



Ιόνιο Πανεπιστήμιο

Τμήμα Πληροφορικής

Ανίχνευση Ιομορφικού Λογισμικού με Χρήση Μηχανικής Μάθησης

Τριμελής Επιτροπή:

1. Νταντογιάν Χριστόφορος (Επιβλέπων Καθηγητής)
2. Τσώχου Αγγελική
3. Μάγγος Εμμανουήλ

Πτυχιακή εργασία

Φανή Τσιλινίκου

ΑΜ : Π2016092

Σεπτέμβριος, 2022

Ανίχνευση Ιομορφικού Λογισμικού με Χρήση Μηχανικής Μάθησης

Βιβλιογραφική Έρευνα

Ευχαριστίες

Με την ολοκλήρωση της πτυχιακής εργασίας, θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες σε όλους όσους συνέβαλλαν στην εκπόνησή της. Αρχικά, θα ήθελα να ευχαριστήσω θερμά τον επιβλέπων καθηγητή μου, κύριο Χριστόφορο Νταντογιάν κυρίως για την εμπιστοσύνη που μου έδειξε από την πρώτη στιγμή της συνεργασίας μας αναθέτοντάς μου ένα ενδιαφέρον κι επίκαιρο θέμα. Επίσης, η επιστημονική του καθοδήγηση, η υπομονή, οι υποδείξεις, η συμπαράσταση, το αμείωτο ενδιαφέρον αλλά και η επιμονή που έδειξε από την αρχή μέχρι το τέλος της εργασίας με βοήθησαν πολύ. Ακόμη, θα ήθελα να ευχαριστήσω όλο το διδακτικό προσωπικό για την προσπάθεια που έκανε όλα αυτά τα χρόνια να με καθοδηγήσουν σωστά και να εκπληρώσω έγκαιρα τις σπουδές μου. Επιπλέον, ένα μεγάλο ευχαριστώ πρέπει να αποδοθεί στους φίλους και συμφοιτητές μου για τις ατελείωτες ώρες διαβάσματος αλλά και τους ατελείωτους καφέδες που μοιραστήκαμε, το άγχος, τις ανησυχίες αλλά και τις χαρές. Τέλος, θερμές ευχαριστίες αλλά και αμείωτη ευγνωμοσύνη θέλω να εκφράσω στην οικογένεια μου που στάθηκε δίπλα μου σε κάθε δυσκολία, με στήριξε, μου συμπαραστάθηκε και κατανόησε κάθε δύσκολη στιγμή σε όλα μου τα φοιτητικά χρόνια.

Περίληψη

Η διαρκής αύξηση της χρήσης πληροφοριακών συστημάτων και διαδικτυακών υπηρεσιών έχει πυροδοτήσει τη γέννηση ενός νέου τύπου κακόβουλου λογισμικού που είναι πιο επικίνδυνο και σκληρό στην ανίχνευσή του. Συγκεκριμένα, σύμφωνα με τις πρόσφατες αναφορές, ο νέος τύπος κακόβουλου λογισμικού χωρίς αρχεία μολύνει τις συσκευές των θυμάτων χωρίς μόνιμο ίχνος (δηλαδή αρχείο) στους σκληρούς δίσκους. Επιπλέον, οι υπάρχουσες μέθοδοι ανίχνευσης κακόβουλου λογισμικού που υπάρχουν στη βιβλιογραφία συχνά αποτυγχάνουν να ανιχνεύσουν το εξελιγμένο κακόβουλο λογισμικό που χρησιμοποιεί διάφορες τεχνικές συσκότισης και κρυπτογράφησης. Αυτή η μελέτη στοχεύει στην παρουσίαση και σύγκριση διαφόρων επιστημονικών άρθρων τα οποία αναφέρονται στην αντιμετώπιση του λογισμικού Android Malware και λύνονται με χρήση μηχανικής μάθησης (Machine Learning). Αρχικά γίνεται μία θεωρητική προσέγγιση του θέματος αναλύοντας τα είδη κακόβουλου λογισμικού ειδικά σε εξυπηρετητές διαδικτύου. Στη συνέχεια, περιγράφονται οι μεθοδολογίες εντοπισμού κακόβουλου λογισμικού και η εφαρμογή τους στον υπό ανάπτυξη μηχανισμό. Επιπλέον, γίνεται μία αναφορά στο μηχανισμό μηχανικής μάθησης. Ακολουθεί η παρουσίαση της έρευνας, όπου γίνεται μία εκτενή ανάλυση των επιστημονικών άρθρων που μελετήθηκαν. Αρχικά, παρουσιάζεται το πρόβλημα και η μεθοδολογία που ακολουθήθηκε για την επίλυση του. Έπειτα, ακολουθεί η περιγραφή των αποτελεσμάτων κι εξάγονται μερικά συμπεράσματα. Τα συμπεράσματα δείχνουν ότι η αντιμετώπιση του εντοπισμού κακόβουλου λογισμικού Android με χρήση τεχνικών μηχανικής εκμάθησης απαιτεί περαιτέρω έρευνα. Θα ήταν επιθυμητό να προταθούν νέα πλαίσια και αλγόριθμοι που είναι αρκετά γρήγοροι και ισχυροί για τον εντοπισμό του κακόβουλου λογισμικού. Με βάση τα αναθεωρημένα έγγραφα, ο παραλληλισμός ή η ενσωμάτωση μιας ή περισσότερων καλών τεχνικών μαζί μπορεί να οδηγήσει σε ακριβή και αποτελεσματικό μηχανισμό. Η μηχανική μάθηση και πιο συγκεκριμένα η μέθοδος Random Forest, παρουσίασε στις περισσότερες περιπτώσεις την υψηλότερη ακρίβεια.

