσεις καταγραφής ευπαθειών δείχνουν ότι κανένα λειτουργικό σύστημα δεν αποτελεί εξαίρεση από τη συνεχή ανακάλυψη νέων αδυναμιών, ανεξαρτήτως φιλοσοφίας σχεδίασης (CVEdetails.com, 2024; SecurityScorecard, 2023). Το γεγονός αυτό υπογραμμίζει ότι η ασφάλεια του macOS πρέπει να αξιολογείται όχι ως απόλυτη κατάσταση, αλλά ως δυναμική ισορροπία μεταξύ αμυντικών τεχνικών και εξελισσόμενων επιθετικών μεθόδων.

Η ύπαρξη προηγμένων επιθέσεων χαμηλού επιπέδου, όπως πλευρικά κανάλια που εκμεταλλεύονται χαρακτηριστικά της υποκείμενης αρχιτεκτονικής, καταδεικνύει τα όρια ακόμη και των πιο αυστηρών μοντέλων ασφάλειας. Έρευνες σε σύγχρονες πλατφόρμες έχουν δείξει ότι επιθέσεις αυτού του τύπου μπορούν να παρακάμψουν παραδοσιακούς μηχανισμούς απομόνωσης, θέτοντας νέα θεωρητικά ερωτήματα σχετικά με την ασφάλεια σε επίπεδο υλικού και λογισμικού (Yu et al., 2023). Στο πλαίσιο αυτό, το macOS εντάσσεται σε μια ευρύτερη κατηγορία λειτουργικών συστημάτων που καλούνται να αντιμετωπίσουν όχι μόνο λογισμικές, αλλά και αρχιτεκτονικές απειλές.

Η θεωρητική τοποθέτηση της ασφάλειας του macOS στο ευρύτερο πλαίσιο της OS Security αναδεικνύει μια προσέγγιση που δίνει έμφαση στην πρόληψη, στην πολυεπίπεδη άμυνα και

στην αυστηρή επιβολή πολιτικών. Η προσέγγιση αυτή ευθυγραμμίζεται με τις σύγχρονες θεωρητικές τάσεις της ασφάλειας λειτουργικών συστημάτων, ενώ ταυτόχρονα αποκαλύπτει τα εγγενή όρια κάθε αμυντικού μοντέλου. Η κατανόηση αυτής της τοποθέτησης αποτελεί αναγκαίο υπόβαθρο για την ανάλυση που ακολουθεί στα επόμενα κεφάλαια, όπου η θεωρία θα συνδεθεί με τη συγκεκριμένη αρχιτεκτονική και τους μηχανισμούς ασφάλειας του macOS.

Με βάση τα παραπάνω, το macOS μπορεί να ιδωθεί ως πλατφόρμα όπου οι υπηρεσίες ασφάλειας δεν είναι αποσπασματικές, αλλά κατανέμονται σε αλληλοσυμπληρούμενα στρώματα. Η αρχιτεκτονική και ο έλεγχος πρόσβασης αναλύονται στο Κεφάλαιο 3, η εμπιστευτικότητα δεδομένων στο Κεφάλαιο 4, η ασφάλεια εφαρμογών στο Κεφάλαιο 5 και η ακεραιότητα συστήματος και πόρων στο Κεφάλαιο 6. Η διάρθρωση αυτή επιτρέπει μια πιο ακριβή θεωρητική και πρακτική ανάγνωση του macOS ως συστήματος άμυνας σε βάθος και προετοιμάζει το έδαφος για τη συγκριτική και κριτική αποτίμηση του Κεφαλαίου 7 (Apple, 2026d; Apple, 2026c).

3.Αρχιτεκτονική Θεμελίωση του macOS

3.1 Το Darwin OS ως βάση του macOS

Το Darwin OS αποτελεί τη θεμελιώδη βάση πάνω στην οποία οικοδομείται το macOS και συνιστά τον πυρήνα της αρχιτεκτονικής του σε επίπεδο λειτουργικού συστήματος. Πρόκειται για ένα ανοικτού κώδικα, Unix-like λειτουργικό σύστημα, το οποίο ενσωματώνει κρίσιμα υποσυστήματα χαμηλού επιπέδου, όπως ο πυρήνας, η διαχείριση μνήμης, το σύστημα διεργασιών και τα βασικά interfaces προς το υλικό. Η επιλογή του Darwin ως βάσης του macOS δεν είναι τυχαία, καθώς επιτρέπει τον συνδυασμό ώριμων αρχών ασφάλειας τύπου Unix με σύγχρονες, εμπορικά ελεγχόμενες επεκτάσεις που υλοποιούνται στα ανώτερα επίπεδα του συστήματος.

Αρχιτεκτονικά, το Darwin βασίζεται στον υβριδικό πυρήνα XNU, ο οποίος συνδυάζει στοιχεία μικροπυρήνα Mach με υποσυστήματα τύπου BSD. Η υβριδική αυτή προσέγγιση επιτρέπει την απομόνωση κρίσιμων λειτουργιών, όπως η διαχείριση μνήμης και διεργασιών, ενώ ταυτόχρονα διατηρεί την απόδοση και τη συμβατότητα που χαρακτηρίζουν τα μονολιθικά συστήματα. Από θεωρητική σκοπιά, η επιλογή αυτή αντανακλά μια συνειδητή ισορροπία μεταξύ ασφάλειας, απόδοσης και πολυπλοκότητας, η οποία είναι κεντρική στη σύγχρονη σχεδίαση λειτουργικών συστημάτων.

Η ασφάλεια στο Darwin OS εδράζεται πρωτίστως στη διαχείριση της εικονικής μνήμης και στον αυστηρό διαχωρισμό του χώρου διευθύνσεων μεταξύ πυρήνα και διεργασιών χρήστη. Η υλοποίηση της μετάφρασης εικονικών σε φυσικές διευθύνσεις στο Darwin αξιοποιεί τις δυνατότητες της υποκείμενης αρχιτεκτονικής, ιδίως σε σύγχρονα συστήματα ARM64, όπου η υποστήριξη πολυεπίπεδων πινάκων σελίδων και hardware-assisted ελέγχων πρόσβασης επιτρέπει λεπτομερή έλεγχο ανάγνωσης, εγγραφής και εκτέλεσης μνήμης (ARM, 2019; Apple, 2024a). Η αρχιτεκτονική αυτή αποτελεί βασικό πυλώνα για την υλοποίηση μηχανισμών προστασίας, όπως η απομόνωση διεργασιών και η τυχαιοποίηση της διάταξης του χώρου διευθύνσεων.

Στο πλαίσιο της θεωρίας ασφάλειας λειτουργικών συστημάτων, το Darwin OS μπορεί να θεωρηθεί ως υπόστρωμα πάνω στο οποίο εφαρμόζονται οι αρχές της άμυνας σε βάθος. Οι βασικοί μηχανισμοί που υλοποιούνται σε επίπεδο πυρήνα λειτουργούν ως το πρώτο και πιο κρίσιμο επίπεδο άμυνας, περιορίζοντας τις δυνατότητες εκμετάλλευσης σφαλμάτων χαμηλού επιπέδου. Μελέτες έχουν δείξει ότι η αποτελεσματικότητα τέτοιων μηχανισμών εξαρτάται σε μεγάλο βαθμό από την ποιότητα της τυχαιοποίησης και την πραγματική εντροπία που επιτυγχάνεται στην πράξη (Herlands et al., 2014; Acharya et al., 2017). Υπό αυτή την οπτική, το Darwin παρέχει το αναγκαίο τεχνικό υπόβαθρο για την υλοποίηση ισχυρών αμυντικών τεχνικών, χωρίς όμως να εξαλείφει πλήρως τις θεωρητικές αδυναμίες που έχουν αναδειχθεί στη βιβλιογραφία.

Ιδιαίτερη σημασία έχει και η κληρονομιά του BSD στο Darwin, η οποία επηρεάζει άμεσα το μοντέλο διαχείρισης χρηστών, ομάδων και δικαιωμάτων. Το παραδοσιακό μοντέλο DAC που υιοθετείται στα Unix-like συστήματα αποτελεί βασικό στοιχείο της λειτουργίας του Darwin, παρέχοντας ένα γνώριμο και ευρέως δοκιμασμένο πλαίσιο ελέγχου πρόσβασης. Ωστόσο, όπως έχει αναδειχθεί σε πλήθος ερευνών, το DAC από μόνο του δεν επαρκεί για την αντιμετώπιση σύγχρονων απειλών, ιδίως σε περιβάλλοντα όπου οι επιθέσεις εκμεταλλεύονται σφάλματα μνήμης και οδηγούν σε αυθαίρετη εκτέλεση κώδικα (Marco-Gisbert & Ripoll, 2019). Το Darwin

λειτουργεί, επομένως, ως απαραίτητη αλλά όχι επαρκής βάση, πάνω στην οποία απαιτούνται πρόσθετα επίπεδα προστασίας.

Η σχέση του Darwin OS με το macOS μπορεί να αναλυθεί και μέσα από το πρίσμα της θεωρητικής διάκρισης μεταξύ πυρήνα και πολιτικών ασφάλειας. Το Darwin υλοποιεί κυρίως μηχανισμούς, δηλαδή τεχνικά μέσα επιβολής περιορισμών, ενώ οι πολιτικές ασφάλειας σε μεγάλο βαθμό καθορίζονται και επιβάλλονται στα ανώτερα επίπεδα του macOS. Η διάκριση αυτή ευθυγραμμίζεται με σύγχρονες θεωρητικές προσεγγίσεις στην ασφάλεια λειτουργικών συστημάτων, σύμφωνα με τις οποίες η αποσύνδεση μηχανισμών και πολιτικών αυξάνει την ευελιξία και μειώνει τον κίνδυνο συστημικών σφαλμάτων.

Η σημασία του Darwin OS γίνεται ιδιαίτερα εμφανής όταν εξεταστούν οι επιπτώσεις ευπαθειών χαμηλού επιπέδου. Καταγεγραμμένα περιστατικά δείχνουν ότι σφάλματα σε θεμελιώδη υποσυστήματα, όπως η διαχείριση μνήμης ή οι βιβλιοθήκες συστήματος, μπορούν να οδηγήσουν σε σοβαρές παραβιάσεις ασφάλειας, ανεξαρτήτως των ανώτερων μηχανισμών προστασίας (Qualys Security Advisory, 2023). Αυτό υπογραμμίζει τον ρόλο του Darwin ως κρίσιμου σημείου άμυνας και ταυτόχρονα ως πιθανού σημείου αποτυχίας, εάν δεν συνοδεύεται από αυστηρό σχεδιασμό και συνεχή αξιολόγηση.

Στο ευρύτερο πλαίσιο της ασφάλειας λειτουργικών συστημάτων, το Darwin OS τοποθετείται ως ένα σύγχρονο, υβριδικό υπόδειγμα που συνδυάζει παραδοσιακές αρχές Unix με αρχιτεκτονικές επιλογές προσανατολισμένες στην ασφάλεια. Η ύπαρξη ανοικτού κώδικα σε βασικά υποσυστήματα επιτρέπει τη δημόσια αξιολόγηση και ανάλυση, ενώ η ενσωμάτωσή του σε ένα κλειστό εμπορικό οικοσύστημα δημιουργεί ένα ενδιαφέρον πεδίο μελέτης ως προς τη σχέση διαφάνειας, ελέγχου και ασφάλειας. Η κατανόηση του Darwin OS ως βάσης του macOS αποτελεί, επομένως, αναγκαίο βήμα για την εις βάθος ανάλυση των επιμέρους μηχανισμών ασφάλειας που θα εξεταστούν στις επόμενες ενότητες του παρόντος κεφαλαίου.

3.2 Ο υβριδικός πυρήνας XNU (Mach + BSD)

Ο πυρήνας XNU αποτελεί την καρδιά του Darwin OS και, κατ’ επέκταση, του macOS, υλοποιώντας μια υβριδική αρχιτεκτονική που συνδυάζει τον μικροπυρήνα Mach με υποσυστήματα τύπου BSD. Η σχεδιαστική αυτή επιλογή εντάσσεται στη μακρά θεωρητική αντιπαράθεση μεταξύ μικροπυρηνικών και μονολιθικών πυρήνων, επιχειρώντας να αντλήσει πλεονεκτήματα και από τις δύο προσεγγίσεις. Από τη μία πλευρά, ο Mach εισάγει έννοιες όπως η απομόνωση βασικών λειτουργιών, η επικοινωνία μέσω μηνυμάτων και η αφαιρετική διαχείριση πόρων, ενώ από την άλλη, το BSD παρέχει ώριμους και αποδοτικούς μηχανισμούς για διεργασίες, αρχεία, δικτυακές στοίβες και παραδοσιακό έλεγχο πρόσβασης.

Αρχιτεκτονικά, ο Mach στον XNU αναλαμβάνει θεμελιώδεις λειτουργίες χαμηλού επιπέδου, όπως τη διαχείριση διεργασιών και νημάτων, τη χρονοδρομολόγηση και τον έλεγχο της εικονικής μνήμης. Η έμφαση στη διαχείριση μνήμης καθιστά τον Mach κρίσιμο για την υλοποίηση των βασικών μηχανισμών ασφάλειας του macOS, καθώς ο τρόπος με τον οποίο οργανώνεται και προστατεύεται ο χώρος διευθύνσεων επηρεάζει άμεσα την απομόνωση διεργασιών και την ανθεκτικότητα απέναντι σε επιθέσεις εκμετάλλευσης μνήμης. Η αξιοποίηση σύγχρονων χαρακτηριστικών αρχιτεκτονικών ARM64, όπως οι πολυεπίπεδοι πίνακες σελίδων και οι hardware-assisted έλεγχοι πρόσβασης, επιτρέπει στο XNU να εφαρμόζει λεπτομερείς πολιτικές ανάγνωσης, εγγραφής και εκτέλεσης μνήμης (ARM, 2019; Apple, 2024a).

Η θεωρητική σημασία του Mach στον τομέα της ασφάλειας έγκειται στη σαφή διάκριση μεταξύ μηχανισμών και πολιτικών. Ο μικροπυρηνικός σχεδιασμός επιδιώκει να περιορίσει το μέγεθος του αξιόπιστου υπολογιστικού πυρήνα, μειώνοντας κατ' αρχήν την επιφάνεια επίθεσης. Ωστόσο, στην πράξη, ο XNU δεν αποτελεί «καθαρό» μικροπυρήνα, καθώς μεγάλο μέρος της λειτουργικότητας εκτελείται εντός του ίδιου χώρου πυρήνα για λόγους απόδοσης. Η υβριδική αυτή προσέγγιση αντανακλά έναν συνειδητό συμβιβασμό, όπου η θεωρητική ασφάλεια εξισορροπείται με πρακτικές απαιτήσεις απόδοσης και συμβατότητας.

Το υποσύστημα BSD ενσωματώνεται στον XNU για να παρέχει τις κλασικές λειτουργίες ενός Unix-like λειτουργικού συστήματος, συμπεριλαμβανομένης της διαχείρισης χρηστών, ομάδων και δικαιωμάτων, καθώς και των συστημικών κλήσεων που χρησιμοποιούνται από τις εφαρμογές. Το μοντέλο Discretionary Access Control που κληρονομείται από το BSD αποτελεί βασικό στοιχείο της ασφάλειας του macOS, αλλά, όπως έχει αναδειχθεί στη βιβλιογραφία, παρουσιάζει εγγενείς αδυναμίες όταν εξετάζεται απομονωμένα από άλλους μηχανισμούς προστασίας (Marco-Gisbert & Ripoll, 2019). Στο πλαίσιο του XNU, το BSD λειτουργεί συμπληρωματικά προς τους μηχανισμούς απομόνωσης μνήμης του Mach, δημιουργώντας ένα πολυεπίπεδο σύστημα ελέγχου.

Η αλληλεπίδραση Mach και BSD στον XNU έχει ιδιαίτερη σημασία για τη στρατηγική Defense-in-Depth που υιοθετεί το macOS. Οι μηχανισμοί του Mach περιορίζουν τις δυνατότητες αυθαίρετης πρόσβασης στη μνήμη και επιβάλλουν σαφή όρια μεταξύ διεργασιών, ενώ το BSD προσθέτει ένα επιπλέον επίπεδο ελέγχου μέσω πολιτικών πρόσβασης και διαχείρισης πόρων. Η συνδυαστική αυτή λειτουργία αυξάνει το κόστος και την πολυπλοκότητα των επιθέσεων, καθώς ένας επιτιθέμενος καλείται να παρακάμψει τόσο τους μηχανισμούς μνήμης όσο και τις πολιτικές πρόσβασης για να επιτύχει πλήρη έλεγχο του συστήματος (Herlands et al., 2014).