Λέξεις κλειδιά: ιομορφικό, κακόβουλο, λογισμικό, μηχανική, μάθηση

Abstract

The continuous increase in the use of information systems and online services has triggered the birth of a new type of malware that is more dangerous and harder to detect. Specifically, according to recent reports, the new type of fileless malware infects victims' devices without a permanent trace (i.e. file) on their hard drives. Additionally, existing malware detection methods found in the literature often fail to detect sophisticated malware that uses various obfuscation and encryption techniques. This study aims to present and compare various scientific articles that refer to dealing with android malware software and are solved using machine learning (Machine Learning). First, a theoretical approach to the subject is made by analyzing the types of malicious software especially in internet servers. Next, the malware detection methodologies and their application to the mechanism under development are described. In addition, a reference is made to the machine learning mechanism. The presentation of the research follows, where an extensive analysis of the scientific articles studied is made. First, the problem and the methodology followed to solve it are presented. Then follows the description of the results and some conclusions are drawn. The conclusions indicate that addressing Android malware detection using machine learning techniques requires further research. It would be desirable to propose new frameworks and algorithms that are fast and robust enough to detect malware. Based on the reviewed papers, parallelizing or integrating one or more good techniques together can lead to an accurate and efficient mechanism. Machine learning, and more specifically the Random Forest method, presented the highest accuracy in most cases.

Keywords: virus, malware, software, machine, learning, android

Πίνακας Περιεχομένων

<i>Ευχαριστίες</i>	iv
Περίληψη	v
Abstract	vi
Πίνακας Περιεχομένων.....	vii
Κατάλογος Πινάκων	x
Κατάλογος Σχημάτων	xii
1. Εισαγωγή	1
2. Είδη κακόβουλου λογισμικού ειδικά σε εξυπηρετητές διαδικτύου.....	5
2.1. Απομακρυσμένη πρόσβαση (Remote Access).....	5
2.1.1. Η έννοια της απομακρυσμένης πρόσβασης.....	5
2.1.2. Η λειτουργία της απομακρυσμένης πρόσβασης	6
2.1.3. Οι τύποι της απομακρυσμένης πρόσβασης.....	7
2.1.4. Τα πρωτόκολλα της απομακρυσμένης πρόσβασης	8
2.2. Κερκόπορτες (Backdoors).....	9
2.2.1. Η έννοια της κερκόπορτας.....	9
2.2.2. Η λειτουργία της κερκόπορτας	10
2.2.3. Ανίχνευση και πρόληψη	15
2.2.4. Διάσημες επιθέσεις σε κερκόπορτα	18
2.3. Κέντρο διαχείρισης δικτύου Botnet (Command and Control Center)	19
2.4. Drive by Download Attack	20

2.4.1.	Αποτροπή από τους ιδιοκτήτες των ιστοτόπων	24
2.4.2.	Αποτροπή από τους χρήστες των ιστοτόπων	25
3.	Εφαρμογές Android	27
3.1.	Αρχιτεκτονική συστήματος Android	27
3.2.	Μηχανισμοί ασφαλείας Android.....	28
3.3.	Ταξινόμηση κακόβουλου λογισμικού Android.....	29
4.	Προσεγγίσεις ανίχνευσης κακόβουλου λογισμικού Android με βάση τη μηχανική εκμάθηση. Εισαγωγή στην μηχανική μάθηση	30
5.	Κοινά μοντέλα και αλγόριθμοι μηχανικής μάθησης	33
5.1.	Εισαγωγή στα κοινά μοντέλα και αλγόριθμους.....	33
5.2.	Αξιολόγηση της αποτελεσματικότητας ανίχνευσης.....	37
5.2.1.	Διαίρεση συνόλων δεδομένων	37
5.2.2.	Αξιολόγηση της απόδοσης του ταξινομητή.....	38
6.	Παραδείγματα προβλημάτων κακόβουλου λογισμικού (malware), λύση τους και παρουσίαση μεθοδολογιών	40
7.	Σύγκριση των τρόπων με τους οποίους λύνεται το κάθε πρόβλημα κακόβουλου λογισμικού (malware) και αποτελέσματα.....	76
8.	Γραφήματα.....	77
9.	Συμπεράσματα	79
10.	Λέξεις Κλειδιά	81
11.	Βιβλιογραφία	83

Κατάλογος Πινάκων

Πίνακας 1 Άρθρο 1: Ανίχνευση κακόβουλου λογισμικού με προσέγγιση μέσω της εγκληματολογίας μνήμης, πολλαπλής μάθησης και όρασης του υπολογιστή.....	40
Πίνακας 2 Άρθρο 2: Ένα νέο πλαίσιο για κακόβουλο λογισμικό που βασίζεται σε εικόνες.....	41
Πίνακας 1 Άρθρο 3: A survey on android malware detection techniques using machine learning algorithm.....	43
Πίνακας 4 Άρθρο 4: Χρήση της εικόνας Rgb του κώδικα μηχανής για την ταξινόμηση του κακόβουλου λογισμικού.....	44
Πίνακας 5 Άρθρο 5: Ταξινόμηση εικόνων κακόβουλου λογισμικού με μοντέλα συνελκτικού νευρωνικού δικτύου.....	466
Πίνακας 6 Άρθρο 6: Ισχυρότητα κακόβουλου λογισμικού Android που βασίζεται σε εικόνα - Ανίχνευση κάτω από αντίθετες επιθέσεις.....	47
Πίνακας 7 Άρθρο 7: Ανίχνευση κακόβουλου λογισμικού Android βάσει δυναμικών δικαιωμάτων με χρήση τεχνικών μηχανικής εκμάθησης.....	49
Πίνακας 8 Άρθρο 8: AdDroid: Πλαίσιο μηχανικής μάθησης βάσει κανόνων για ανάλυση κακόβουλου λογισμικού Android.....	50
Πίνακας 9 Άρθρο 9: Ένα σύστημα ανίχνευσης κακόβουλου λογισμικού Android βασισμένο σε μηχανική μάθηση.....	51
Πίνακας 10 Άρθρο 10: Ταξινόμηση κακόβουλου λογισμικού Android με τη βοήθεια μηχανικής μάθησης.....	52
Πίνακας 11: Άρθρο 11: Ανίχνευση κακόβουλου λογισμικού Android μέσω μηχανικής εκμάθησης σε δομές πυρήνα.....	53
Πίνακας 12 Άρθρο 12: Μια προσέγγιση μηχανικής εκμάθησης για τον εντοπισμό κακόβουλου λογισμικού Android.....	54

Πίνακας 13 Άρθρο 12: Μια ανασκόπηση των προσεγγίσεων εντοπισμού κακόβουλου λογισμικού Android με βάση τη μηχανική εκμάθηση.....	55
Πίνακας 14 Άρθρο 14: Μια έρευνα σχετικά με τις τεχνικές ανίχνευσης κακόβουλου λογισμικού Android.....	56
Πίνακας 15 Άρθρο 15: Ανάλυση της απόδοσης εντοπισμού κακόβουλου λογισμικού Android χρησιμοποιώντας ταξινομητές μηχανικής μάθησης.....	57
Πίνακας 16: Άρθρο 16: Ανίχνευση κακόβουλου λογισμικού Android με χρήση ταξινομητών παράλληλης μηχανικής εκμάθησης.....	58
Πίνακας 17 Άρθρο 17: Εφαρμογή ταξινομητών μηχανικής μάθησης σε δυναμική ανίχνευση κακόβουλου λογισμικού Android σε κλίμακα	59
Πίνακας 18 Άρθρο 18: Μια ολοκληρωμένη έρευνα για τις τεχνικές μηχανικής μάθησης για την ανίχνευση κακόβουλου λογισμικού Android	60
Πίνακας 19 Άρθρο 19: Μηχανική εκμάθηση για ανίχνευση κακόβουλου λογισμικού Android κάνοντας χρήση αδειών και κλήσεων API.....	61
Πίνακας 20 Άρθρο 20: Σημαντική αναγνώριση άδειας για Android που βασίζεται σε μηχανική μάθηση ανίχνευση κακόβουλου λογισμικού	63
Πίνακας 21 Άρθρο 21: Ναι, η μηχανική μάθηση μπορεί να είναι πιο ασφαλής! Μια μελέτη περίπτωσης για την ανίχνευση κακόβουλου λογισμικού Android	64
Πίνακας 22 Άρθρο 22: Μια μέθοδος συνδυασμού για την ανίχνευση κακόβουλου λογισμικού Android με βάση γραφήματα ροής ελέγχου και αλγόριθμους μηχανικής εκμάθησης.....	65
Πίνακας 23 Άρθρο 23: Αποτελεσματικός εντοπισμός κακόβουλου λογισμικού Android με βάση τη χρήση των API ροής δεδομένων και τη μηχανική εκμάθηση	66
Πίνακας 24 Άρθρο 24: EMULATOR vs REAL PHONE: Ανίχνευση κακόβουλου λογισμικού Android με χρήση μηχανικής εκμάθησης.....	67

Πίνακας 25 Άρθρο 25: Ανίχνευση κακόβουλου λογισμικού Android με χρήση βελτιστοποιημένης επιλογής λειτουργιών και μηχανικής εκμάθησης βάσει γενετικού αλγόριθμου.....	68
Πίνακας 26 Άρθρο 26: Αξιολόγηση μοντέλων μηχανικής εκμάθησης για ανίχνευση κακόβουλου λογισμικού Android: Μια μελέτη σύγκρισης	69
Πίνακας 27 Άρθρο 27: Εφαρμογή αλγορίθμων μηχανικής μάθησης για ανίχνευση κακόβουλου λογισμικού Android	70
Πίνακας 28 Άρθρο 28: Μια έρευνα για την τεχνολογία στατικής ανίχνευσης κακόβουλου λογισμικού Android με βάση τη μηχανική μάθηση	71
Πίνακας 29 Άρθρο 29: Ανίχνευση κακόβουλου λογισμικού Android με βάση χρήσιμες κλήσεις API και μηχανική εκμάθηση	72
Πίνακας 30 Άρθρο 30: Ανίχνευση κακόβουλου λογισμικού Android με βάση λειτουργίες που βασίζονται σε εικόνες και τεχνικές μηχανικής εκμάθησης	74