Παρά τα πλεονεκτήματα της υβριδικής αρχιτεκτονικής, η βιβλιογραφία έχει δείξει ότι η αποτελεσματικότητα των μηχανισμών ασφάλειας εξαρτάται σε μεγάλο βαθμό από την ποιότητα της υλοποίησης και από την πραγματική εντροπία που επιτυγχάνεται στην πράξη. Μελέτες που εξετάζουν την τυχαιοποίηση του χώρου διευθύνσεων σε διαφορετικά λειτουργικά συστήματα καταδεικνύουν ότι ακόμη και μικρές προβλεψιμότητες μπορούν να υπονομεύσουν την ανθεκτικότητα των μηχανισμών του πυρήνα (Díaz et al., 2021; Acharya et al., 2017). Υπό αυτή την οπτική, ο XNU δεν αποτελεί εξαίρεση, αλλά εντάσσεται στο γενικότερο πλαίσιο των λειτουργικών συστημάτων που καλούνται να ισορροπήσουν μεταξύ θεωρητικής ασφάλειας και πρακτικών περιορισμών.

Η διαχείριση μνήμης στον XNU έχει εξελιχθεί σημαντικά τα τελευταία χρόνια, ενσωματώνοντας σύγχρονες τεχνικές οργάνωσης και απομόνωσης φυσικών και εικονικών σελίδων. Η μετάβαση προς πιο αποδοτικές δομές, όπως οι memory folios, αποσκοπεί όχι μόνο στη βελτίωση της απόδοσης, αλλά και στη σαφέστερη οριοθέτηση και διαχείριση της μνήμης, στοιχείο που έχει άμεσες επιπτώσεις στην ασφάλεια (Wilcox, 2021; Torvalds, 2021b). Παρότι οι συγκεκριμένες εξελίξεις αφορούν πρωτίστως τον ευρύτερο χώρο των Unix-like πυρήνων, το θεωρητικό τους υπόβαθρο είναι συναφές με τις επιλογές που υλοποιούνται και στον XNU.

Ο υβριδικός πυρήνας XNU μπορεί να θεωρηθεί ως θεμελιώδης δομικός λίθος της ασφάλειας του macOS, ενσωματώνοντας τόσο θεωρητικές αρχές μικροπυρηνικού σχεδιασμού όσο και πρακτικές υλοποιήσεις τύπου BSD. Η αρχιτεκτονική αυτή δεν εξαλείφει τις εγγενείς προκλήσεις της ασφάλειας λειτουργικών συστημάτων, αλλά παρέχει ένα ισχυρό και ευέλικτο υπόβαθρο πάνω στο οποίο οικοδομούνται οι ανώτεροι μηχανισμοί προστασίας του macOS. Η κατανόηση της λειτουργίας και των ορίων του XNU αποτελεί, επομένως, κρίσιμο βήμα για την ανάλυση των

επιμέρους τεχνικών ασφάλειας που θα εξεταστούν στις επόμενες ενότητες του παρόντος κεφαλαίου.

3.3 Διαχείριση διεργασιών και μνήμης

Η διαχείριση διεργασιών και μνήμης στο macOS αποτελεί έναν από τους κεντρικούς άξονες της αρχιτεκτονικής του και συνιστά κρίσιμο παράγοντα για τη συνολική ασφάλεια του συστήματος. Στο επίπεδο του πυρήνα XNU, οι διεργασίες και τα νήματα οργανώνονται και ελέγχονται με τρόπο που επιδιώκει να εξασφαλίσει τόσο την αποδοτική αξιοποίηση των πόρων όσο και την αυστηρή απομόνωση μεταξύ διαφορετικών εκτελεστικών οντοτήτων. Η απομόνωση αυτή δεν αποτελεί απλώς ζήτημα ορθής λειτουργίας, αλλά θεμελιώδη προϋπόθεση για την αποτροπή επιθέσεων που εκμεταλλεύονται σφάλματα μνήμης ή μη εξουσιοδοτημένη πρόσβαση σε κρίσιμους πόρους.

Σε επίπεδο διεργασιών, το macOS ακολουθεί το κλασικό μοντέλο Unix, όπου κάθε διεργασία εκτελείται σε ξεχωριστό εικονικό χώρο διευθύνσεων και διαθέτει μοναδικό αναγνωριστικό. Ο διαχωρισμός αυτός επιτρέπει στον πυρήνα να επιβάλλει σαφή όρια μεταξύ διεργασιών χρήστη και διεργασιών συστήματος, περιορίζοντας τη δυνατότητα άμεσης αλληλεπίδρασης ή παρέμβασης. Η δημιουργία νέων διεργασιών και νημάτων πραγματοποιείται μέσω μηχανισμών που κληρονομούν επιλεκτικά πόρους και δικαιώματα, στοιχείο που συνδέεται άμεσα με την αρχή του ελάχιστου προνομίου και με τον έλεγχο πρόσβασης σε επίπεδο πυρήνα (Kerrisk, 2023).

Η διαχείριση μνήμης στο macOS βασίζεται στην έννοια της εικονικής μνήμης, όπου κάθε διεργασία αντιλαμβάνεται έναν ιδιωτικό, συνεχόμενο χώρο διευθύνσεων ανεξάρτητα από τη φυσική διάταξη της μνήμης. Η μετάφραση εικονικών σε φυσικές διευθύνσεις υλοποιείται με τη συνδρομή του υλικού και ειδικότερα της μονάδας διαχείρισης μνήμης (MMU), η οποία εφαρμόζει πολιτικές πρόσβασης σε επίπεδο σελίδας. Η υποστήριξη σύγχρονων αρχιτεκτονικών, όπως η ARM64, επιτρέπει στο macOS να αξιοποιεί πολυεπίπεδους πίνακες σελίδων και λεπτομερείς σημαίες πρόσβασης, ενισχύοντας την απομόνωση και μειώνοντας τον κίνδυνο μη εξουσιοδοτημένης ανάγνωσης ή εγγραφής μνήμης (ARM, 2019; Apple, 2024a).

Η ασφάλεια της διαχείρισης μνήμης συνδέεται στενά με την τυχαιοποίηση της διάταξης του χώρου διευθύνσεων. Η ASLR εφαρμόζεται σε πολλαπλά επίπεδα, καλύπτοντας τη θέση του κώδικα, των βιβλιοθηκών, της στοίβας και του σωρού. Ωστόσο, η βιβλιογραφία έχει δείξει ότι η θεωρητική ισχύς της ASLR εξαρτάται από την πραγματική εντροπία που επιτυγχάνεται και από τους περιορισμούς που επιβάλλονται από την αρχιτεκτονική και την υλοποίηση (Acharya et al., 2017; Díaz et al., 2021). Σε περιπτώσεις όπου η τυχαιοποίηση είναι προβλέψιμη ή περιορισμένη, ένας επιτιθέμενος μπορεί να ανακτήσει κρίσιμες διευθύνσεις και να υπονομεύσει την απομόνωση διεργασιών.

Η διαχείριση μνήμης στο macOS περιλαμβάνει επίσης μηχανισμούς προστασίας που αποτρέπουν την εκτέλεση κώδικα σε μη εκτελέσιμες περιοχές και περιορίζουν την τροποποίηση κρίσιμων δομών. Οι πολιτικές αυτές εφαρμόζονται σε συνεργασία με το υλικό και αποτελούν βασικό στοιχείο της άμυνας σε βάθος, καθώς δυσχεραίνουν την εκμετάλλευση ευπαθειών τύπου buffer overflow ή use-after-free. Παρά ταύτα, εμπειρικές μελέτες έχουν καταδείξει ότι η ύπαρξη σφαλμάτων στη διαχείριση μνήμης παραμένει μία από τις βασικές αιτίες επιτυχημένων επιθέσεων, ιδίως όταν συνδυάζεται με τεχνικές παράκαμψης μηχανισμών προστασίας (Marco-Gisbert & Ripoll, 2019).

Σημαντική διάσταση της διαχείρισης διεργασιών και μνήμης είναι και η δυναμική φύση του συστήματος. Η φόρτωση βιβλιοθηκών κατά τον χρόνο εκτέλεσης, η δημιουργία και καταστροφή διεργασιών και η ανακατανομή μνήμης δημιουργούν ένα συνεχώς μεταβαλλόμενο περιβάλλον, στο οποίο οι μηχανισμοί ασφάλειας καλούνται να λειτουργήσουν αξιόπιστα. Μελέτες που εξετάζουν τη συμπεριφορά της ASLR και συναφών μηχανισμών σε πραγματικά σενάρια χρήσης δείχνουν ότι η πολυπλοκότητα αυτή μπορεί να οδηγήσει σε αποκλίσεις μεταξύ θεωρητικής και πρακτικής ασφάλειας (Herlands et al., 2014).

Η διαχείριση μνήμης και διεργασιών συνδέεται άμεσα και με την απόδοση και τη διαθεσιμότητα του συστήματος. Οι επιλογές που γίνονται σε επίπεδο πυρήνα επηρεάζουν όχι μόνο την ασφάλεια, αλλά και την εμπειρία χρήστη, καθώς αυστηρότεροι περιορισμοί μπορούν να επιβαρύνουν την απόδοση ή να οδηγήσουν σε ανεπιθύμητους τερματισμούς εφαρμογών. Η ισορροπία αυτή αποτελεί διαχρονική πρόκληση στη σχεδίαση λειτουργικών συστημάτων και αντανακλά τη γενικότερη θεωρητική αντίληψη ότι η ασφάλεια αποτελεί αποτέλεσμα συμβιβασμών μεταξύ αντικρουόμενων απαιτήσεων (Wilcox, 2021).

Η διαχείριση διεργασιών και μνήμης στο macOS συνιστά θεμελιώδη μηχανισμό ασφάλειας που επηρεάζει άμεσα την ανθεκτικότητα του συστήματος απέναντι σε σύγχρονες απειλές. Η αποτελεσματικότητά της εξαρτάται από τη στενή συνεργασία λογισμικού και υλικού, από την ποιότητα της υλοποίησης και από τη συνεχή προσαρμογή στις εξελισσόμενες τεχνικές επίθεσης. Η κατανόηση των μηχανισμών αυτών αποτελεί απαραίτητο υπόβαθρο για την ανάλυση των ειδικότερων τεχνικών ασφάλειας του macOS που θα εξεταστούν στις επόμενες ενότητες του παρόντος κεφαλαίου.

3.4 Μηχανισμοί απομόνωσης (process isolation)

Η απομόνωση διεργασιών αποτελεί θεμελιώδη μηχανισμό ασφάλειας στα σύγχρονα λειτουργικά συστήματα και συνιστά έναν από τους βασικούς πυλώνες πάνω στους οποίους στηρίζεται η αρχιτεκτονική του macOS. Η έννοια της process isolation αναφέρεται στη δυνατότητα του λειτουργικού συστήματος να διασφαλίζει ότι κάθε διεργασία εκτελείται σε ένα αυστηρά οριοθετημένο περιβάλλον, χωρίς άμεση ή έμμεση πρόσβαση στη μνήμη, στους πόρους ή στην εκτελεστική κατάσταση άλλων διεργασιών. Η απομόνωση αυτή δεν αποτελεί απλώς μέτρο σταθερότητας, αλλά κρίσιμη προϋπόθεση για την αποτροπή επιθέσεων που στοχεύουν στη διάδοση κακόβουλου κώδικα ή στην κλιμάκωση προνομίων.

Στο macOS, η process isolation υλοποιείται πρωτίστως μέσω της αρχιτεκτονικής εικονικής μνήμης που παρέχει ο πυρήνας XNU. Κάθε διεργασία διαθέτει έναν ανεξάρτητο εικονικό χώρο διευθύνσεων, ο οποίος χαρτογραφείται σε φυσική μνήμη με τρόπο που αποτρέπει την άμεση πρόσβαση από άλλες διεργασίες. Η μονάδα διαχείρισης μνήμης του υλικού επιβάλλει περιορισμούς ανάγνωσης, εγγραφής και εκτέλεσης σε επίπεδο σελίδας, καθιστώντας τεχνικά αδύνατη την αυθαίρετη προσπέλαση μνήμης χωρίς παρέμβαση του πυρήνα (ARM, 2019; Apple, 2024a). Η προσέγγιση αυτή αποτελεί τον πρώτο και πιο θεμελιώδη φραγμό απέναντι σε επιθέσεις εκμετάλλευσης μνήμης.

Η απομόνωση διεργασιών ενισχύεται περαιτέρω από τον σαφή διαχωρισμό μεταξύ χώρου χρήστη και χώρου πυρήνα. Ο πυρήνας εκτελείται σε προνομιακό επίπεδο, στο οποίο δεν μπορούν να αποκτήσουν άμεση πρόσβαση οι διεργασίες χρήστη. Οι αλληλεπιδράσεις μεταξύ των δύο αυτών επιπέδων πραγματοποιούνται αποκλειστικά μέσω ελεγχόμενων μηχανισμών, όπως οι system calls, οι οποίοι λειτουργούν ως αυστηρά οριοθετημένα σημεία εισόδου. Ο διαχωρισμός αυτός περιορίζει δραστικά τη δυνατότητα μιας διεργασίας να επηρεάσει τη

λειτουργία του πυρήνα, ακόμη και σε περιπτώσεις όπου έχουν παραβιαστεί επιμέρους μηχανισμοί προστασίας.

Η θεωρητική σημασία της process isolation αναδεικνύεται ιδιαίτερα στο πλαίσιο των επιθέσεων χαμηλού επιπέδου. Πολλές σύγχρονες τεχνικές εκμετάλλευσης βασίζονται στη δυνατότητα ενός επιτιθέμενου να μεταβεί από έναν περιορισμένο χώρο εκτέλεσης σε έναν ευρύτερο ή πιο προνομιούχο. Η αυστηρή απομόνωση διεργασιών αυξάνει σημαντικά το κόστος τέτοιων επιθέσεων, καθώς απαιτείται η επιτυχής παράκαμψη πολλαπλών μηχανισμών για την επίτευξη πλήρους ελέγχου (Herlands et al., 2014; Marco-Gisbert & Ripoll, 2019). Υπό αυτή την οπτική, η process isolation λειτουργεί ως κρίσιμο στοιχείο της στρατηγικής Defense-in-Depth.

Η πρακτική αποτελεσματικότητα της απομόνωσης διεργασιών εξαρτάται, ωστόσο, από την ποιότητα της υλοποίησης και από την απουσία σφαλμάτων στον πυρήνα και στα υποκείμενα υποσυστήματα. Η βιβλιογραφία έχει καταδείξει ότι ευπάθειες στη διαχείριση μνήμης ή στη μετάφραση διευθύνσεων μπορούν να υπονομεύσουν ακόμη και αυστηρά σχεδιασμένα μοντέλα απομόνωσης, επιτρέποντας την αυθαίρετη ανάγνωση ή εγγραφή σε ξένη μνήμη (Díaz et al., 2021). Το γεγονός αυτό υπογραμμίζει ότι η process isolation δεν μπορεί να θεωρηθεί απόλυτη, αλλά πρέπει να αξιολογείται ως μέρος ενός ευρύτερου συνόλου αμυντικών μηχανισμών.

Η απομόνωση διεργασιών στο macOS δεν περιορίζεται μόνο στη μνήμη, αλλά επεκτείνεται και στη διαχείριση πόρων και δικαιωμάτων. Κάθε διεργασία εκτελείται με συγκεκριμένο σύνολο προνομίων, τα οποία καθορίζουν την πρόσβασή της σε αρχεία, συσκευές και υπηρεσίες συστήματος. Η κληρονομικότητα αυτών των προνομίων ελέγχεται αυστηρά κατά τη δημιουργία νέων διεργασιών, περιορίζοντας τη δυνατότητα ανεξέλεγκτης εξάπλωσης δικαιωμάτων. Η προσέγγιση αυτή ευθυγραμμίζεται με την αρχή του ελάχιστου προνομίου και ενισχύει τη συνολική απομόνωση του συστήματος (Kerrisk, 2023).

Σημαντικό ρόλο στην ενίσχυση της process isolation διαδραματίζει και η τυχαιοποίηση της διάταξης του χώρου διευθύνσεων. Η ASLR καθιστά δυσχερή την πρόβλεψη κρίσιμων διευθύνσεων μνήμης, μειώνοντας την πιθανότητα επιτυχούς εκμετάλλευσης σφαλμάτων που θα μπορούσαν να παρακάμψουν τα όρια απομόνωσης. Ωστόσο, όπως έχει αναδειχθεί σε σχετικές μελέτες, η αποτελεσματικότητα της ASLR εξαρτάται από την πραγματική εντροπία και μπορεί να υπονομευθεί από προβλεψιμότητες στην υλοποίηση (Acharya et al., 2017; Díaz et al., 2021). Συνεπώς, η τυχαιοποίηση λειτουργεί ως ενισχυτικός, αλλά όχι επαρκής από μόνος του, μηχανισμός απομόνωσης.

Η απομόνωση διεργασιών συνδέεται άμεσα και με τη διαθεσιμότητα του συστήματος. Σε περιπτώσεις σφαλμάτων ή κακόβουλης συμπεριφοράς, η δυνατότητα περιορισμού των επιπτώσεων σε μία μόνο διεργασία αποτρέπει τη γενικευμένη κατάρρευση του συστήματος. Η ιδιότητα αυτή είναι ιδιαίτερα σημαντική σε περιβάλλοντα desktop, όπου η σταθερότητα και η αξιοπιστία αποτελούν βασικές απαιτήσεις. Από θεωρητική σκοπιά, η process isolation λειτουργεί ως μηχανισμός ελεγχόμενης αποτυχίας, επιτρέποντας στο σύστημα να συνεχίσει τη λειτουργία του ακόμη και υπό συνθήκες μερικής παραβίασης.