Κατάλογος Σχημάτων

Σχήμα 1 Αρχιτεκτονική πλατφόρμας Android (developers, 2021).....	27
---	----

1. Εισαγωγή

Η αλματώδης εξέλιξη της τεχνολογίας τις τελευταίες δεκαετίες αποτελεί καταλυτικό παράγοντα στην ανάπτυξη και την διάδοση του διαδικτύου, φέροντάς το στη θέση που έχει σήμερα ως αναπόσπαστο κομμάτι της καθημερινής ζωής σχεδόν όλων των ανθρώπων, ειδικότερα στις δυτικές κοινωνίες, όχι μόνο για επικοινωνία, αλλά και για την πραγματοποίηση συναλλαγών με δημόσιους και ιδιωτικούς τομείς. Κάθε πολίτης, συνεπώς, παρέχει στο διαδίκτυο κρίσιμες και προσωπικές πληροφορίες, όπως στοιχεία για την προσωπική αλλά και κοινωνική του ζωή, την οικονομική του κατάσταση, τις πολιτικές και θρησκευτικές ιδέες, τις προσωπικές του προτιμήσεις και ενδιαφέροντα κ.α.. Η συνεχόμενη αύξηση των χρηστών και η προσθήκη νέων πληροφοριών έχει καταστήσει τους ιστοτόπους στόχο κυβερνοεπιθέσεων. Ουσιαστικά, αποτελούν ένα χώρο που προσπαθούν να αντλήσουν πληροφορίες και δεδομένα και να εξαπατήσουν το χρήστη, σκοπεύοντας σε οικονομικά, πολιτικά, θρησκευτικά και άλλα συμφέροντα.

Η παραβίαση ενός ιστοτόπου καθιστά τον διαχειριστή του σε θέση συνεχών ελέγχων με προγράμματα anti-malware, έτσι ώστε να καταφέρει να εντοπίσει την κερκόπορτα ή άλλο είδος λογισμικού και να επαναφέρει τον ιστότοπο στην αρχική του κατάσταση. Οι υπάρχουσες τεχνικές χρησιμοποιούνται ως τεχνικές εντοπισμού ύποπτων συναρτήσεων ή υπογραφών και χρησιμοποιούνται από κακόβουλες εφαρμογές που καταγράφουν κάθε κακόβουλο αρχείο, αποθηκεύοντας την τιμή συνάρτησης κατακερματισμού (όπως η MD5 ή η SHA1), κάτι που αποτελεί πρόβλημα. Τα τρία πιο σημαντικά ελαττώματα των τεχνικών αυτών είναι τα εξής:

1. Στην περίπτωση εντοπισμού ύποπτων συναρτήσεων, η εφαρμογή παρουσιάζει έναν μεγάλο αριθμό σφαλμάτων true – negative, καθώς η ύπαρξη μίας ή περισσότερων «ύποπτων» συναρτήσεων στα υπό εξέταση αρχεία, δεν αποτελεί αξιόπιστο στοιχείο για την ύπαρξη ή όχι κακόβουλου κώδικα.
2. Στην περίπτωση καταγραφής της τιμής κατακερματισμού των αρχείων, απαιτείται το αρχείο να είναι ήδη γνωστό στους μηχανισμούς, αλλά λόγω της φύσης των εν λόγω μηχανισμών δημιουργίας σύνοψης, η παραμικρή αλλαγή στα περιεχόμενα του αρχείου θα οδηγήσει σε τελείως διαφορετική τιμή, με

αποτέλεσμα η σύγκριση μεταξύ των τιμών να οδηγήσει σε λανθασμένο συμπέρασμα.

3. Σε όλες τις περιπτώσεις το κακόβουλο λογισμικό πρέπει να είναι γνωστό στον μηχανισμό. Καθώς οι κακόβουλοι χρήστες γνωρίζουν αυτήν την αδυναμία, είναι σε θέση να τροποποιούν ή να αναβαθμίζουν τους υπάρχοντες κακόβουλους κώδικες ή ακόμα και να τους επανασχεδιάζουν, με σκοπό να ξεγελούν τους μηχανισμούς αντιμετώπισης, πετυχαίνοντας έτσι την όσο το δυνατόν μεγαλύτερη διάρκεια παραμονής τους στους ιστοτόπους.

Η μηχανική μάθηση ως έννοια υφίσταται από το 1959, όταν ο Arthur Samuel όρισε τη μηχανική μάθηση ως "Πεδίο μελέτης που δίνει στους υπολογιστές την ικανότητα να μαθαίνουν, χωρίς να έχουν ρητά προγραμματιστεί" (Egham, 2018). Η αξιοποίησή της άρχισε τα τελευταία χρόνια, αφού τα υπολογιστικά συστήματα πλέον μπορούν να εκτελέσουν πολλούς πολύπλοκους μαθηματικούς υπολογισμούς σε γρήγορο χρονικό διάστημα. Με αυτό τον τρόπο, οι αλγόριθμοι για τη μηχανική μάθηση βοηθούν τον υπολογιστή να διαχωρίζει αν ένα μήνυμα ηλεκτρονικού ταχυδρομείου (email) είναι επιθυμητό ή ανεπιθύμητο (spam), να διαβάζει οπτικούς χαρακτήρες από κάμερες (OCR) κ.α..