Οι μηχανισμοί απομόνωσης διεργασιών στο macOS αποτελούν θεμελιώδες στοιχείο της αρχιτεκτονικής ασφάλειας του συστήματος και ενσωματώνουν βασικές αρχές της σύγχρονης θεωρίας OS Security. Η αποτελεσματικότητά τους εξαρτάται από τη στενή συνεργασία υλικού και λογισμικού, από την ποιότητα της υλοποίησης και από τη συνεχή προσαρμογή στις εξελισσόμενες τεχνικές επίθεσης. Η κατανόηση των μηχανισμών αυτών αποτελεί αναγκαίο

υπόβαθρο για την ανάλυση πιο εξειδικευμένων τεχνικών απομόνωσης και περιορισμού που εφαρμόζονται στο macOS και θα εξεταστούν στις επόμενες ενότητες του παρόντος κεφαλαίου.

3.5 Διαχείριση χρηστών, ομάδων και προνομίων

Η διαχείριση χρηστών, ομάδων και προνομίων στο macOS αποτελεί θεμελιώδες στοιχείο της αρχιτεκτονικής ασφάλειας του συστήματος και ενσωματώνει βασικές αρχές της παραδοσιακής Unix φιλοσοφίας σε συνδυασμό με σύγχρονες επεκτάσεις ελέγχου πρόσβασης. Σε επίπεδο Darwin και πυρήνα XNU, το macOS υιοθετεί το κλασικό μοντέλο ταυτοποίησης χρηστών μέσω μοναδικών αναγνωριστικών (UID) και ομάδων (GID), το οποίο λειτουργεί ως βασικός μηχανισμός αντιστοίχισης διεργασιών με δικαιώματα πρόσβασης σε πόρους. Το μοντέλο αυτό επιτρέπει τη σαφή διάκριση μεταξύ διαφορετικών χρηστών και τη συγκρότηση συλλογικών οντοτήτων μέσω ομάδων, διευκολύνοντας την εφαρμογή πολιτικών πρόσβασης με σχετικά χαμηλή πολυπλοκότητα.

Σε θεωρητικό επίπεδο, η διαχείριση χρηστών και ομάδων στο macOS εδράζεται στο Discretionary Access Control, όπου η πρόσβαση σε αρχεία, συσκευές και άλλους πόρους καθορίζεται από τον ιδιοκτήτη του αντικειμένου και από τις ομάδες στις οποίες ανήκει. Το μοντέλο αυτό, αν και ευρέως διαδεδομένο, παρουσιάζει εγγενείς περιορισμούς ασφάλειας, καθώς επιτρέπει τη μεταβίβαση δικαιωμάτων με διακριτική ευχέρεια, δημιουργώντας δυνητικά σημεία κατάχρησης. Η βιβλιογραφία έχει δείξει ότι, σε περιβάλλοντα όπου οι διεργασίες εκτελούνται με υπερβολικά δικαιώματα, ένα επιτυχές σφάλμα μνήμης μπορεί να οδηγήσει σε άμεση παραβίαση της αρχής του ελάχιστου προνομίου (Marco-Gisbert & Ripoll, 2019).

Η έννοια των προνομίων στο macOS εκτείνεται πέρα από τη δυαδική διάκριση μεταξύ απλού χρήστη και υπερχρήστη. Αν και ο λογαριασμός με UID 0 εξακολουθεί να διαθέτει καθολικά δικαιώματα, το σύστημα επιδιώκει τον περιορισμό της χρήσης πλήρους υπερχρήστη μέσω μηχανισμών ελεγχόμενης ανύψωσης προνομίων. Η πρακτική αυτή αποσκοπεί στη μείωση της επιφάνειας επίθεσης, περιορίζοντας τον χρόνο και το εύρος κατά το οποίο μια διεργασία εκτελείται με αυξημένα δικαιώματα. Η σημασία αυτής της προσέγγισης αναδεικνύεται από την ανάλυση πραγματικών περιστατικών κλιμάκωσης προνομίων, όπου η κατάχρηση δικαιωμάτων υπερχρήστη αποτελεί συχνό τελικό στόχο επιθέσεων (Qualys Security Advisory, 2023; CVEdetails.com, 2024).

Η διαχείριση ομάδων επιτρέπει στο macOS να εφαρμόζει πιο λεπτομερείς πολιτικές πρόσβασης, κατανέμοντας δικαιώματα σε σύνολα χρηστών με κοινά χαρακτηριστικά ή ρόλους. Η προσέγγιση αυτή ευθυγραμμίζεται με τη λογική του Role-Based Access Control, καθώς διευκολύνει την εφαρμογή της αρχής του διαχωρισμού καθηκόντων και περιορίζει την ανάγκη εκχώρησης ατομικών δικαιωμάτων σε κάθε χρήστη ξεχωριστά. Παρά ταύτα, η αποτελεσματικότητα του μοντέλου εξαρτάται από τον ορθό σχεδιασμό των ομάδων και από την αποφυγή υπερβολικής συγκέντρωσης προνομίων, η οποία θα μπορούσε να υπονομεύσει την αρχή του ελάχιστου προνομίου.

Η διαχείριση προνομίων στο macOS συνδέεται στενά με τη διαχείριση διεργασιών, καθώς τα δικαιώματα ενός χρήστη μεταβιβάζονται στις διεργασίες που εκτελεί. Ο πυρήνας XNU επιβάλλει αυστηρούς ελέγχους κατά τη δημιουργία και εκτέλεση διεργασιών, διασφαλίζοντας ότι τα προνόμια δεν επεκτείνονται αυθαίρετα πέρα από τα καθορισμένα όρια. Η κληρονομικότητα προνομίων ελέγχεται μέσω μηχανισμών που περιορίζουν την ανεξέλεγκτη διάδοση δικαιωμάτων, στοιχείο που είναι κρίσιμο για την αποτροπή επιθέσεων lateral movement εντός του συστήματος (Kerrisk, 2023).

Η θεωρητική σημασία της διαχείρισης χρηστών και προνομίων αναδεικνύεται και στο πλαίσιο της στρατηγικής Defense-in-Depth. Ακόμη και αν ένας επιτιθέμενος καταφέρει να παρακάμψει μηχανισμούς απομόνωσης ή να εκμεταλλευτεί σφάλματα μνήμης, το επίπεδο προνομίων με το οποίο εκτελείται η παραβιασμένη διεργασία καθορίζει το εύρος των δυνατοτήτων του. Μελέτες έχουν δείξει ότι ο συνδυασμός περιορισμένων προνομίων και αυστηρής απομόνωσης διεργασιών μπορεί να μειώσει δραστικά τις επιπτώσεις μιας επιτυχούς εκμετάλλευσης, ακόμη και όταν άλλοι μηχανισμοί έχουν αποτύχει (Herlands et al., 2014).

Η διαχείριση χρηστών, ομάδων και προνομίων επηρεάζει άμεσα και τη λογοδοσία του συστήματος. Η σαφής αντιστοίχιση ενεργειών σε συγκεκριμένους χρήστες ή ομάδες επιτρέπει την αποτελεσματική καταγραφή και ανάλυση συμβάντων ασφάλειας, στοιχείο που είναι κρίσιμο για τη διερεύνηση περιστατικών και τη βελτίωση των αμυντικών μηχανισμών. Η δυνατότητα απόδοσης ευθύνης λειτουργεί αποτρεπτικά και συμπληρώνει τους προληπτικούς μηχανισμούς ασφάλειας, ενισχύοντας τη συνολική ανθεκτικότητα του συστήματος.

Η διαχείριση χρηστών, ομάδων και προνομίων στο macOS αποτελεί έναν από τους βασικούς άξονες της αρχιτεκτονικής του και ενσωματώνει τόσο παραδοσιακές όσο και σύγχρονες αρχές ασφάλειας λειτουργικών συστημάτων. Η αποτελεσματικότητά της εξαρτάται από τον συνδυασμό ορθού σχεδιασμού πολιτικών, αυστηρής επιβολής σε επίπεδο πυρήνα και συνεπούς εφαρμογής της αρχής του ελάχιστου προνομίου. Η κατανόηση του τρόπου με τον οποίο το macOS διαχειρίζεται χρήστες και προνόμια αποτελεί κρίσιμο υπόβαθρο για την ανάλυση πιο εξειδικευμένων μηχανισμών ασφάλειας που θα εξεταστούν στις επόμενες ενότητες του παρόντος κεφαλαίου.

3.6 Σχέση αρχιτεκτονικής και ασφάλειας

Η σχέση μεταξύ αρχιτεκτονικής και ασφάλειας στο macOS είναι άρρηκτα συνδεδεμένη και καθορίζει σε μεγάλο βαθμό τόσο τις δυνατότητες άμυνας όσο και τα εγγενή όρια του συστήματος. Η αρχιτεκτονική του λειτουργικού συστήματος δεν αποτελεί ουδέτερο υπόβαθρο πάνω στο οποίο προστίθενται εκ των υστέρων μηχανισμοί προστασίας, αλλά ενεργό συντελεστή που διαμορφώνει τις απειλές, τις αμυντικές επιλογές και την αποτελεσματικότητα των πολιτικών ασφάλειας. Στο πλαίσιο αυτό, η ασφάλεια του macOS μπορεί να γίνει κατανοητή μόνο εάν εξεταστεί ως άμεση συνέπεια των αρχιτεκτονικών επιλογών που υλοποιούνται στο επίπεδο του Darwin και του πυρήνα XNU.

Η υβριδική αρχιτεκτονική του XNU, η οποία συνδυάζει στοιχεία μικροπυρήνα Mach με υποσυστήματα τύπου BSD, επηρεάζει άμεσα το μοντέλο απειλών και τις δυνατότητες άμυνας. Από θεωρητική σκοπιά, η διάκριση λειτουργιών και η σαφής οριοθέτηση μεταξύ υποσυστημάτων επιδιώκουν τη μείωση της επιφάνειας επίθεσης και τη συγκράτηση των επιπτώσεων μιας πιθανής παραβίασης. Ωστόσο, όπως έχει αναδειχθεί στη βιβλιογραφία, η πρακτική υλοποίηση υβριδικών πυρήνων συνεπάγεται συμβιβασμούς, καθώς μεγάλο μέρος της λειτουργικότητας εκτελείται στον ίδιο χώρο πυρήνα για λόγους απόδοσης, περιορίζοντας τα θεωρητικά οφέλη της απομόνωσης (Marco-Gisbert & Ripoll, 2019).

Κεντρικό στοιχείο της σχέσης αρχιτεκτονικής και ασφάλειας αποτελεί η διαχείριση μνήμης. Η αρχιτεκτονική εικονικής μνήμης στο macOS, σε συνδυασμό με τις δυνατότητες της υποκείμενης αρχιτεκτονικής ARM64, επιτρέπει την επιβολή αυστηρών περιορισμών πρόσβασης και την εφαρμογή πολιτικών που βασίζονται σε hardware-assisted μηχανισμούς προστασίας (ARM, 2019; Apple, 2024a). Η επιλογή αυτή ενισχύει σημαντικά την απομόνωση διεργασιών και τη

δυσκολία εκμετάλλευσης σφαλμάτων μνήμης, αλλά ταυτόχρονα καθιστά την ασφάλεια εξαρτώμενη από την ορθότητα και την πληρότητα της αρχιτεκτονικής υλοποίησης.

Η έννοια της εντροπίας αναδεικνύεται ως κρίσιμος συνδετικός κρίκος μεταξύ αρχιτεκτονικής και ασφάλειας. Μηχανισμοί όπως η ASLR αντλούν τη θεωρητική τους ισχύ από την τυχαιοποίηση της διάταξης του χώρου διευθύνσεων, όμως η πραγματική τους αποτελεσματικότητα καθορίζεται από αρχιτεκτονικούς περιορισμούς και υλοποιητικές λεπτομέρειες. Μελέτες έχουν δείξει ότι η πραγματική εντροπία συχνά αποκλίνει από τη θεωρητικά αναμενόμενη, δημιουργώντας προβλέψιμα μοτίβα που μπορούν να αξιοποιηθούν από επιτιθέμενους (Acharya et al., 2017; Herlands et al., 2014). Στο πλαίσιο αυτό, η αρχιτεκτονική του macOS δεν εξαλείφει την ανάγκη για επιπλέον επίπεδα άμυνας, αλλά καθορίζει τα όρια εντός των οποίων οι μηχανισμοί αυτοί μπορούν να λειτουργήσουν αποτελεσματικά.

Η σχέση αρχιτεκτονικής και ασφάλειας γίνεται ιδιαίτερα εμφανής στις επιθέσεις κλιμάκωσης προνομίων. Η αρχιτεκτονική διάκριση μεταξύ χώρου χρήστη και χώρου πυρήνα, καθώς και ο τρόπος με τον οποίο υλοποιούνται οι μεταβάσεις μεταξύ αυτών των επιπέδων, καθορίζουν το πόσο εύκολα μπορεί μια ευπάθεια να μετατραπεί σε πλήρη έλεγχο του συστήματος. Καταγεγραμμένες περιπτώσεις εκμετάλλευσης δείχνουν ότι αρχιτεκτονικές αδυναμίες ή ασαφείς οριοθετήσεις αποτελούν κρίσιμα σημεία αποτυχίας, ανεξαρτήτως των ανώτερων πολιτικών ασφάλειας (Qualys Security Advisory, 2023).

Η αρχιτεκτονική επηρεάζει επίσης τη δυνατότητα εφαρμογής της στρατηγικής Defense-in-Depth. Η ύπαρξη σαφώς διακριτών επιπέδων, τόσο σε επίπεδο λογισμικού όσο και υλικού, επιτρέπει τη συνδυαστική χρήση πολλαπλών μηχανισμών άμυνας που λειτουργούν συμπληρωματικά. Παρά ταύτα, η βιβλιογραφία έχει καταδείξει ότι η υπερβολική εξάρτηση από ομοιογενείς αρχιτεκτονικές παραδοχές μπορεί να οδηγήσει σε ταυτόχρονη αποτυχία πολλαπλών επιπέδων άμυνας, ιδίως σε επιθέσεις χαμηλού επιπέδου ή πλευρικών καναλιών (Yu et al., 2023). Το γεγονός αυτό υπογραμμίζει ότι η αρχιτεκτονική πρέπει να σχεδιάζεται με γνώμονα όχι μόνο την απόδοση και τη λειτουργικότητα, αλλά και την ετερογένεια των αμυντικών μηχανισμών.

Η σχέση αρχιτεκτονικής και ασφάλειας στο macOS επεκτείνεται και στη διαθεσιμότητα του συστήματος. Οι αρχιτεκτονικές επιλογές που ευνοούν την αυστηρή απομόνωση και τον περιορισμό προνομίων μπορούν να συμβάλουν στη σταθερότητα και στην ανθεκτικότητα απέναντι σε σφάλματα, αλλά ενδέχεται να εισάγουν επιπτώσεις στην απόδοση ή στη χρηστικότητα. Η ισορροπία αυτή αποτελεί διαχρονική πρόκληση στη σχεδίαση λειτουργικών συστημάτων και αντανακλά τη θεωρητική παραδοχή ότι η ασφάλεια δεν μπορεί να μεγιστοποιηθεί ανεξάρτητα από άλλες παραμέτρους του συστήματος (Wilcox, 2021).

Η ασφάλεια του macOS δεν μπορεί να θεωρηθεί ανεξάρτητη από την αρχιτεκτονική του, αλλά αναδύεται ως αποτέλεσμα των θεμελιωδών σχεδιαστικών επιλογών που έχουν υιοθετηθεί σε όλα τα επίπεδα του συστήματος. Η κατανόηση της σχέσης αυτής επιτρέπει μια πιο ρεαλιστική αποτίμηση των δυνατοτήτων και των ορίων της ασφάλειας, μετατοπίζοντας την ανάλυση από μεμονωμένους μηχανισμούς σε μια ολιστική θεώρηση του λειτουργικού συστήματος. Η ενότητα αυτή λειτουργεί ως συνδετικός κρίκος μεταξύ της αρχιτεκτονικής ανάλυσης του Κεφαλαίου 3 και της πιο εξειδικευμένης μελέτης των επιμέρους μηχανισμών ασφάλειας που θα ακολουθήσουν στα επόμενα κεφάλαια.

4 .Εμπιστευτικότητα Δεδομένων στο macOS

Η εμπιστευτικότητα δεδομένων στο macOS δεν υλοποιείται από έναν μόνο μηχανισμό, αλλά από μια αλληλουχία τεχνολογιών που ξεκινά από το σύστημα αρχείων και καταλήγει σε hardware-backed κλειδιά και πολιτικές πρόσβασης. Για να αξιολογηθεί σωστά η ασφάλεια της πλατφόρμας, πρέπει να εξεταστεί πώς συνεργάζονται το APFS, οι μηχανισμοί Data Protection, το FileVault, το Secure Enclave και η Keychain και πώς το καθένα απαντά σε διαφορετικό σενάριο απειλής. Το παρόν κεφάλαιο αναλύει αυτήν ακριβώς τη στοίβα εμπιστευτικότητας, με στόχο να αποσαφηνίσει όχι μόνο τι προστατεύεται, αλλά και πότε, από ποιον και υπό ποιες προϋποθέσεις (Apple, 2026d; Apple, 2026a).

4.1 Το APFS και η φιλοσοφία σχεδίασής του

Το Apple File System (APFS) σχεδιάστηκε ως σύγχρονο σύστημα αρχείων για flash storage και για ενοποιημένη διαχείριση αποθηκευτικών μέσων σε όλο το οικοσύστημα της Apple. Βασικά χαρακτηριστικά του, όπως το copy-on-write, τα clones, τα snapshots και το space sharing, δεν εξυπηρετούν μόνο επιδόσεις και ευελιξία, αλλά δημιουργούν και ένα πιο ελεγχόμενο περιβάλλον για ατομικές μεταβολές, αναίρεση αλλαγών και ασφαλέστερη συνύπαρξη συστημικού και χρήστη data state. Το APFS είναι, επομένως, κάτι περισσότερο από τεχνική αντικατάσταση του HFS+: είναι το υπόστρωμα πάνω στο οποίο οικοδομούνται κρίσιμοι μηχανισμοί ασφάλειας του macOS (Apple, 2020a; Apple, 2024j).

Από πλευράς ασφάλειας, η σημαντικότερη συνεισφορά του APFS είναι ότι επιτρέπει στο macOS να οργανώνει διακριτά volumes με κοινό container, να υιοθετεί snapshots για ασφαλείς ενημερώσεις και να ενσωματώνει εγγενώς κρυπτογράφηση σε επίπεδο συστήματος αρχείων. Το γεγονός ότι το system volume και το data volume συνυπάρχουν ως διακριτές οντότητες διευκολύνει τόσο την ακεραιότητα του συστήματος όσο και τη σαφέστερη οριοθέτηση του τι θεωρείται λειτουργικό περιεχόμενο και τι δεδομένο χρήστη. Αυτή η σχεδιαστική απόφαση επηρεάζει άμεσα και την εμπιστευτικότητα, επειδή επιτρέπει στο λειτουργικό να εφαρμόζει διαφορετικές πολιτικές κρυπτογράφησης και πρόσβασης ανά κατηγορία περιεχομένου (Apple, 2022b; Apple, 2024j).

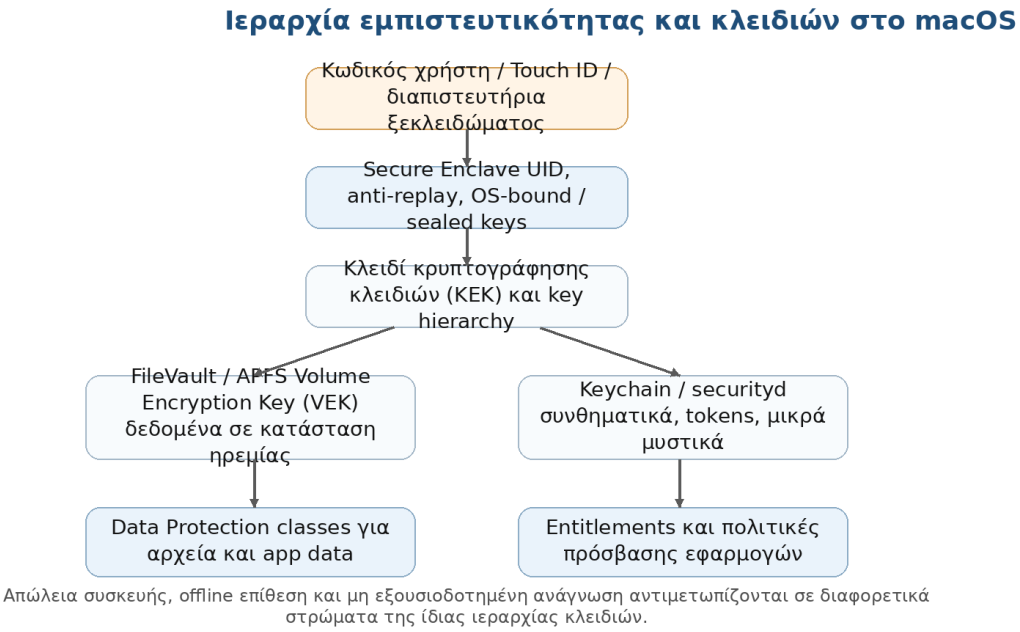
Ωστόσο, το APFS από μόνο του δεν αρκεί για να εγγυηθεί εμπιστευτικότητα. Ένα σύγχρονο σύστημα αρχείων μπορεί να προσφέρει οργανωτικές και κρυπτογραφικές δυνατότητες, αλλά το εάν τα δεδομένα θα παραμείνουν πράγματι απόρρητα εξαρτάται από το πώς συνδέονται τα volume keys με συνθηματικά, secure tokens, hardware secrets και υπηρεσίες διαχείρισης κλειδιών. Γι' αυτό η αξιολόγηση της εμπιστευτικότητας στο macOS δεν σταματά στη δομή του APFS, αλλά συνεχίζεται με το πώς το λειτουργικό ενεργοποιεί και διαχειρίζεται κρυπτογράφηση δίσκου, κλάσεις προστασίας και αποθήκευση μυστικού υλικού (Apple, 2025a; Apple, 2026a).

4.2 Κρυπτογράφηση σε επίπεδο αρχείων και δίσκου

Η κρυπτογράφηση στο macOS είναι πολυεπίπεδη. Στο χαμηλότερο επίπεδο υπάρχει η προστασία ολόκληρου volume, η οποία είναι κρίσιμη για απειλές όπως απώλεια φορητού υπολογιστή, αφαίρεση του δίσκου ή offline απόπειρα ανάγνωσης δεδομένων. Σε ανώτερο επίπεδο, η πλατφόρμα υποστηρίζει κλάσεις προστασίας και διαφοροποιημένη συμπεριφορά για αρχεία και app data, ώστε η πρόσβαση σε ορισμένα κλειδιά να συνδέεται με την κατάσταση ξεκλειδώματος του χρήστη ή με ειδικότερα σενάρια εφαρμογής. Έτσι, η εμπιστευτικότητα δεν περιορίζεται σε ένα απλό «δίσκος κρυπτογραφημένος / μη κρυπτογραφημένος», αλλά κατανέμεται σε στρώματα (Apple, 2024b; Apple, 2024c).

Ιδιαίτερο ενδιαφέρον παρουσιάζει το γεγονός ότι σε Mac με Apple silicon η Data Protection προεπιλέγεται σε κλάση C, αλλά στηρίζεται σε volume key αντί για πλήρως per-file model, αναπαράγοντας ουσιαστικά για τα user data το μοντέλο του FileVault όταν αυτό δεν έχει ακόμη ενεργοποιηθεί πλήρως. Οι προγραμματιστές μπορούν, όπου απαιτείται, να επιλέξουν ισχυρότερες κλάσεις ή διαφορετικό τρόπο πρόσβασης για πιο ευαίσθητα δεδομένα. Η διαφοροποίηση αυτή δείχνει ότι το macOS προσπαθεί να ισορροπήσει ανάμεσα σε ομοιόμορφη προστασία, απόδοση και πρακτική διαχείριση κλειδιών για desktop workloads (Apple, 2024b; Apple, 2024c).

Παρά ταύτα, η κρυπτογράφηση volume και η Data Protection δεν προστατεύουν από κάθε μορφή απειλής. Αν ο νόμιμος χρήστης έχει ήδη ξεκλειδώσει το σύστημα και μια κακόβουλη εφαρμογή εκτελείται στο πλαίσιο του λογαριασμού του, το πρόβλημα μετατοπίζεται από την offline προστασία στον έλεγχο πρόσβασης, στην ασφάλεια εφαρμογών και στις πολιτικές ιδιωτικότητας. Η εικόνα 2 συνοψίζει αυτή τη διαστρωμάτωση, δείχνοντας ότι η εμπιστευτικότητα στο macOS εξαρτάται τόσο από τη σωστή κρυπτογραφική ιεραρχία όσο και από τα μηχανικά όρια πρόσβασης που επιβάλλονται πάνω από αυτήν (Apple, 2024d; Apple, 2026a).



Εικόνα 2 Ιεραρχία εμπιστευτικότητας, κρυπτογράφησης και κλειδιών στο macOS.

4.3 FileVault: αρχιτεκτονική, λειτουργία και όρια

Το FileVault αποτελεί τον κεντρικό μηχανισμό full-volume encryption του macOS και είναι το μέτρο που μετατρέπει την κρυπτογραφική δυνατότητα του APFS σε πραγματική προστασία δεδομένων at rest. Με την ενεργοποίησή του, τα δεδομένα του startup volume προστατεύονται με AES-XTS και η πρόσβαση στο κλειδί κρυπτογράφησης ενσωματώνεται σε μια ιεραρχία που συνδέεται με το συνθηματικό του χρήστη, με κλειδιά ανάκτησης και με hardware secrets. Έτσι, η πρόσβαση στα δεδομένα δεν αρκεί να γίνει στο επίπεδο του δίσκου, αλλά απαιτεί επιτυχή συμμετοχή στον μηχανισμό ξεκλειδώματος που ορίζει το ίδιο το λειτουργικό (Apple, 2026a).

Η επιχειρησιακή αξία του FileVault γίνεται ιδιαίτερα εμφανής στα σενάρια διαχείρισης συσκευών και εταιρικής ασφάλειας. Η Apple συνδέει το ξεκλείδωμα του volume με secure

token, bootstrap token και volume ownership, ώστε η διαχείριση του δικαιώματος αποκρυπτογράφησης να μη βασίζεται απλώς σε κάποιον λογαριασμό διαχειριστή, αλλά σε ελεγχόμενες σχέσεις εμπιστοσύνης μεταξύ λογαριασμών, MDM και πολιτικής συσκευής. Αυτό επιτρέπει τόσο τη θεσμική ανάκτηση πρόσβασης όσο και την αποφυγή ανεξέλεγκτης διάχυσης του δικαιώματος αποκρυπτογράφησης (Apple, 2024h; Apple, 2025a).

Το FileVault, ωστόσο, έχει σαφές όριο αποτελεσματικότητας: προστατεύει κυρίως από offline επιθέσεις, απώλεια ή κλοπή συσκευής και μη εξουσιοδοτημένη πρόσβαση πριν από το login. Μόλις το volume ξεκλειδωθεί και ο χρήστης εργαστεί κανονικά, η ασφάλεια εξαρτάται περισσότερο από το sandboxing, το TCC, τα entitlements, το SIP και τους μηχανισμούς ακεραιότητας του συστήματος. Συνεπώς, το FileVault είναι αναγκαίο αλλά όχι επαρκές στοιχείο της εμπιστευτικότητας στο macOS· λειτουργεί ως θεμέλιο, όχι ως πλήρης υποκατάστατη όλων των υπόλοιπων επιπέδων άμυνας (Apple, 2025a; Apple, 2026a).

4.4 Secure Enclave και hardware-based security

Το Secure Enclave εισάγει στο macOS μια hardware-based διάσταση ασφάλειας που αλλάζει ποιοτικά τον τρόπο με τον οποίο προστατεύονται τα κλειδιά. Πρόκειται για απομονωμένο υποσύστημα με δικό του Boot ROM, δικό του λειτουργικό περιβάλλον και δικό του μηχανισμό προστασίας μνήμης, το οποίο διαχειρίζεται ευαίσθητες κρυπτογραφικές λειτουργίες και συσχετίζει κρίσιμα μυστικά με το ίδιο το hardware της συσκευής. Αυτή η απομόνωση σημαίνει ότι ακόμη και ένα ιδιαίτερα ισχυρό compromise του main operating system δεν μεταφράζεται αυτομάτως σε δυνατότητα εξαγωγής των hardware-bound μυστικών (Apple, 2024e).

Η συμβολή του Secure Enclave στην εμπιστευτικότητα δεν έγκειται μόνο στη δημιουργία κλειδιών, αλλά και στην επιβολή αντι-rollback και OS-bound ιδιοτήτων. Μέσω μηχανισμών όπως το Sealed Key Protection, τα key encryption keys μπορούν να δεσμεύονται όχι μόνο με το UID της συσκευής αλλά και με μετρήσεις του λογισμικού και της ενεργής πολιτικής ασφάλειας. Στην πράξη αυτό σημαίνει ότι η απόπειρα εκκίνησης με χαλαρότερη πολιτική, απενεργοποιημένο secure boot ή αλλοιωμένο περιβάλλον δεν είναι απλώς θέμα ελέγχου εκκίνησης, αλλά επηρεάζει και τη δυνατότητα χρήσης των ίδιων των κλειδιών που προστατεύουν τα δεδομένα (Apple, 2021e; Apple, 2024e).

Παρά την ισχύ του μοντέλου αυτού, η ύπαρξη hardware root of trust δεν συνεπάγεται απόλυτη ασφάλεια. Η ερευνητική βιβλιογραφία για το Apple silicon έχει δείξει ότι πλευρικά κανάλια και χαμηλού επιπέδου μικροαρχιτεκτονικές αλληλεπιδράσεις μπορούν να δημιουργήσουν διαδρομές επίθεσης που δεν εξουδετερώνονται απλώς με την ύπαρξη ασφαλούς coprocessor. Επομένως, το Secure Enclave πρέπει να αξιολογείται ως κρίσιμος αλλά όχι μοναδικός πυλώνας εμπιστευτικότητας, ο οποίος ενισχύεται όταν συνδυάζεται με σωστή πολιτική εκκίνησης, ενημερώσεις και ελεγχόμενη πρόσβαση εφαρμογών στα προστατευμένα δεδομένα (Yu et al., 2023; Apple, 2024e).

4.5 Διαχείριση κλειδιών και προστασία δεδομένων χρήστη

Σε κάθε σύστημα κρυπτογράφησης, το καθοριστικό ερώτημα δεν είναι μόνο «ποιος αλγόριθμος χρησιμοποιείται», αλλά κυρίως «πώς παράγονται, πού αποθηκεύονται, ποιος εξουσιοδοτεί τη χρήση τους και πότε ανακαλούνται τα κλειδιά». Στο macOS, η διαχείριση κλειδιών διαπερνά όλες τις προηγούμενες υποενότητες: το FileVault προστατεύει volumes, η Data Protection επιβάλει κλάσεις για αρχεία, ενώ το Secure Enclave διασφαλίζει ότι κρίσιμο μυστικό υλικό παραμένει συνδεδεμένο με τη συσκευή και την ενεργή πολιτική ασφαλείας. Η

ωριμότητα της πλατφόρμας κρίνεται σε μεγάλο βαθμό από αυτήν ακριβώς τη συνολική key hierarchy (Barker, 2020; Apple, 2024e).

Στο επίπεδο εφαρμογών και λογαριασμών χρήστη, η Keychain και το securityd παίζουν καθοριστικό ρόλο. Η Keychain προσφέρει ασφαλή αποθήκευση μικρών αλλά εξαιρετικά ευαίσθητων αντικειμένων - passwords, login tokens, certificates, application secrets - και συνδυάζει κρυπτογραφική προστασία με ελέγχους που βασίζονται σε code signing, entitlements και ACLs. Αυτό σημαίνει ότι η πρόσβαση σε μυστικά δεν καθορίζεται μόνο από το filesystem path, αλλά και από το ποια εφαρμογή ζητά το αντικείμενο, σε ποιον λογαριασμό ανήκει και ποια trust relationship έχει δηλωθεί κατά τη δημιουργία του στοιχείου (Apple, 2024d).

Ο Πίνακας 1 συνοψίζει τη λογική αυτού του οικοσυστήματος εμπιστευτικότητας. Το συμπέρασμα είναι ότι το macOS δεν επιδιώκει μια απλή, ενιαία απάντηση σε κάθε απειλή, αλλά διαφορετικό μηχανισμό για διαφορετικό στάδιο της αλυσίδας επίθεσης: άλλο εργαλείο για απώλεια συσκευής, άλλο για app secrets, άλλο για αρχεία χρήστη και άλλο για hardware-bound κλειδιά. Αυτή η διαφοροποίηση αυξάνει την πολυπλοκότητα διαχείρισης, αλλά ταυτόχρονα μειώνει την πιθανότητα ενός και μόνο failure mode να καταρρεύσει ολόκληρη την εμπιστευτικότητα της πλατφόρμας (Apple, 2024b; Apple, 2024d; Apple, 2026d).

Πίνακας 1 Αντιστοίχιση μηχανισμών εμπιστευτικότητας, απειλών και βασικών ορίων στο macOS.