Η χρήση των smartphone αυξάνεται ραγδαία με την πάροδο των ετών, με περισσότερες από 408 εκ. συσκευές να πωλούνται στο τέταρτο τρίμηνο του 2017 και να παρουσιάζεται αύξηση 174% σε σχέση με τις 149 εκ. συσκευές που πωλήθηκαν την ίδια περίοδο του 2011. Πιο συγκεκριμένα, οι χρήστες Android έχουν αυξηθεί κατά 1,639.15%, δηλαδή από 76 εκ. το 2011 έφτασαν τα 1.3 δις. το 2017. Αυτή η αξιοσημείωτη αύξηση των χρηστών Android έχει εκμεταλλευτεί από τους δημιουργούς κακόβουλου λογισμικού, που την εκμεταλλεύονται για να επηρεάσουν και να βλάψουν έναν τεράστιο αριθμό από χρήστες (Egham, 2018).

Το κύριο μέλημα στην πλατφόρμα Android είναι ο εντοπισμός του κακόβουλου λογισμικού. Υπάρχουν δύο κύριες προσεγγίσεις για την ανίχνευση κακόβουλου λογισμικού που βασίζονται στη μηχανική μάθηση και άλλες προσεγγίσεις που δεν βασίζονται σε αυτή. Η ανίχνευση του κακόβουλου λογισμικού που βασίζεται στα συστήματα της μηχανικής μάθησης, χρησιμοποιούν διαφορετικούς αλγόριθμους αυτής στη διαδικασία ανίχνευσης. Οι ανιχνευτές κακόβουλου λογισμικού που δεν βασίζονται σε μηχανική μάθηση χρησιμοποιούν υπογραφές, άδειες ή δυναμική ανάλυση για την

ανίχνευση των κακόβουλων αρχείων. Με την τεράστια παραγωγή κακόβουλου λογισμικού στις μέρες μας, οι προσεγγίσεις για την ανίχνευση κακόβουλου λογισμικού Android που δεν βασίζεται σε μηχανική εκμάθηση καταναλώνουν χρόνο, καθώς και η αδυναμία εντοπισμού κακόβουλου λογισμικού πολλές φορές δεν εμφανίζεται. Έτσι, στις συσκευές Android πρέπει να παρέχεται η δυνατότητα εντοπισμού κακόβουλου λογισμικού με χρήση αλγόριθμων μηχανικής μάθησης. Οι αλγόριθμοι μηχανικής μάθησης βρίσκουν εφαρμογή όλο και περισσότερο στον τομέα της ασφάλειας, ως απάντηση στην αυξανόμενη και απρόβλεπτη πολυπλοκότητα του σύγχρονου κακόβουλου λογισμικού (Anderson, Kharkar, Filar, & Roth, 2017).

Ανάμεσα στις διάφορες υπάρχουσες προσεγγίσεις για τον εντοπισμό κακόβουλου λογισμικού, οι αλγόριθμοι και οι τεχνικές μηχανικής μάθησης πέτυχαν υψηλή ακρίβεια στον εντοπισμό του κακόβουλου λογισμικού (Balaganesh, Chakrabarti, & Midhunchakkaravarthy, 2018). Πολλοί ερευνητές πρότειναν διάφορα πλαίσια για τον εντοπισμό κακόβουλου λογισμικού. Αυτά τα πλαίσια αναπτύσσονται με βάση διαφορετική μηχανική μάθηση με αλγόριθμους όπως SVM, Naive Bayes, Perceptron και Deep Αλγόριθμοι νευρωνικών δικτύων.

Αυτό το άρθρο ερευνά την κατάσταση που επικρατεί στο κακόβουλο λογισμικό και τις τεχνικές ανίχνευσης κακόβουλου λογισμικού εστιάζοντας στη μηχανική μάθηση με χρήση αλγορίθμων βασισμένοι στη μάθηση που χρησιμοποιούνται για τον εντοπισμό κακόβουλου λογισμικού στις συσκευές. Το υπόλοιπο αυτής της εργασίας οργανώνεται ως εξής:

Το 1ο κεφάλαιο – το παρόν – αποτελεί την εισαγωγή της εργασίας όπου αναφέρονται γενικές πληροφορίες των λογισμικών.

Στο 2ο κεφάλαιο παρουσιάζονται τα είδη του κακόβουλου λογισμικού ειδικά για εξυπηρετητές του διαδικτύου που εγκαθίσταται στους σύγχρονους ιστοτόπους. Πιο συγκεκριμένα, αναλύονται: η απομακρυσμένη πρόσβαση (remote access), οι κερκόπορτες (backdoors), το κέντρο διαχείρισης δικτύου botnet (command and control center) και το drive by download attack.

Στο 3ο κεφάλαιο παρουσιάζονται οι μεθοδολογίες εντοπισμού κακόβουλου λογισμικού και η εφαρμογή τους στον υπό ανάπτυξη μηχανισμό. Πιο αναλυτικά, γίνεται χαρτογράφηση και έλεγχος ακεραιότητας των αρχείων του ιστότοπου, στατική

ανάλυση ή στοχευμένη αναζήτηση, υπολογισμός εντροπίας αρχείων και στατιστική ανάλυση αρχείων.

Στο 4ο κεφάλαιο γίνεται μια αναφορά και περί του μηχανισμού της μηχανικής μάθησης, εισαγωγή στην μηχανική μάθηση, δίδεται επεξήγηση γιατί επιλέχθηκε ο αλγόριθμος Naïve Bayes ως εργαλείο εντοπισμού κακόβουλου κώδικα και γίνεται ανάλυση μέσω παραδειγμάτων του τρόπου αξιοποίησής του από τον υπό ανάπτυξη μηχανισμό εντοπισμού κακόβουλου λογισμικού με αλγόριθμο μηχανικής μάθησης.