Μηχανισμός	Κύριος στόχος	Τύπος απειλής	Βασικό όριο
APFS + snapshots	Οργάνωση system/data state και ασφαλείς μεταβάσεις	Ασυνέπεια state, πρόχειρες τροποποιήσεις	Δεν αρκεί χωρίς κρυπτογράφηση και πολιτικές κλειδιών
FileVault	Προστασία ολόκληρου startup volume at rest	Απώλεια συσκευής, αφαίρεση δίσκου, offline ανάγνωση	Μετά το login, η προστασία εξαρτάται από άλλους μηχανισμούς
Secure Enclave / SKP	Hardware-bound μυστικό υλικό και anti-replay	Εξαγωγή κλειδιών, χαλάρωση πολιτικής εκκίνησης	Δεν εξαλείφει μικροαρχιτεκτονικούς ή side-channel κινδύνους
Keychain + securityd	Ασφαλής αποθήκευση passwords, tokens και certificates	Κατάχρηση application secrets, ανεξέλεγκτη πρόσβαση σε credentials	Η υπερβολική παραχώρηση trust ή entitlements αυξάνει το ρίσκο

5. Μηχανισμοί Προστασίας Εφαρμογών

Η ασφάλεια εφαρμογών αποτελεί ίσως το πιο εμφανές σημείο διαφοροποίησης του macOS σε σχέση με άλλα desktop λειτουργικά συστήματα. Η Apple έχει οικοδομήσει ένα μοντέλο εμπιστοσύνης που ξεκινά πριν από την πρώτη εκτέλεση του λογισμικού και συνεχίζεται σε όλη τη διάρκεια ζωής του, από την υπογραφή κώδικα και τη notarization έως το runtime confinement και τη μεταγενέστερη άμυνα απέναντι σε malware. Το κεφάλαιο αυτό εξετάζει πώς η συγκεκριμένη αλυσίδα μετατρέπει την έννοια της «ασφαλούς εφαρμογής» σε τεχνική πολιτική που εκτελείται από το ίδιο το λειτουργικό σύστημα (Apple, 2026b; Apple, 2024g).

5.1 Code Signing και αλυσίδα εμπιστοσύνης

Το code signing είναι η πρώτη ύλη της αλυσίδας εμπιστοσύνης εφαρμογών στο macOS. Μέσω ψηφιακής υπογραφής, το λειτουργικό σύστημα μπορεί να συνδέσει ένα binary με έναν δηλωμένο προγραμματιστή ή κανάλι διανομής και ταυτόχρονα να ελέγξει αν το περιεχόμενο του πακέτου έχει αλλοιωθεί μετά την παραγωγή του. Το σημαντικό στοιχείο δεν είναι απλώς η ταυτοποίηση του εκδότη, αλλά η δυνατότητα του συστήματος να απορρίψει κώδικα που δεν φέρει έγκυρη και συνεπή κρυπτογραφική σφραγίδα. Με τον τρόπο αυτό, η ασφάλεια εφαρμογών ξεκινά πριν ακόμη από την εκτέλεση, στο στάδιο της παραδοχής του ίδιου του κώδικα ως έγκυρου αντικειμένου του οικοσυστήματος (Apple, 2021a; Apple, 2026b).

Η σημασία του code signing επεκτείνεται και στο εσωτερικό της πολιτικής του macOS. Τα entitlements, οι δυνατότητες sandboxing, η σχέση με την Keychain και πολλές αποφάσεις του συστήματος πρόσβασης δεν βασίζονται απλώς στο όνομα ενός εκτελέσιμου αρχείου, αλλά στην υπογεγραμμένη ταυτότητά του. Αυτό επιτρέπει στη πλατφόρμα να συνδέει προνόμια και εξαιρέσεις με συγκεκριμένες, επαληθεύσιμες ταυτότητες λογισμικού και όχι με ευμετάβλητα χαρακτηριστικά του filesystem. Η υπογραφή, με άλλα λόγια, λειτουργεί και ως μηχανισμός ταυτοποίησης και ως άγκυρα πολιτικής (Apple, 2021a; Apple, 2026b).

Παρά τα πλεονεκτήματα αυτά, το code signing δεν αποδεικνύει ότι μια εφαρμογή είναι «καλή» ή άτρωτη· αποδεικνύει κυρίως ποιος τη παρήγαγε και ότι δεν έχει αλλοιωθεί έκτοτε. Ένα ευάλωτο, κακοσχεδιασμένο ή ακόμη και κακόβουλο πρόγραμμα μπορεί να είναι απολύτως ορθά υπογεγραμμένο. Συνεπώς, το code signing είναι απαραίτητο αλλά όχι επαρκές στοιχείο ασφαλείας. Η αποτελεσματικότητά του προκύπτει μόνο όταν συνδυάζεται με Gatekeeper, notarization, sandboxing και μεταγενέστερους μηχανισμούς απόκρισης (Apple, 2024f; Apple, 2024g).

5.2 Gatekeeper και πολιτικές εκτέλεσης λογισμικού

Ο Gatekeeper είναι ο βασικός μηχανισμός πολιτικής που ελέγχει εάν ένα λογισμικό θα μπορέσει να εκτελεστεί για πρώτη φορά στο macOS. Αξιοποιεί στοιχεία όπως το quarantine attribute, την υπογραφή κώδικα, τη σχέση με έγκυρο Developer ID ή App Store distribution και την ύπαρξη notarization ticket, ώστε να αποφασίζει αν το λογισμικό πληροί το ελάχιστο επίπεδο εμπιστοσύνης που απαιτείται για να φθάσει στον χρήστη. Η λογική του δεν είναι να εγγυηθεί απόλυτη καθαρότητα, αλλά να περιορίσει την αυθαίρετη εκτέλεση άγνωστου ή μη ιχνηλατήσιμου κώδικα (Apple, 2024f).

Σε αντίθεση με κλασικά permission models, ο Gatekeeper εισάγει έναν έλεγχο προέλευσης και ακεραιότητας πριν από το runtime. Έτσι, το ζήτημα δεν είναι μόνο τι θα επιτρέψει μια εφαρμογή να κάνει αφού εκτελεστεί, αλλά αν θα της επιτραπεί εξαρχής να ξεκινήσει υπό

φυσιολογικές συνθήκες χρήστη. Σε αυτό το επίπεδο φαίνεται και η στρατηγική της Apple να μεταφέρει τμήμα της άμυνας πιο κοντά στη διανομή λογισμικού, δηλαδή πριν το malware αποκτήσει έστω και μία φορά εκτελέσιμο foothold στο σύστημα (Apple, 2024f; Apple, 2026b).

Η εικόνα 3 συνοψίζει την αλληλουχία από την υπογραφή έως τον Gatekeeper και τη μεταγενέστερη επιβολή πολιτικής. Παρότι ο μηχανισμός είναι ισχυρός, παραμένει πολιτικό φίλτρο και όχι αδιαπέραστο τεχνικό τείχος. Ο χρήστης ή ο διαχειριστής μπορεί σε ορισμένα σενάρια να επιλέξει την εκτέλεση λογισμικού εκτός των προεπιλογών, ενώ περιβάλλοντα ανάπτυξης και εξειδικευμένων εργαλείων απαιτούν συχνά εξαιρέσεις. Αυτό σημαίνει ότι η αποτελεσματικότητα του Gatekeeper εξαρτάται και από την πειθαρχία χρήσης του οικοσυστήματος, όχι μόνο από την ίδια την τεχνική υλοποίηση (Apple, 2024f).

Εικόνα 3 Αλυσίδα εμπιστοσύνης εφαρμογών στο macOS από την υπογραφή έως τη μεταγενέστερη άμυνα.

5.3 Notarization και προστασία από κακόβουλες εφαρμογές

Η notarization προσθέτει ένα επιπλέον επίπεδο ελέγχου ανάμεσα στον προγραμματιστή και στον τελικό χρήστη. Η Apple απαιτεί από το λογισμικό που διανέμεται εκτός Mac App Store αλλά εντός του φυσιολογικού οικοσυστήματος εμπιστοσύνης να υποβληθεί σε automated έλεγχο και να λάβει αντίστοιχο ticket. Η διαδικασία αυτή δεν υποκαθιστά την υπογραφή κώδικα· την συμπληρώνει, εισάγοντας μια κεντρική υπηρεσία αξιολόγησης πριν από την πρώτη εκτέλεση στο σύστημα του χρήστη (Apple Developer, n.d.-c; Apple, 2024f).

Η προστασία από malware στο macOS, όμως, δεν σταματά στη notarization. Μηχανισμοί όπως το XProtect και οι μεταγενέστερες ενημερώσεις κανόνων ανίχνευσης επιτρέπουν στο λειτουργικό να αντιδρά και αφού ένα binary έχει ήδη διανεμηθεί, να ανακαλεί εμπιστοσύνη ή να μπλοκάρει γνωστές οικογένειες κακόβουλου λογισμικού. Έτσι, η Apple υιοθετεί ένα μοντέλο όπου η αποδοχή λογισμικού είναι διαρκώς αναθεωρήσιμη και όχι μονοσήμαντη απόφαση που λαμβάνεται άπαξ κατά την πρώτη εγκατάσταση (Apple, 2024g).

Το σημαντικό όριο αυτής της προσέγγισης είναι ότι η notarization και η ανίχνευση malware εξετάζουν κυρίως γνωστές ή μη αποδεκτές ιδιότητες ενός πακέτου, όχι την πλήρη

σημασιολογική ορθότητα ή την απουσία σφαλμάτων ασφαλείας. Ένα νόμιμο και notarized πρόγραμμα μπορεί να εξακολουθεί να περιέχει σοβαρές αδυναμίες ή να καταλήξει σε κατάχρηση εάν παραβιαστεί μεταγενέστερα. Η αξία της notarization βρίσκεται συνεπώς στη μείωση της επιφάνειας εισόδου και της ανώνυμης διανομής malware, όχι στην απόλυτη πιστοποίηση ποιότητας λογισμικού (Apple, 2024g; Apple Developer, n.d.-c).

5.4 Sandboxing (Seatbelt): αρχιτεκτονική και περιορισμοί

Το App Sandbox μεταφέρει την ασφάλεια εφαρμογών από το στάδιο της αποδοχής στο στάδιο της εκτέλεσης. Πρόκειται για τεχνολογία mandatory access control, επιβαλλόμενη από τον πυρήνα, η οποία περιορίζει μια εφαρμογή σε ελάχιστο σύνολο δυνατοτήτων που δηλώνονται μέσω entitlements και συνδυάζονται με containerized storage και αυστηρότερη μεσολάβηση πρόσβασης. Η λογική του δεν είναι να αποτρέψει την ύπαρξη bug, αλλά να περιορίσει δραστικά τις συνέπειες όταν το bug εκμεταλλευθεί επιτυχώς (Apple Developer, n.d.-a; Apple Developer, n.d.-d).

Το sandboxing είναι κρίσιμο ιδιαίτερα για εφαρμογές που επεξεργάζονται περιεχόμενο υψηλού ρίσκου - αρχεία από το διαδίκτυο, μηνύματα, previews ή δεδομένα που προέρχονται από μη αξιόπιστους χρήστες. Εάν μια τέτοια εφαρμογή συμβιβαστεί, το sandbox δυσκολεύει την πλευρική κίνηση προς τον υπόλοιπο λογαριασμό, την αυθαίρετη πρόσβαση σε αρχεία και την κατάχρηση συστημικών υπηρεσιών. Σε συνδυασμό με το TCC, δημιουργεί ένα δεύτερο στρώμα όπου ακόμη και μια έγκυρα υπογεγραμμένη εφαρμογή δεν αποκτά αυτόματα πρόσβαση σε ευαίσθητους πόρους χρήστη (Apple, 2021d; Apple, 2026b).

Η αποτελεσματικότητα του App Sandbox περιορίζεται από δύο πρακτικά δεδομένα. Πρώτον, δεν είναι όλες οι desktop εφαρμογές πλήρως sandboxed, ειδικά σε περιβάλλοντα όπου απαιτείται βαθιά ενσωμάτωση με το σύστημα ή legacy συμβατότητα. Δεύτερον, κάθε entitlement ή εξαίρεση αυξάνει το λειτουργικό αποτύπωμα και, κατ' επέκταση, την πιθανή επιφάνεια επίθεσης. Το sandbox, επομένως, είναι εξαιρετικά χρήσιμο όταν παραμένει στενό και πειθαρχημένο, αλλά αποδυναμώνεται όταν η εφαρμογή απαιτεί διαδοχικές διευρύνσεις των δικαιωμάτων της (Apple Developer, n.d.-d; Apple, 2021d).

5.5 Hardened Runtime

Το Hardened Runtime προσθέτει ένα σύνολο περιορισμών στο runtime περιβάλλον της εφαρμογής, με σκοπό να περιορίσει τεχνικές όπως arbitrary code injection, μη ελεγχόμενη φόρτωση βιβλιοθηκών και άλλες μορφές τροποποίησης της εκτέλεσης. Στην ουσία, είναι μια προσπάθεια να προστατευθεί όχι μόνο η προέλευση του κώδικα, αλλά και η ακεραιότητα του τρόπου με τον οποίο ο κώδικας αυτός εκτελείται στη μνήμη. Για λογισμικό που διανέμεται στο σύγχρονο οικοσύστημα του macOS, το Hardened Runtime αποτελεί πια βασικό συμπλήρωμα του code signing και της notarization (Apple Developer, n.d.-b; Apple, 2024f).

Η προσέγγιση αυτή είναι ιδιαίτερα σημαντική απέναντι σε επιθέσεις που δεν βασίζονται στην εγκατάσταση νέου μη υπογεγραμμένου binary, αλλά στη χειραγώγηση ήδη έγκυρης εφαρμογής. Το εάν μια εφαρμογή επιτρέπει JIT, dynamic code loading ή ειδικές εξαιρέσεις δεν είναι ουδέτερο ζήτημα λειτουργικότητας, αλλά μέρος του μοντέλου απειλής της. Το Hardened Runtime επιτρέπει στην Apple να μετατρέψει πολλές από αυτές τις δυνατότητες σε explicit exceptions, άρα και σε ευκολότερα ελεγχόμενο ρίσκο από όσο σε ένα εντελώς ανοικτό runtime περιβάλλον (Apple Developer, n.d.-b).

Και εδώ, όμως, εμφανίζεται η κλασική ένταση ανάμεσα σε χρηστικότητα και ασφάλεια. Ορισμένες κατηγορίες εφαρμογών - προγράμματα ανάπτυξης, browsers, virtualization και debugging εργαλεία - απαιτούν λειτουργίες που έρχονται σε τριβή με αυστηρούς runtime περιορισμούς. Αυτό σημαίνει ότι η πλατφόρμα χρειάζεται να προσφέρει ελεγχόμενες εξαιρέσεις, ενώ παράλληλα προσπαθεί να μην καταστήσει τις εξαιρέσεις αυτές το νέο κανονικό. Η αξιολόγηση του μοντέλου πρέπει συνεπώς να λαμβάνει υπόψη όχι μόνο την τεχνική ισχύ του, αλλά και την πειθαρχία με την οποία εφαρμόζεται στην πράξη (Apple, 2026b; Apple Developer, n.d.-b).

5.6 Πλεονεκτήματα και αδυναμίες του μοντέλου ασφάλειας εφαρμογών του macOS

Το συνολικό μοντέλο ασφάλειας εφαρμογών του macOS παρουσιάζει σημαντικά πλεονεκτήματα. Συνδυάζει έλεγχο προέλευσης, κεντρική πολιτική αποδοχής, runtime περιορισμούς και μεταγενέστερη ανάκληση εμπιστοσύνης, δημιουργώντας πολλαπλά σημεία παρεμβολής πριν και μετά την εκτέλεση λογισμικού. Για τον μη εξειδικευμένο χρήστη, αυτό μεταφράζεται σε ισχυρότερες ασφαλείς προεπιλογές: είναι δυσκολότερο να εκτελεστεί αυθαίρετος κώδικας χωρίς τριβή, και ακόμη δυσκολότερο να αποκτήσει άμεσα απεριόριστη πρόσβαση σε δεδομένα ή σε συστημικούς μηχανισμούς (Apple, 2024f; Apple, 2024g; Apple, 2026b).

Παράλληλα, το μοντέλο έχει εμφανείς αδυναμίες και περιορισμούς. Η ύπαρξη υπογραφής ή notarization δεν εξαλείφει τις λογικές ευπάθειες μιας εφαρμογής. Το sandbox δεν είναι καθολικό, ενώ η ανάγκη για legacy συμβατότητα, διαχειριστικά εργαλεία ή εξειδικευμένα επαγγελματικά workloads δημιουργεί συχνά εξαιρέσεις. Επιπλέον, η υψηλή εξάρτηση από τις υπηρεσίες εμπιστοσύνης της Apple σημαίνει ότι το οικοσύστημα συγκεντρώνει σημαντική δύναμη ορισμού του τι θεωρείται επιτρεπτό λογισμικό, γεγονός που ενισχύει την ασφάλεια αλλά μειώνει τη διαφάνεια και την αυτονομία σε ορισμένα σενάρια.

Συνολικά, η ασφάλεια εφαρμογών στο macOS είναι ισχυρή ακριβώς επειδή δεν αναθέτει όλο το βάρος σε έναν μόνο μηχανισμό. Το ουσιαστικό της πλεονέκτημα είναι η διαδοχή ελέγχων, από τη δημιουργία έως το runtime. Το ουσιαστικό της όριο είναι ότι παραμένει μέρος ενός ευρύτερου οικοσυστήματος εμπιστοσύνης, το οποίο απαιτεί τακτικές ενημερώσεις, ορθή διαχείριση εξαιρέσεων και επίγνωση ότι κάθε έγκυρα παραδεκτή εφαρμογή μπορεί ακόμη να αποτελέσει σημείο κινδύνου εάν το υπόλοιπο σύστημα πολιτικής χαλαρώσει ή παρακαμφθεί (Apple, 2024g; Apple, 2026b).