Στο 5ο κεφάλαιο δίδεται επεξήγηση του τρόπου εκπαίδευσης του μηχανισμού και γίνεται οριοθέτηση του δειγματικού χώρου στον οποίο εισήχθησαν οι πληροφορίες του μηχανισμού μηχανικής μάθησης, η δημιουργία υπογραφών μηχανισμού και η εκπαίδευση μηχανισμού μηχανικής μάθησης.

Στο 6ο κεφάλαιο γίνεται σύγκριση των τρόπων με τους οποίους λύνεται το κάθε πρόβλημα του malware σε 30 επιστημονικά άρθρα. Πιο συγκεκριμένα, παρουσιάζεται συνοπτικά το πρόβλημα, η λύση του, η μεθοδολογία που ακολουθήθηκε, καθώς και τα κύρια αποτελέσματα.

Τέλος, στο 7ο κεφάλαιο παρουσιάζονται τα συμπεράσματα και οι παρατηρήσεις για την ορθή λειτουργία του μηχανισμού ως προς τον εντοπισμό κακόβουλου λογισμικού και κατά πόσο είναι εφικτός ο περιορισμός τυχόν λαθών, με σκοπό την βέλτιστη λειτουργία του.

2. Είδη κακόβουλου λογισμικού ειδικά σε εξυπηρετητές διαδικτύου

2.1. Απομακρυσμένη πρόσβαση (Remote Access)

2.1.1. Η έννοια της απομακρυσμένης πρόσβασης

Η απομακρυσμένη πρόσβαση είναι η δυνατότητα ενός εξουσιοδοτημένου ατόμου να έχει πρόσβαση σε υπολογιστή ή δίκτυο από γεωγραφική απόσταση μέσω μιας σύνδεσης δικτύου. Η απομακρυσμένη πρόσβαση επιτρέπει στους χρήστες να συνδέονται με τα συστήματα που χρειάζονται όταν βρίσκονται φυσικά μακριά. Αυτό είναι ιδιαίτερα σημαντικό για υπαλλήλους που εργάζονται σε υποκαταστήματα, ταξιδεύουν ή τηλεεργάζονται. Η απομακρυσμένη πρόσβαση επιτρέπει στους χρήστες που βρίσκονται μακριά να έχουν πρόσβαση σε αρχεία και άλλους πόρους συστήματος σε οποιεσδήποτε συσκευές ή διακομιστές που είναι συνδεδεμένοι στο δίκτυο ανά πάσα στιγμή. Αυτό αυξάνει την παραγωγικότητα των εργαζομένων και δίνει τη δυνατότητα στους υπαλλήλους να συνεργάζονται καλύτερα με συναδέλφους τους σε όλο τον κόσμο.

Μια στρατηγική απομακρυσμένης πρόσβασης δίνει στους οργανισμούς την ευελιξία να προσλαμβάνουν τα καλύτερα talenta ανεξάρτητα από την τοποθεσία, να αφαιρούν σιλό και να προωθούν τη συνεργασία μεταξύ ομάδων, γραφείων και τοποθεσιών. Οι επαγγελματίες τεχνικής υποστήριξης μπορούν να χρησιμοποιήσουν την απομακρυσμένη πρόσβαση για να συνδεθούν με τους υπολογιστές των χρηστών από απομακρυσμένες τοποθεσίες, για να τους βοηθήσουν να επιλύσουν προβλήματα με τα συστήματα ή το λογισμικό τους. Μια κοινή μέθοδος παροχής απομακρυσμένης πρόσβασης είναι μέσω μιας σύνδεσης εικονικού ιδιωτικού δικτύου (VPN). Ένα VPN δημιουργεί μια ασφαλή και κρυπτογραφημένη σύνδεση μέσω ενός λιγότερο ασφαλούς δικτύου, όπως το διαδίκτυο. Η τεχνολογία VPN αναπτύχθηκε για να επιτρέπει στους απομακρυσμένους χρήστες και στα υποκαταστήματα να συνδέονται με ασφάλεια σε εταιρικές εφαρμογές και άλλους πόρους (Rosencrance, 2021).

2.1.2. Η λειτουργία της απομακρυσμένης πρόσβασης

Η απομακρυσμένη πρόσβαση επιτυγχάνεται με τον συνδυασμό λογισμικού, υλικού και συνδεσιμότητας δικτύου. Για παράδειγμα, η παραδοσιακή απομακρυσμένη πρόσβαση πριν από την ευρεία διαθεσιμότητα της συνδεσιμότητας στο Διαδίκτυο επιτυγχανόταν χρησιμοποιώντας λογισμικό εξομοίωσης τερματικού που έλεγχε την πρόσβαση μέσω ενός μόντεμ υλικού συνδεδεμένου σε ένα τηλεφωνικό δίκτυο.

Σήμερα, η απομακρυσμένη πρόσβαση επιτυγχάνεται πιο συχνά χρησιμοποιώντας:

- Λογισμικό: Χρησιμοποιώντας μια ασφαλή λύση λογισμικού όπως ένα VPN.
- Υλικό: Με τη σύνδεση κεντρικών υπολογιστών μέσω ενσύρματης διεπαφής δικτύου ή διεπαφής δικτύου Wi-Fi.
- Δίκτυο: Με σύνδεση μέσω Διαδικτύου.