6. Ακεραιότητα Συστήματος και Προστασία Πόρων

Η ακεραιότητα του macOS δεν εξαρτάται από έναν μόνο έλεγχο υπογραφής, αλλά από μια αλυσίδα προστασίας που εκτείνεται από τη διαδικασία εκκίνησης μέχρι την πρόσβαση εφαρμογών σε κρίσιμους καταλόγους και δεδομένα χρήστη. Στο κεφάλαιο αυτό εξετάζονται οι μηχανισμοί που διασφαλίζουν ότι ο πυρήνας, το system volume, οι ενημερώσεις και οι προστατευμένοι πόροι δεν μεταβάλλονται εύκολα ακόμη και από κώδικα που κανονικά θα διέθετε αυξημένα προνόμια. Η ακεραιότητα στο macOS είναι, επομένως, υπόθεση hardware root of trust, verified boot, read-only περιεχομένου, policy enforcement και ελεγχόμενης ιδιωτικότητας (Apple, 2026c).

6.1 System Integrity Protection (SIP)

Το System Integrity Protection (SIP) αποτελεί έναν από τους πιο χαρακτηριστικούς μηχανισμούς με τους οποίους η Apple περιόρισε τη σχεδόν απόλυτη εξουσία του λογαριασμού root στο macOS. Ο βασικός του στόχος είναι να εμποδίσει μη εξουσιοδοτημένες μεταβολές σε κρίσιμους καταλόγους, αρχεία και διεπαφές του συστήματος, ακόμη και όταν μια διεργασία εκτελείται με υψηλά προνόμια. Με αυτόν τον τρόπο, το SIP αλλάζει τη θεωρητική υπόθεση ότι η κατάκτηση του root ισοδυναμεί αυτομάτως με ολοκληρωμένο έλεγχο του λειτουργικού συστήματος (Apple, 2021b).

Από άποψη ασφάλειας, το SIP λειτουργεί ως πολιτική ακεραιότητας τύπου MAC που προστατεύει την trusted base του macOS από αυθαίρετες εγγραφές και επεμβάσεις. Δεν ενδιαφέρεται να περιορίσει την πρόσβαση του χρήστη στα δικά του δεδομένα, αλλά να αποτρέψει τροποποιήσεις που θα αλλοίωναν το ίδιο το λειτουργικό περιβάλλον ή θα εισήγαγαν μόνιμο, stealthy compromise σε επίπεδο συστήματος. Σε συνδυασμό με το signed system volume, μετατρέπει την ακεραιότητα από απλή δυνατότητα ελέγχου σε καθημερινή, ενεργή πολιτική άρνησης μη εξουσιοδοτημένων αλλαγών (Apple, 2021b; Apple, 2022b).

Το SIP, ωστόσο, δεν είναι εντελώς αποκομμένο από τη λειτουργική πραγματικότητα. Υπάρχουν περιβάλλοντα συντήρησης, ανάπτυξης ή έρευνας όπου απαιτείται χαλάρωση της πολιτικής, ενώ η ίδια η Apple παρέχει διαδικασίες αλλαγής policy μέσω recoveryOS και startup security settings. Αυτό επιβεβαιώνει ότι το SIP είναι ισχυρό ακριβώς επειδή εντάσσεται σε μια συνολικότερη αλυσίδα ελέγχων: όταν η πολιτική χαλαρώνει, αυτό πρέπει να είναι συνειδητή, ελέγξιμη και εμφανής απόφαση και όχι παρενέργεια μιας απλής ανύψωσης προνομίων (Apple, 2021b; Apple, 2021c).

6.2 Read-only system volume και sealed system snapshots

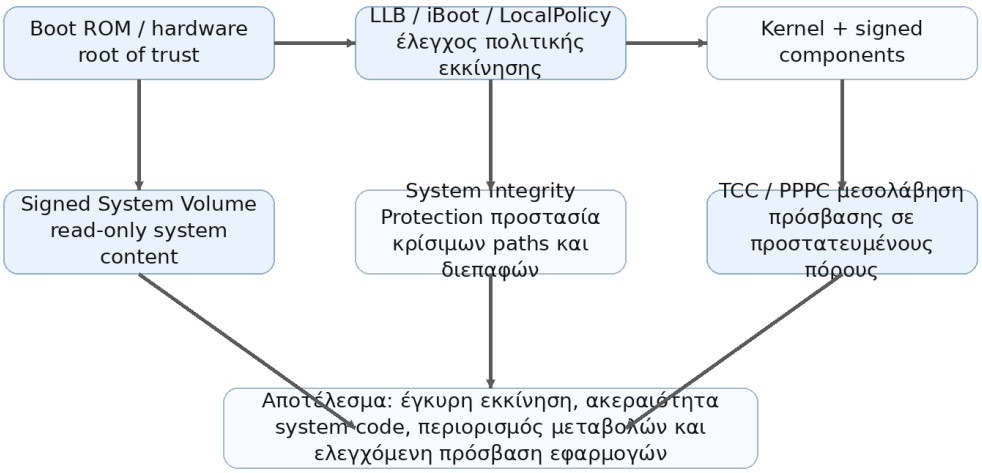
Η μετάβαση σε read-only system volume στο macOS 10.15 και στη συνέχεια σε signed system volume (SSV) στο macOS 11 σηματοδότησε μια σημαντική μετατόπιση από την προστασία «με κανόνες» στην προστασία «με δομικό σχεδιασμό». Το λειτουργικό σύστημα δεν βασίζεται πλέον μόνο στην απαγόρευση εγγραφής σε ορισμένα paths, αλλά διαχωρίζει θεμελιωδώς το system content από τα δεδομένα χρήστη και προσθέτει κρυπτογραφική σφράγιση πάνω από το περιεχόμενο του system volume. Έτσι, η ακεραιότητα δεν είναι απλώς θέμα άδειας, αλλά θέμα επαληθεύσιμης γνησιότητας του ίδιου του περιεχομένου (Apple, 2022b).

Το signed system volume χρησιμοποιεί μηχανισμό που επιτρέπει στον πυρήνα να επαληθεύει την ακεραιότητα του system content και να απορρίπτει οποιοδήποτε code ή non-code data δεν αντιστοιχεί σε έγκυρη υπογραφή της Apple. Η συνέπεια αυτής της επιλογής είναι σημαντική: η

αλλοίωση του συστήματος γίνεται σαφώς δυσκολότερη όχι μόνο επειδή προστατεύονται επιλεγμένα αρχεία, αλλά επειδή προστατεύεται το συνολικό, σφραγισμένο state του system volume. Επιπλέον, η χρήση snapshots ευνοεί ασφαλέστερες ενημερώσεις, rollback behavior και καθαρότερη μετάβαση μεταξύ system states (Apple, 2022b; Apple, 2026c).

Βέβαια, το SSV δεν λύνει κάθε ζήτημα ακεραιότητας στο macOS. Προστατεύει το system volume, όχι όλα τα user data, τις ρυθμίσεις τρίτων εργαλείων ή κάθε πιθανό persistence mechanism εκτός των προστατευμένων περιοχών. Για τον λόγο αυτό χρειάζεται να διαβάζεται σε συνάρτηση με το SIP, το secure boot και το TCC. Η εικόνα 4 δείχνει ακριβώς αυτήν τη λογική: η ακεραιότητα του συστήματος είναι ισχυρή όταν οι επιμέρους μηχανισμοί αλληλοενισχύονται, όχι όταν απομονώνονται αναλυτικά ο ένας από τον άλλον (Apple, 2022b; Apple, 2026c).

Αλυσίδα ακεραιότητας από την εκκίνηση έως το runtime



Εικόνα 4 Αλυσίδα ακεραιότητας από την εκκίνηση έως το runtime στο macOS.

6.3 Διαχείριση ενημερώσεων και ασφάλεια firmware

Η ακεραιότητα στο macOS ξεκινά πριν από τη φόρτωση του kernel, μέσα από την αλυσίδα secure boot των συστημάτων με Apple silicon. Το Boot ROM, το iBoot, οι μετρήσεις λογισμικού, η LocalPolicy και οι υπόλοιπες συνιστώσες της διαδικασίας εκκίνησης λειτουργούν ως σειρά ελέγχων που διασφαλίζουν ότι το σύστημα ξεκινά από αναμενόμενη, υπογεγραμμένη και πολιτικά αποδεκτή κατάσταση. Αν το σημείο εκκίνησης υπονομευθεί, τότε κάθε μεταγενέστερη runtime προστασία χτίζεται πάνω σε ασταθές υπόβαθρο· γι' αυτό η Apple τοποθετεί το hardware root of trust στο κέντρο της αλυσίδας ακεραιότητας (Apple, 2021c; Apple, 2026c).

Η ίδια λογική συνεχίζεται στις ενημερώσεις λογισμικού και firmware. Η ασφαλής ενημέρωση δεν είναι απλώς μηχανισμός συντήρησης, αλλά βασική υπηρεσία ακεραιότητας, επειδή επιτρέπει την εγκατάσταση νέου, έγκυρου state χωρίς να ανοίγει εύκολα τον δρόμο σε rollback ή σε μη εξουσιοδοτημένη αντικατάσταση images. Στα συστήματα της Apple, οι ενημερώσεις και οι αλλαγές πολιτικής συνδέονται με trusted διαδικασίες, ενώ η έννοια της ιδιοκτησίας volume και των bootstrap tokens επηρεάζει και το ποιος έχει θεσμικά τη δυνατότητα να εξουσιοδοτεί κρίσιμες μεταβολές (Apple, 2024h; Apple, 2026c).

Από επιχειρησιακή άποψη, αυτή η προσέγγιση ωφελεί ιδιαίτερα φορητές και διαχειριζόμενες συσκευές, όπου η κλοπή, η πρόσβαση σε recovery περιβάλλον ή η απόπειρα εκκίνησης από εναλλακτικό μέσο αποτελούν πραγματικά σενάρια απειλής. Το μειονέκτημα είναι ότι η ασφάλεια εκκίνησης και firmware γίνεται στενότερα δεμένη με το hardware και με τα επίσημα κανάλια πολιτικής, μειώνοντας την ευκολία για custom boot flows ή ανεπίσημα σενάρια συντήρησης. Στο macOS, όμως, αυτή η μείωση ελευθερίας είναι συνειδητό αντάλλαγμα για ισχυρότερη ακεραιότητα θεμελίωσης (Apple, 2021c; Apple, 2024h).

6.4 Transparency, Consent and Control (TCC) framework

Το Transparency, Consent and Control (TCC) συχνά παρουσιάζεται ως μηχανισμός ιδιωτικότητας, αλλά στην πράξη αποτελεί και μηχανισμό ακεραιότητας πρόσβασης σε προστατευμένους πόρους. Μέσω του TCC, το macOS μεσολαβεί όταν εφαρμογές επιχειρούν να προσπελάσουν κατηγορίες δεδομένων ή τοποθεσιών όπως Desktop, Documents, Downloads, κάμερα, μικρόφωνο, φωτογραφίες ή άλλες προστατευμένες πηγές. Αυτό σημαίνει ότι ακόμη και μια εφαρμογή που έχει ήδη εκτελεστεί με επιτυχία δεν θεωρείται αυτομάτως αξιόπιστη για κάθε μορφή πρόσβασης σε user data (Apple, 2021d).

Στα managed περιβάλλοντα, το TCC δεν είναι απλώς διάλογος συγκατάθεσης προς τον τελικό χρήστη. Η Apple επιτρέπει την προρύθμιση σημαντικών αποφάσεων μέσω Privacy Preferences Policy Control (PPPC) payloads, κάτι που είναι ιδιαίτερα κρίσιμο για εργαλεία backup, EDR, remote support ή άλλες εφαρμογές που χρειάζονται επαναλαμβανόμενη πρόσβαση σε προστατευμένους πόρους. Η τεκμηρίωση της Apple αναφέρει ρητά ότι, όταν υπάρχουν πολλαπλές σχετικές πολιτικές, το λειτουργικό επιλέγει την πιο περιοριστική, γεγονός που δείχνει ότι το TCC δεν είναι απλώς UI layer αλλά κανονικό enforcement plane με σαφή προτεραιότητα προς την άρνηση (Apple, 2024i).

Η αξία του TCC ως μηχανισμού ακεραιότητας έγκειται στο ότι αποτρέπει τη «σιωπηλή» μετάβαση από ένα compromise εφαρμογής σε πλήρη πρόσβαση στο σύνολο του προφίλ χρήστη. Παρά ταύτα, δεν παύει να εξαρτάται από σωστή διαχείριση εξαιρέσεων και από τη συμπεριφορά του χρήστη απέναντι στα prompts. Εάν μια κακόβουλη ή υπερπρονομιούχα εφαρμογή λάβει Full Disk Access ή άλλες ευρείες εξαιρέσεις, μεγάλο μέρος του προστατευτικού οφέλους αποδυναμώνεται. Συνεπώς, το TCC είναι ισχυρό ως default-deny φράγμα, αλλά εύθραυστο όταν οι εξαιρέσεις γίνουν υπερβολικά γενναιόδωρες (Apple, 2021d; Apple, 2024i).

6.5 Ιδιωτικότητα έναντι λειτουργικότητας

Η αυστηρή προστασία πόρων στο macOS δημιουργεί αναπόφευκτα τριβές με τη λειτουργικότητα. Εφαρμογές backup, antivirus, EDR, indexing, remote administration και development tooling χρειάζονται συχνά πρόσβαση σε καταλόγους ή υπηρεσίες που το σύστημα θεωρεί ευαίσθητες. Όσο ισχυρότερη είναι η προεπιλεγμένη άρνηση, τόσο πιο συχνά εμφανίζονται prompts, entitlements ή ανάγκη για διαχειριστικές εξαιρέσεις. Η ένταση αυτή δεν είναι ατύχημα, αλλά θεμελιώδες χαρακτηριστικό κάθε συστήματος που προσπαθεί να προστατεύσει user data από λογισμικό που εκτελείται νόμιμα αλλά δεν είναι αυτονόητα αξιόπιστο (Apple, 2021d; Apple, 2024i).

Σε οργανωμένα περιβάλλοντα, το πρόβλημα αντιμετωπίζεται με MDM, PPPC profiles και κεντρική επιβολή πολιτικής. Έτσι μειώνεται το friction για τον τελικό χρήστη, αλλά μεταφέρεται μεγάλο μέρος της ευθύνης στον οργανισμό που ορίζει τις εξαιρέσεις. Η παραχώρηση Full Disk Access ή ευρειών προσβάσεων σε συστημικά εργαλεία διευκολύνει τις

επιχειρησιακές ανάγκες, όμως διευρύνει σημαντικά και το πιθανό blast radius σε περίπτωση compromise αυτών των εργαλείων. Η ισορροπία, επομένως, δεν επιτυγχάνεται με πλήρη χαλάρωση, αλλά με προσεκτική, ελάχιστη και ελέγξιμη παραχώρηση δικαιωμάτων.

Η περίπτωση του macOS δείχνει ότι η ιδιωτικότητα και η λειτουργικότητα δεν είναι αντίπαλες αρχές, αλλά μεταβλητές που πρέπει να συντονίζονται. Η πλατφόρμα επιλέγει σαφώς να θυσιάσει μέρος της ευχρηστίας προς όφελος της ελεγχόμενης πρόσβασης, ιδίως για μη ειδικούς χρήστες. Για power users και διαχειριστές, όμως, η ίδια επιλογή συχνά βιώνεται ως περιορισμός. Η ακαδημαϊκή αξιολόγηση οφείλει συνεπώς να αναγνωρίσει ότι η «καλή ασφάλεια» στο desktop δεν κρίνεται μόνο από την ισχύ των μηχανισμών, αλλά και από το πόσο βιώσιμη είναι η χρήση τους χωρίς να ωθούν τους χρήστες σε διαρκείς παρακάμψεις (Apple, 2024i).

6.6 Αξιολόγηση της αποτελεσματικότητας των μηχανισμών

Συνολικά, το macOS εμφανίζει ισχυρό μοντέλο ακεραιότητας επειδή ενοποιεί hardware root of trust, secure boot, signed system content, runtime policy και μεσολαβημένη πρόσβαση σε πόρους. Η δύναμη του μοντέλου δεν βρίσκεται σε έναν μόνο μηχανισμό, αλλά στον τρόπο που κάθε στρώμα καλύπτει την αστοχία του προηγούμενου: αν αποτύχει ο έλεγχος origin εφαρμογής, υπάρχει ακόμη sandbox και TCC· αν μια διεργασία αποκτήσει υψηλά προνόμια, συναντά SIP και read-only system content· αν επιχειρηθεί offline ή policy-level αλλοίωση, παρεμβαίνει η αλυσίδα secure boot (Apple, 2026c).

Τα όρια του μοντέλου, ωστόσο, παραμένουν υπαρκτά. Η ασφάλεια μπορεί να αποδυναμωθεί όταν ο χρήστης ή ο διαχειριστής χαλαρώσει συνειδητά τις πολιτικές, όταν εμφανιστούν σοβαρές ευπάθειες kernel ή όταν η απειλή μετακινηθεί σε μικροαρχιτεκτονικά ή πλευρικά κανάλια που δεν αντιμετωπίζονται άμεσα από μηχανισμούς policy enforcement. Η ύπαρξη έρευνας σε side channels για Apple silicon υπενθυμίζει ότι ακόμη και η ισχυρή ακεραιότητα λογισμικού δεν συνεπάγεται καθολική απουσία ρίσκου (Yu et al., 2023).