Τα VPN απομακρυσμένης πρόσβασης συνδέουν μεμονωμένους χρήστες σε ιδιωτικά δίκτυα. Με ένα VPN απομακρυσμένης πρόσβασης, κάθε χρήστης χρειάζεται έναν πελάτη VPN ικανό να συνδεθεί στον διακομιστή VPN του ιδιωτικού δικτύου. Όταν ένας χρήστης είναι συνδεδεμένος στο δίκτυο μέσω ενός προγράμματος - πελάτη VPN, το λογισμικό κρυπτογραφεί την κίνηση πριν την παραδώσει μέσω του Διαδικτύου. Ο διακομιστής VPN, ή πύλη, βρίσκεται στην άκρη του στοχευμένου δικτύου και αποκρυπτογραφεί τα δεδομένα και τα στέλνει στον κατάλληλο κεντρικό υπολογιστή εντός του ιδιωτικού δικτύου.

Ένας υπολογιστής πρέπει να διαθέτει λογισμικό που του επιτρέπει να συνδέεται και να επικοινωνεί με ένα σύστημα ή έναν πόρο που φιλοξενείται από την υπηρεσία απομακρυσμένης πρόσβασης του οργανισμού. Μόλις ο υπολογιστής ενός χρήστη συνδεθεί στον απομακρυσμένο κεντρικό υπολογιστή, μπορεί να εμφανίσει ένα παράθυρο με την επιφάνεια εργασίας του υπολογιστή προορισμού.

Οι επιχειρήσεις μπορούν να χρησιμοποιούν απομακρυσμένους επιτραπέζιους υπολογιστές, για να επιτρέπουν στους χρήστες να συνδέονται με τις εφαρμογές και τα δίκτυά τους εξ αποστάσεως. Οι απομακρυσμένοι επιτραπέζιοι υπολογιστές χρησιμοποιούν λογισμικό εφαρμογής -- μερικές φορές ενσωματωμένο στο λειτουργικό σύστημα (OS) του απομακρυσμένου κεντρικού υπολογιστή -- που

επιτρέπει στις εφαρμογές να εκτελούνται απομακρυσμένα σε διακομιστή δικτύου και να εμφανίζονται τοπικά ταυτόχρονα.

Οι χρήστες μπορούν να έχουν πρόσβαση με ασφάλεια σε εφαρμογές και διακομιστές εσωτερικού χώρου και cloud από οπουδήποτε, σε οποιαδήποτε συσκευή με μια ποικιλία μεθόδων ελέγχου ταυτότητας, συμπεριλαμβανομένης της απομακρυσμένης απλής σύνδεσης, η οποία παρέχει στους χρήστες εύκολη και ασφαλή πρόσβαση στις εφαρμογές που χρειάζονται χωρίς να διαμορφώνουν VPN ή να τροποποιούν το τείχος προστασίας πολιτικές. Επιπλέον, οι οργανισμοί μπορούν να χρησιμοποιήσουν τον έλεγχο ταυτότητας πολλαπλών παραγόντων για να επαληθεύσουν την ταυτότητα ενός χρήστη, συνδυάζοντας πολλαπλά διαπιστευτήρια μοναδικά για ένα άτομο.

2.1.3. Οι τύποι της απομακρυσμένης πρόσβασης

Παραδοσιακά, οι επιχειρήσεις χρησιμοποιούσαν μόντεμ και τεχνολογίες dial - up για να επιτρέπουν στους υπαλλήλους να συνδέονται σε δίκτυα γραφείου μέσω τηλεφωνικών δικτύων συνδεδεμένων σε διακομιστές απομακρυσμένης πρόσβασης. Οι συσκευές που είναι συνδεδεμένες σε δίκτυα τηλεφώνου, χρησιμοποιούν αναλογικά μόντεμ για να καλούν εκχωρημένους αριθμούς τηλεφώνου για να πραγματοποιούν συνδέσεις και να στέλνουν ή να λαμβάνουν μηνύματα.

Η ευρυζωνική σύνδεση παρέχει στους απομακρυσμένους χρήστες επιλογές σύνδεσης υψηλής ταχύτητας σε επιχειρηματικά δίκτυα και στο διαδίκτυο. Υπάρχουν διάφοροι τύποι ευρυζωνικών συνδέσεων, συμπεριλαμβανομένων των εξής:

- Η καλωδιακή ευρυζωνική σύνδεση μοιράζεται το εύρος ζώνης σε πολλούς χρήστες και, ως εκ τούτου, οι ρυθμοί δεδομένων ανάντη μπορεί να είναι αργοί κατά τις ώρες υψηλής χρήσης σε περιοχές με πολλούς συνδρομητές.
- Η ευρυζωνική γραμμή DSL (Digital Subscriber Line) παρέχει δικτύωση υψηλής ταχύτητας μέσω τηλεφωνικού δικτύου, χρησιμοποιώντας τεχνολογία ευρυζωνικού μόντεμ. Ωστόσο, το DSL λειτουργεί μόνο σε περιορισμένη φυσική απόσταση και ενδέχεται να μην είναι διαθέσιμο σε ορισμένες περιοχές εάν η τοπική τηλεφωνική υποδομή δεν υποστηρίζει τεχνολογία DSL.