Παρά τις επιφυλάξεις αυτές, η συνολική εικόνα παραμένει θετική: το macOS έχει οικοδομήσει μια αλυσίδα ακεραιότητας σαφώς ισχυρότερη από εκείνη που ιστορικά χαρακτήριζε τα desktop λειτουργικά συστήματα. Το κρίσιμο εύρημα είναι ότι η ακεραιότητα στο macOS δεν είναι «χαρακτηριστικό», αλλά αρχιτεκτονική επιλογή που οργανώνει την ίδια τη σχέση του χρήστη, του root, του hardware και των εφαρμογών με το σύστημα. Αυτή η επιλογή αποτελεί έναν από τους κύριους λόγους για τους οποίους η πλατφόρμα παρουσιάζει ξεχωριστό ακαδημαϊκό ενδιαφέρον (Apple, 2026c; Apple, 2026d).

7. Συγκριτική Θεώρηση και Κριτική Αξιολόγηση

Μετά την ανάλυση των επιμέρους υπηρεσιών ασφάλειας στο macOS, το επόμενο βήμα είναι η συγκριτική και κριτική αποτίμηση. Το ερώτημα δεν είναι απλώς εάν το macOS διαθέτει πολλούς μηχανισμούς, αλλά πώς τοποθετείται απέναντι σε άλλα ώριμα desktop οικοσυστήματα, ποια πλεονεκτήματα του προσφέρει η καθετοποίηση υλικού και λογισμικού, ποιο είναι το κόστος αυτής της επιλογής και πού εντοπίζονται τα όρια του μοντέλου του. Το παρόν κεφάλαιο εξετάζει τις διαστάσεις αυτές σε σχέση κυρίως με τα Windows και το Ubuntu/Linux (Microsoft, 2025b; Ubuntu, 2025a).

7.1 Σύγκριση macOS με Windows και Linux σε επίπεδο ασφάλειας

Το macOS, τα Windows και το Ubuntu/Linux υιοθετούν διαφορετικές φιλοσοφίες ασφαλείας. Το macOS βασίζεται σε ισχυρή καθετοποίηση και σε σχετικά ομοιόμορφες προεπιλογές, τα Windows επιχειρούν να προστατεύσουν ένα πολύ πιο ανοικτό και ετερογενές οικοσύστημα εφαρμογών και hardware, ενώ το Ubuntu/Linux δίνει μεγαλύτερη έμφαση στη συνθετότητα μέσω πολιτικών του kernel, στη διαχειριστική ελευθερία και στη διαφορά ανά διανομή. Αυτή η θεμελιώδης διαφορά φιλοσοφίας εξηγεί γιατί παρόμοιοι μηχανισμοί - εφαρμοστικός έλεγχος, secure boot, MAC policies - παράγουν συχνά διαφορετική εμπειρία ασφάλειας στην πράξη (Microsoft, 2025b; Ubuntu, 2025a).

Στο επίπεδο της εκτέλεσης εφαρμογών, το macOS διαθέτει Gatekeeper, notarization και υποχρεωτική σύνδεση πολλών λειτουργιών με code signing. Τα Windows προσφέρουν ισχυρά εργαλεία application control, όπως το App Control for Business και σχετική πολιτική allowlisting, αλλά η αξιοποίησή τους εξαρτάται περισσότερο από το εκάστοτε επίπεδο διαχείρισης και από την εταιρική πολιτική. Το Ubuntu/Linux στηρίζεται περισσότερο σε repositories, package signing και σε μηχανισμούς confinement όπως το AppArmor, οι οποίοι είναι ισχυροί αλλά συχνά λιγότερο ενιαίοι ως προς την εμπειρία του τελικού desktop χρήστη (Microsoft, 2025a; Ubuntu, 2026a).

Στο επίπεδο της ακεραιότητας και της εκκίνησης, και τα τρία οικοσυστήματα διαθέτουν πλέον σοβαρά μέσα προστασίας. Τα Windows 11 συνδέουν Secure Boot και Trusted Boot με ευρύτερη αλυσίδα προστασίας συστήματος, ενώ το Ubuntu διαθέτει UEFI Secure Boot, kernel lockdown και ισχυρά storage features, όπως full disk encryption και επιλογές integrity verification. Ο Πίνακας 2 συνοψίζει τη σύγκριση: το macOS υπερέχει σε συνοχή και ασφαλείς προεπιλογές, τα Windows σε κλίμακα enterprise tooling και policy breadth, ενώ το Ubuntu/Linux σε ευελιξία και ορατότητα των μηχανισμών στον διαχειριστή (Microsoft, 2025b; Ubuntu, 2026b; Ubuntu, 2025a).

Πίνακας 2 Συγκριτική αποτίμηση βασικών αξόνων ασφάλειας σε macOS, Windows 11 και Ubuntu/Linux.

Άξονας	macOS	Windows 11	Ubuntu/Linux
Είσοδος εφαρμογών	Gatekeeper, notarization, ισχυρή έμφαση σε code signing	App Control for Business, SmartScreen, μεγάλη εξάρτηση από policy deployment	Repositories, package signing, ποικιλία ανά διανομή και εργαλείο
Runtime περιορισμοί	Sandbox, entitlements, Hardened Runtime, TCC	Sandboxing και application control με μεγαλύτερη ποικιλία enterprise profiles	AppArmor/SELinux ανά οικοσύστημα, συχνά πιο χειροκίνητη παραμετροποίηση
Εμπιστευτικότητα δεδομένων	FileVault, Keychain, Secure Enclave, hardware-bound keys	BitLocker, credential protections, TPM-centric enterprise model	LUKS/FDE, fscrypt, TPM και ποικίλες λύσεις ανά διανομή
Ακεραιότητα εκκίνησης	Boot ROM, iBoot, LocalPolicy, SSV, SIP	Secure Boot + Trusted Boot, ισχυρή υποστήριξη enterprise attestation	UEFI Secure Boot, kernel lockdown, dm-verity / fs-verity ανά χρήση
Διαχειριστική φιλοσοφία	Ισχυρές ασφαλείς προεπιλογές και μικρότερη ελευθερία	Μεγάλο policy breadth για enterprise οργανισμούς	Υψηλή ευελιξία, αλλά περισσότερη ευθύνη στον διαχειριστή

7.2 Πλεονεκτήματα του κλειστού οικοσυστήματος

Το βασικό πλεονέκτημα του κλειστού οικοσυστήματος της Apple είναι η ισχυρή προέλευση και η συνεκτικότητα. Όταν ο ίδιος κατασκευαστής ελέγχει το υλικό, το firmware, τον boot chain, το λειτουργικό σύστημα, τα κύρια κανάλια υπογραφής και μεγάλο μέρος του distribution model, η μετάφραση μιας πολιτικής ασφαλείας σε πρακτική υλοποίηση γίνεται ευκολότερη και πιο ομοιόμορφη. Αυτό σημαίνει λιγότερη κατακερματισμένη συμπεριφορά, μικρότερη απόκλιση μεταξύ θεωρίας και προεπιλεγμένης πρακτικής και, συχνά, ταχύτερη διάθεση βελτιώσεων σε όλο το στόλο συσκευών (Apple, 2026d).

Η συνοχή αυτή ευνοεί ιδιαίτερα τους μη εξειδικευμένους χρήστες. Η ασφάλεια δεν απαιτεί να ενεργοποιηθούν δεκάδες πρόσθετες πολιτικές ή να συντεθεί χειροκίνητα ένα μοντέλο εμπιστοσύνης· μεγάλο μέρος του οικοδομείται από προεπιλογή. Αυτό είναι κρίσιμο για καθημερινά desktop περιβάλλοντα, όπου τα συνηθέστερα compromise σενάρια δεν αφορούν ειδικευμένους αντιπάλους αλλά απροσεξία χρήστη, phishing, αμφίβολες λήψεις και κακή διαχείριση προνομίων. Σε αυτά τα σενάρια, η Apple ωφελείται από τη δυνατότητα να διαμορφώνει ενιαία την «κανονική» ασφαλή συμπεριφορά του συστήματος (Apple, 2024f; Apple, 2026b).

Επιπλέον, το κλειστό οικοσύστημα επιτρέπει στενότερη σύνδεση hardware και πολιτικής. Το Secure Enclave, το LocalPolicy, το FileVault και το signed system volume δεν είναι απομονωμένα features· είναι δυνατά ακριβώς επειδή η Apple μπορεί να θεωρεί δεδομένο ένα συγκεκριμένο μοντέλο εκκίνησης, υπογραφής και διαχείρισης συσκευής. Αυτό δεν σημαίνει ότι τα ανοικτά οικοσυστήματα δεν μπορούν να υλοποιήσουν ισχυρή ασφάλεια, αλλά ότι χρειάζονται

περισσότερο συντονισμό, περισσότερη διαχειριστική ικανότητα και, συνήθως, μεγαλύτερη ανοχή στην ετερογένεια (Apple, 2024e; Apple, 2026c; Ubuntu, 2025a).

7.3 Κριτική θεώρηση: περιορισμοί, bypasses και ερευνητικά ευρήματα

Η ισχύς του μοντέλου της Apple δεν πρέπει να συγχέεται με απουσία ορίων. Η ερευνητική βιβλιογραφία έχει δείξει ότι ακόμη και σε σύγχρονα Apple silicon systems μπορούν να εμφανιστούν μικροαρχιτεκτονικά πλευρικά κανάλια, τα οποία παρακάμπτουν εν μέρει το αφηγηματικό σχήμα της απόλυτα ασφαλούς πλατφόρμας. Η ύπαρξη τέτοιων επιθέσεων δεν ακυρώνει το συνολικό επίπεδο προστασίας, αλλά υπενθυμίζει ότι η ασφάλεια είναι πολυδιάστατη και ότι μια πλατφόρμα μπορεί να υπερέχει σε policy enforcement και παρ' όλα αυτά να παραμένει εκτεθειμένη σε χαμηλού επιπέδου τεχνικές (Yu et al., 2023).

Παράλληλα, το κλειστό οικοσύστημα συγκεντρώνει μεγάλο όγκο εμπιστοσύνης σε έναν φορέα. Η Apple ορίζει τα βασικά κανάλια υπογραφής, τις πολιτικές αποδοχής λογισμικού, τις ενημερώσεις, τα entitlements και πολλές από τις επιτρεπτές εξαιρέσεις. Αυτή η συγκέντρωση εξουσίας ενισχύει τη συνοχή και τη διαχειρισιμότητα, αλλά μειώνει την εξωτερική auditability και περιορίζει τη δυνατότητα του χρήστη ή του οργανισμού να αναδιαμορφώσει σε βάθος το μοντέλο ασφαλείας σύμφωνα με δικές του προτεραιότητες. Για ορισμένα περιβάλλοντα υψηλής διαφάνειας, αυτό συνιστά πραγματικό μειονέκτημα.

Η κριτική αποτίμηση πρέπει συνεπώς να είναι διπλή. Από τη μία, το macOS επιτυγχάνει υψηλό βαθμό προληπτικής προστασίας για τυπικά desktop σενάρια. Από την άλλη, η ασφάλεια του μοντέλου εξαρτάται σε μεγάλο βαθμό από τη διαρκή ορθότητα, ταχύτητα και κανονιστική επιλογή του ίδιου του κατασκευαστή. Με απλά λόγια, το οικοσύστημα ελαχιστοποιεί πολλές κλασικές επιφάνειες επίθεσης, αλλά μεταφέρει βαρύ φορτίο εμπιστοσύνης σε λιγότερα, πιο κεντρικά σημεία αποτυχίας ή διακυβέρνησης (Apple, 2024g; Yu et al., 2023).

7.4 Η ισορροπία μεταξύ usability και security

Η ισορροπία μεταξύ usability και security παραμένει ένα από τα πιο δύσκολα ζητήματα σε κάθε desktop λειτουργικό σύστημα. Στο macOS, η ασφάλεια εφαρμογών και η προστασία πόρων συχνά εμφανίζονται στον τελικό χρήστη ως prompts, blocks, notarization checks, αιτήματα entitlements ή ανάγκη πρόσθετων εγκρίσεων. Αυτές οι τριβές μπορεί να είναι χρήσιμες όταν αποτρέπουν μη αναμενόμενη πρόσβαση, αλλά μπορούν να γίνουν κουραστικές ή αδιαφανείς όταν ο χρήστης δεν κατανοεί το πλαίσιο της απόφασης που του ζητείται να πάρει (Apple, 2021d; Apple, 2024f).

Σε πιο ανοικτά οικοσυστήματα, όπως το Ubuntu/Linux, η λειτουργικότητα συχνά κερδίζει από τη μεγαλύτερη ελευθερία εγκατάστασης, παραμετροποίησης και σύνδεσης εργαλείων με το σύστημα, αλλά το τίμημα είναι ότι περισσότερη ευθύνη μεταφέρεται στον διαχειριστή ή στον έμπειρο χρήστη. Τα Windows κινούνται κάπου ενδιάμεσα, διαθέτοντας ισχυρά enterprise-oriented controls που όμως συχνά απαιτούν ώριμη πολιτική deployment για να αξιοποιηθούν πλήρως. Το macOS επιλέγει σαφέστερα τη λογική των ασφαλών προεπιλογών, ακόμη κι αν αυτό σημαίνει περισσότερη friction για ορισμένες κατηγορίες workload (Microsoft, 2025a; Ubuntu, 2025a).

Το κρίσιμο σημείο είναι ότι η χρηστικότητα δεν είναι αντίπαλος της ασφάλειας, αλλά προϋπόθεση βιωσιμότητάς της. Εάν ένα σύστημα είναι τόσο αυστηρό ώστε οι χρήστες να

αναζητούν συνεχώς τρόπους παράκαμψης, η θεωρητική ισχύς των μηχανισμών του αποδυναμώνεται δραστικά. Η επιτυχία του macOS βρίσκεται στο ότι, για πολλούς κοινούς χρήστες, οι ασφαλείς προεπιλογές παραμένουν σχετικά αόρατες. Το αδύναμο σημείο του είναι ότι για πιο σύνθετα ή επαγγελματικά περιβάλλοντα η ίδια αορατότητα συχνά μετατρέπεται σε αδιαφάνεια και λειτουργική τριβή.

7.5 Μελλοντικές κατευθύνσεις εξέλιξης της ασφάλειας στο macOS

Οι μελλοντικές κατευθύνσεις της ασφάλειας στο macOS φαίνεται ότι θα κινηθούν προς ακόμη στενότερη σύνδεση hardware, πολιτικής και διαχείρισης. Η γενική τάση της Apple δείχνει μεγαλύτερη εξάρτηση από hardware roots of trust, περισσότερο declarative device management, συχνότερες background security βελτιώσεις και ακόμη πιο λεπτομερή μεσολάβηση πρόσβασης σε κρίσιμους πόρους. Η πορεία αυτή είναι συνεπής με τη φιλοσοφία της πλατφόρμας: περισσότερη ασφάλεια μέσω ολοένα πιο ελεγχόμενων διαδρομών εκτέλεσης και διαχείρισης (Apple, 2026d; Apple, 2026c).

Παράλληλα, η αύξηση της πολυπλοκότητας των εφαρμογών, των επεκτάσεων χρήστη, των cloud διαπιστευτηρίων και της διαχείρισης εταιρικών στόλων θα διατηρήσει στο προσκήνιο τα ερωτήματα για το ποιο επίπεδο εξαιρέσεων είναι αποδεκτό και πώς θα αποφεύγεται η υπερσυσσώρευση προνομίων. Η μελλοντική εξέλιξη της ασφάλειας του macOS πιθανότατα δεν θα είναι απλώς προσθήκη νέων μηχανισμών, αλλά βελτίωση του τρόπου με τον οποίο τα υπάρχοντα στρώματα συνεργάζονται, εξηγούνται στους χρήστες και παραμετροποιούνται με μικρότερο λειτουργικό κόστος.

Από ερευνητική άποψη, παραμένουν ανοικτά πεδία όπως η εμπειρική μέτρηση της αποτελεσματικότητας των prompts ιδιωτικότητας, η αξιολόγηση της ασφάλειας των εξαιρέσεων entitlements σε πραγματικά οικοσυστήματα λογισμικού και η σχέση μικροαρχιτεκτονικών επιθέσεων με hardware-assisted μηχανισμούς προστασίας. Επομένως, η εξέλιξη του macOS πιθανόν θα συνεχίσει να είναι τεχνολογικά εντυπωσιακή, αλλά και γόνιμο αντικείμενο ακαδημαϊκής αμφισβήτησης και αξιολόγησης.

8. Συμπεράσματα

Το παρόν κεφάλαιο συνοψίζει τα βασικά ευρήματα της εργασίας και απαντά στα αρχικά ερευνητικά ερωτήματα. Ο στόχος δεν είναι απλώς μια περιγραφική επανάληψη των μηχανισμών που αναλύθηκαν, αλλά η αποτίμηση της συνολικής αρχιτεκτονικής λογικής του macOS ως συστήματος ασφάλειας και η ανάδειξη των σημείων στα οποία η πλατφόρμα επιτυγχάνει ουσιαστική διαφοροποίηση από άλλα desktop λειτουργικά συστήματα.

8.1 Ανακεφαλαίωση βασικών ευρημάτων

Το βασικό εύρημα της εργασίας είναι ότι η ασφάλεια του macOS δεν μπορεί να κατανοηθεί ως άθροισμα ανεξάρτητων features. Η αρχιτεκτονική του Darwin και του XNU, οι πολιτικές ελέγχου πρόσβασης, η τμηματοποίηση system και data state μέσω APFS, η κρυπτογράφηση FileVault, η Keychain, το Secure Enclave, το code signing, ο Gatekeeper, η notarization, το sandboxing, το SIP, το signed system volume και το TCC συγκροτούν μια ενιαία λογική defense-in-depth. Το καθένα απαντά σε διαφορετικό σημείο της αλυσίδας επίθεσης και η αποτελεσματικότητα του συστήματος προκύπτει από αυτήν ακριβώς τη συνδυαστική λειτουργία.

Δεύτερο σημαντικό εύρημα είναι ότι το macOS επιτυγχάνει υψηλό επίπεδο προστασίας επειδή μεταφέρει κρίσιμες αποφάσεις εμπιστοσύνης πιο κοντά στο hardware, στην εκκίνηση και στη διανομή λογισμικού. Η πλατφόρμα δεν περιμένει να εντοπίσει μόνο την κακόβουλη συμπεριφορά μετά την εμφάνισή της, αλλά προσπαθεί να περιορίσει τι μπορεί να ξεκινήσει, τι μπορεί να αλλάξει, ποιος μπορεί να αποκτήσει κλειδιά και ποια εφαρμογή μπορεί να αγγίξει συγκεκριμένα δεδομένα. Αυτή η φιλοσοφία προληπτικού περιορισμού είναι ίσως το πιο χαρακτηριστικό γνώρισμα της ασφάλειας του macOS.

8.2 Απάντηση στα ερευνητικά ερωτήματα

Ως προς το πρώτο ερευνητικό ερώτημα, η εργασία έδειξε ότι οι κλασικές αρχές ασφάλειας λειτουργικών συστημάτων - least privilege, separation of duties, MAC, defense-in-depth - μεταφράζονται στο macOS σε αρκετά συνεκτικούς μηχανισμούς. Η αρχιτεκτονική του συστήματος, η διαχείριση μνήμης και οι πολιτικές προνομίων δεν αποτελούν ουδέτερο υπόστρωμα, αλλά ενεργά στρώματα προστασίας. Ιδίως η μείωση της απόλυτης ισχύος του root μέσω SIP και signed system content αποδεικνύει ότι η Apple επιχειρεί να ανασχεδιάσει και κλασικές υποθέσεις των Unix-like desktop περιβαλλόντων.

Ως προς το δεύτερο ερευνητικό ερώτημα, η εμπιστευτικότητα στο macOS αποδείχθηκε πολυεπίπεδη. Το APFS παρέχει το κατάλληλο δομικό υπόβαθρο, το FileVault προστατεύει το startup volume, οι κλάσεις Data Protection και η Keychain διαφοροποιούν την πρόσβαση σε ευαίσθητα αντικείμενα και το Secure Enclave προσθέτει hardware-bound διαχείριση μυστικών. Άρα, η εμπιστευτικότητα δεν είναι ένα checkbox κρυπτογράφησης, αλλά οικοσύστημα κλειδιών, πολιτικών και διαβαθμίσεων ανάλογα με το είδος απειλής.

Ως προς το τρίτο ερευνητικό ερώτημα, η ασφάλεια εφαρμογών και η ακεραιότητα συστήματος στο macOS συγκροτούν όντως ώριμο μοντέλο εμπιστοσύνης. Το code signing, ο Gatekeeper και η notarization ελέγχουν την είσοδο λογισμικού, ενώ το sandboxing, το hardened runtime, το secure boot, το SSV και το TCC καθορίζουν τι μπορεί να συμβεί μετά την είσοδο αυτή. Το μοντέλο δεν είναι άτρωτο, αλλά εμφανίζει σαφή σχεδιαστική συνέπεια και υψηλότερο

βαθμό σύζευξης ανάμεσα σε πολιτική, υλοποίηση και hardware από ό,τι συναντάται συνήθως στα παραδοσιακά desktop λειτουργικά συστήματα.

8.3 Συνολική αξιολόγηση της αρχιτεκτονικής ασφάλειας του macOS

Η συνολική αξιολόγηση της αρχιτεκτονικής ασφάλειας του macOS είναι θετική, ακριβώς επειδή η πλατφόρμα καταφέρνει να ενσωματώνει διαφορετικές υπηρεσίες ασφάλειας σε ένα συνεκτικό μοντέλο. Η εμπιστευτικότητα, η ακεραιότητα και η ασφάλεια εφαρμογών δεν εξελίσσονται ως παράλληλες, ασύνδετες τεχνολογίες, αλλά ως αμοιβαία ενισχυόμενα στρώματα. Αυτό καθιστά το macOS ιδιαίτερα ενδιαφέρον ως μελέτη περίπτωσης για τη μετάβαση της ασφάλειας από «πρόσθετο χαρακτηριστικό» σε θεμελιώδη σχεδιαστική ιδιότητα του ίδιου του λειτουργικού συστήματος.

Ταυτόχρονα, η εργασία κατέδειξε ότι το υψηλό επίπεδο προστασίας συνοδεύεται από συγκεκριμένα κόστη: μεγαλύτερη εξάρτηση από τον κατασκευαστή, μικρότερη ευελιξία, αυξημένη πολυπλοκότητα εξαιρέσεων και μόνιμη ανάγκη ενημέρωσης. Η ασφάλεια του macOS είναι ισχυρή όχι επειδή εξαλείφει το ρίσκο, αλλά επειδή καθιστά πιο δύσκολη, πιο ακριβή και πιο ορατή την επιτυχή επίθεση. Υπό αυτήν την έννοια, η πλατφόρμα συνιστά ώριμο, αλλά όχι «τελικό», υπόδειγμα σύγχρονης ασφάλειας desktop λειτουργικών συστημάτων.

8.4 Προτάσεις για μελλοντική έρευνα

Μελλοντική θεωρητική και εμπειρική έρευνα θα μπορούσε να εστιάσει σε τρεις βασικές κατευθύνσεις. Πρώτον, στη μέτρηση της πραγματικής αποτελεσματικότητας των αλυσίδων εμπιστοσύνης εφαρμογών απέναντι σε σύγχρονες supply chain τεχνικές και σε σενάρια κατάχρησης έγκυρου αλλά ευάλωτου λογισμικού. Δεύτερον, στην ποσοτική αξιολόγηση της ισορροπίας μεταξύ ιδιωτικότητας και χρηστικότητας, ειδικά σε ό,τι αφορά το TCC, τα entitlements και τα prompts συγκατάθεσης. Τρίτον, στη διερεύνηση του κατά πόσο οι hardware-assisted μηχανισμοί του Apple silicon παραμένουν ανθεκτικοί απέναντι σε νέες μορφές μικροαρχιτεκτονικών επιθέσεων.

Συνολικά, το macOS παραμένει εξαιρετικά γόνιμο αντικείμενο μελέτης, επειδή βρίσκεται στο σημείο τομής ανάμεσα σε κλασικές αρχές ασφάλειας λειτουργικών συστημάτων και σε μια νέα γενιά tightly integrated hardware-software security architectures. Η περαιτέρω έρευνα πάνω στην πλατφόρμα δεν αφορά μόνο το ίδιο το macOS, αλλά και το ευρύτερο ερώτημα για το πώς θα μοιάζει η ασφάλεια των desktop λειτουργικών συστημάτων στην επόμενη δεκαετία.

Βιβλιογραφία

Acharya, J., Orlitsky, A., Suresh, A. T., & Tyagi, H. (2017). Estimating Rényi entropy of discrete distributions. IEEE Transactions on Information Theory, 63(1), 38-56. https://doi.org/10.1109/TIT.2016.2620435

Apple. (2020a). Apple File System Reference. Apple Developer. https://developer.apple.com/support/downloads/Apple-File-System-Reference.pdf

Apple. (2021a). App code signing process in macOS. Apple Platform Security. https://support.apple.com/guide/security/sec3ad8e6e53/web

Apple. (2021b). System Integrity Protection. Apple Platform Security. https://support.apple.com/guide/security/system-integrity-protection-secb7ea06b49/web

Apple. (2021c). Boot process for a Mac with Apple silicon. Apple Platform Security. https://support.apple.com/guide/security/boot-process-secac71d5623/web

Apple. (2021d). Controlling app access to files in macOS. Apple Platform Security. https://support.apple.com/guide/security/controlling-app-access-to-files-secddd1d86a6/web

Apple. (2021e). Sealed Key Protection (SKP). Apple Platform Security. https://support.apple.com/guide/security/sealed-key-protection-skp-secdc7c6c88e/web

Apple. (2022a). Intro to app security for macOS. Apple Platform Security. https://support.apple.com/guide/security/secf826eff27/web

Apple. (2022b). Signed system volume security. Apple Platform Security. https://support.apple.com/guide/security/signed-system-volume-security-secd698747c9/web

Apple. (2024a). macOS virtual address translation. https://opensource.apple.com/source/xnu/xnu-7195.81.3/osfmk/arm64/proc_reg.h.auto.html

Apple. (2024b). Data Protection overview. Apple Platform Security. https://support.apple.com/guide/security/data-protection-overview-secf6276da8a/web

Apple. (2024c). Data Protection classes. Apple Platform Security. https://support.apple.com/guide/security/data-protection-classes-secb010e978a/web

Apple. (2024d). Keychain data protection. Apple Platform Security. https://support.apple.com/guide/security/keychain-data-protection-secb0694df1a/web

Apple. (2024e). The Secure Enclave. Apple Platform Security. https://support.apple.com/guide/security/the-secure-enclave-sec59b0b31ff/web

Apple. (2024f). Gatekeeper and runtime protection in macOS. Apple Platform Security. https://support.apple.com/guide/security/sec5599b66df/web

Apple. (2024g). Protecting against malware in macOS. Apple Platform Security. https://support.apple.com/guide/security/protecting-against-malware-sec469d47bd8/web

Apple. (2024h). Use secure token, bootstrap token, and volume ownership in deployments. Apple Platform Deployment. https://support.apple.com/guide/deployment/dep24dbdcf9e/web

Apple. (2024i). Privacy Preferences Policy Control device management payload settings for Apple devices. Apple Platform Deployment. https://support.apple.com/guide/deployment/privacy-preferences-policy-control-payload-dep38df53c2a/web

Apple. (2024j). Role of Apple File System. Apple Platform Security. https://support.apple.com/guide/security/seca6147599e/web

Apple. (2025a). Intro to FileVault. Apple Platform Deployment. https://support.apple.com/guide/deployment/intro-to-filevault-dep82064ec40/web

Apple. (2026a). Volume encryption with FileVault in macOS. Apple Platform Security. https://support.apple.com/guide/security/volume-encryption-with-filevault-sec4c6dc1b6e/web

Apple. (2026b). App security overview. Apple Platform Security. https://support.apple.com/guide/security/app-security-overview-sec35dd877d0/web

Apple. (2026c). Operating system integrity. Apple Platform Security. https://support.apple.com/guide/security/operating-system-integrity-sec8b776536b/web

Apple. (2026d). Intro to Apple platform security. Apple Platform Security. https://support.apple.com/guide/security/intro-to-apple-platform-security-seccd5016d31/web

Apple Developer. (n.d.-a). App Sandbox. https://developer.apple.com/documentation/security/app-sandbox

Apple Developer. (n.d.-b). Configuring the hardened runtime. https://developer.apple.com/documentation/xcode/configuring-the-hardened-runtime

Apple Developer. (n.d.-c). Notarizing macOS software before distribution. https://developer.apple.com/documentation/security/notarizing-macos-software-before-distribution

Apple Developer. (n.d.-d). Configuring the macOS App Sandbox. https://developer.apple.com/documentation/xcode/configuring-the-macos-app-sandbox

ARM. (2019). ARMv8-A address translation. https://documentation-service.arm.com/static/5efa1d23dbdee951c1ccdec5

Barker, E. (2020). Recommendation for Key Management: Part 1 - General (NIST SP 800-57 Part 1 Rev. 5). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-57pt1r5

CVEdetails.com. (2024). Top 50 products by total number of distinct vulnerabilities. https://www.cvedetails.com/top-50-products.php

Díaz, R. V., Rivera-Dourado, M., Pérez-Jove, R., Vila Avendaño, P., & Vázquez-Naya, J. M. (2021). Address space layout randomization comparative analysis on Windows 10 and Ubuntu 18.04 LTS. Engineering Proceedings, 7(1), 26. https://doi.org/10.3390/engproc2021007026

Ding, Y., Peng, Z., Zhou, Y., & Zhang, C. (2014). Android low entropy demystified. In Proceedings of the IEEE International Conference on Communications (ICC 2014) (pp. 659-664). IEEE. https://doi.org/10.1109/ICC.2014.6883394

Herlands, W., Hobson, T., & Donovan, P. J. (2014). Effective entropy: Security-centric metric for memory randomization techniques. In Proceedings of the 7th Workshop on Cyber Security Experimentation and Test (CSET 2014). USENIX Association. https://www.usenix.org/conference/cset14/workshop-program/presentation/herlands

Kaplan, D., Kedmi, S., Hay, R., & Dayan, A. (2014). Attacking the Linux PRNG on Android: Weaknesses in seeding of entropic pools and low boot time entropy. In Proceedings of the 8th USENIX Workshop on Offensive Technologies (WOOT 2014). USENIX Association. https://www.usenix.org/conference/woot14/workshop-program/presentation/kaplan

Kerrisk, M. (2023). pthread_create(3) - Linux manual. https://man7.org/linux/man-pages/man3/pthread_create.3.html

Li, L., Just, J. E., & Sekar, R. (2006). Address-space randomization for Windows systems. In Proceedings of the 22nd Annual Computer Security Applications Conference (ACSAC 2006) (pp. 329-338). IEEE. https://doi.org/10.1109/ACSAC.2006.10

Marco-Gisbert, H., & Ripoll, I. (2014). On the effectiveness of full ASLR on 64-bit Linux. In Proceedings of the In-Depth Security Conference.

Marco-Gisbert, H., & Ripoll, I. (2019). Address space layout randomization next generation. Applied Sciences, 9(14), 2928. https://doi.org/10.3390/app9142928

Microsoft. (2025a). App Control for Business and AppLocker Overview. Microsoft Learn. https://learn.microsoft.com/en-us/windows/security/application-security/application-control/app-control-for-business/appcontrol-and-applocker-overview

Microsoft. (2025b). System security - Windows 11. Microsoft Learn. https://learn.microsoft.com/en-us/windows/security/book/operating-system-security-system-security

Nemenman, I., Shafee, F., & Bialek, W. (2001). Entropy and inference, revisited. In Advances in Neural Information Processing Systems 14 (pp. 471-478). MIT Press. https://proceedings.neurips.cc/paper/2001/hash/d46e1fcf4c07ce4a69ee07e4134bcef1-Abstract.html

Qualys Security Advisory. (2023). Looney Tunables: Local privilege escalation in the glibc's ld.so (CVE-2023-4911). https://www.qualys.com/2023/10/03/cve-2023-4911/looney-tunables-local-privilege-escalation-glibc-ld-so.txt

SecurityScorecard. (2023). CVE security vulnerability database. https://www.cvedetails.com/vulnerabilities-by-types.php

Shannon, C. E. (1948). A mathematical theory of communication. Bell System Technical Journal, 27(3), 379-423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x

StatCounter Global Stats. (2023a). Desktop operating system market share worldwide. https://gs.statcounter.com/os-market-share/desktop/worldwide

Torvalds, L. (2021a). Memory folios commit. GitHub. https://github.com/torvalds/linux/commit/49f8275c7d92

Torvalds, L. (2021b). Memory management in Linux 5.15. GitHub. https://github.com/torvalds/linux/blob/v5.15/Documentation/x86/x86_64/mm.rst

Ubuntu. (2025a). Ubuntu security features. Canonical Documentation. https://documentation.ubuntu.com/security/security-features/

Ubuntu. (2026a). AppArmor. Canonical Documentation. https://ubuntu.com/server/docs/how-to/security/apparmor/

Ubuntu. (2026b). UEFI Secure Boot. Canonical Documentation. https://documentation.ubuntu.com/security/security-features/platform-protections/secure-boot/

Whitehouse, O. (2007). An analysis of address space layout randomization on Windows Vista. Symantec Advanced Threat Research.

Wilcox, M. (2021). Memory folios. LWN.net. https://lwn.net/Articles/856016/

Yu, J., Dutta, A., Jaeger, T., Kohlbrenner, D., & Fletcher, C. W. (2023). Synchronization storage channels (S2C): Timer-less cache side channel attacks on the Apple M1 via hardware synchronization instructions. In Proceedings of the 32nd USENIX Security Symposium (pp. 1973-1990). USENIX Association. https://www.usenix.org/conference/usenixsecurity23/presentation/yu-jiyong