- Οι υπηρεσίες κινητής τηλεφωνίας διαδικτύου είναι προσβάσιμες από κινητές συσκευές μέσω ασύρματης σύνδεσης από οποιαδήποτε τοποθεσία όπου υπάρχει διαθέσιμο δίκτυο κινητής τηλεφωνίας.
- Οι δορυφορικές υπηρεσίες Διαδικτύου χρησιμοποιούν τηλεπικοινωνιακούς δορυφόρους για να παρέχουν στους χρήστες πρόσβαση στο Διαδίκτυο σε περιοχές όπου δεν είναι διαθέσιμη η επίγεια πρόσβαση στο Διαδίκτυο, καθώς και για προσωρινές κινητές εγκαταστάσεις.
- Η ευρυζωνική τεχνολογία οπτικών ινών επιτρέπει στους χρήστες να μεταφέρουν μεγάλες ποσότητες δεδομένων γρήγορα και απρόσκοπτα.

2.1.4. Τα πρωτόκολλα της απομακρυσμένης πρόσβασης

Τα κοινά πρωτόκολλα απομακρυσμένης πρόσβασης και VPN περιλαμβάνουν τα ακόλουθα:

- Το Πρωτόκολλο Point-to-Point (PPP) επιτρέπει στους κεντρικούς υπολογιστές να δημιουργήσουν μια άμεση σύνδεση μεταξύ δύο τελικών σημείων.
- Το IPsec -- Internet Protocol Security -- είναι ένα σύνολο πρωτοκόλλων ασφαλείας που χρησιμοποιούνται για την ενεργοποίηση των υπηρεσιών ελέγχου ταυτότητας και κρυπτογράφησης για την ασφάλεια της μεταφοράς πακέτων IP μέσω του Διαδικτύου.
- Το Point-to-Point Tunneling (PPTP) είναι ένα από τα παλαιότερα πρωτόκολλα για την υλοποίηση VPN. Ωστόσο, με τα χρόνια, έχει αποδειχθεί ευάλωτο σε πολλούς τύπους επιθέσεων. Αν και το PPTP δεν είναι ασφαλές, παραμένει σε ορισμένες περιπτώσεις.
- Το Layer Two Tunneling Protocol (L2TP) είναι ένα πρωτόκολλο VPN που δεν προσφέρει κρυπτογράφηση ή κρυπτογραφικό έλεγχο ταυτότητας για την κίνηση που διέρχεται από τη σύνδεση. Ως αποτέλεσμα, συνήθως συνδέεται με το IPsec, το οποίο παρέχει αυτές τις υπηρεσίες.
- Το Remote Authentication Dial-In User Service (RADIUS) είναι ένα πρωτόκολλο που αναπτύχθηκε το 1991 και δημοσιεύτηκε ως προδιαγραφή κομματιού Internet Standard το 2000, για να επιτρέπει στους διακομιστές

απομακρυσμένης πρόσβασης να επικοινωνούν με έναν κεντρικό διακομιστή για τον έλεγχο ταυτότητας των χρηστών μέσω τηλεφώνου και την εξουσιοδότηση της πρόσβασής τους στο ζητούμενο σύστημα ή υπηρεσία.

- Terminal Access Controller Access Control System (TACACS) είναι ένα πρωτόκολλο απομακρυσμένου ελέγχου ταυτότητας που ήταν αρχικά κοινό στα δίκτυα Unix, που επιτρέπει σε έναν διακομιστή απομακρυσμένης πρόσβασης να προωθήσει τον κωδικό πρόσβασης ενός χρήστη σε έναν διακομιστή ελέγχου ταυτότητας για να καθορίσει εάν θα πρέπει να επιτρέπεται η πρόσβαση σε ένα δεδομένο σύστημα. Το TACACS + είναι ένα ξεχωριστό πρωτόκολλο που έχει σχεδιαστεί για να χειρίζεται τον έλεγχο ταυτότητας και την εξουσιοδότηση και να λαμβάνει υπόψη την πρόσβαση διαχειριστή σε συσκευές δικτύου, όπως δρομολογητές και μεταγωγείς.

2.2. Κερκόπορτες (Backdoors)

2.2.1. Η έννοια της κερκόπορτας

Η κερκόπορτα είναι ένα μέσο πρόσβασης σε ένα σύστημα υπολογιστή ή σε κρυπτογραφημένα δεδομένα που παρακάμπτουν τους συνήθεις μηχανισμούς ασφαλείας του συστήματος.

Ένας προγραμματιστής μπορεί να δημιουργήσει μια κερκόπορτα έτσι ώστε να είναι δυνατή η πρόσβαση σε μια εφαρμογή ή λειτουργικό σύστημα για αντιμετώπιση προβλημάτων ή άλλους σκοπούς. Ωστόσο, οι εισβολείς χρησιμοποιούν συχνά κερκόπορτες που εντοπίζουν ή εγκαθιστούν οι ίδιοι ως μέρος ενός exploit. Σε ορισμένες περιπτώσεις, ένα σκουλήκι ή ένας ιός έχει σχεδιαστεί για να εκμεταλλεύεται μια κερκόπορτα που δημιουργήθηκε από μια προηγούμενη επίθεση.

Είτε είναι εγκατεστημένο ως εργαλείο διαχείρισης, είτε ως μέσο επίθεσης, είτε ως μηχανισμός που επιτρέπει στην κυβέρνηση να έχει πρόσβαση σε κρυπτογραφημένα δεδομένα, μια κερκόπορτα αποτελεί κίνδυνο για την ασφάλεια, επειδή υπάρχουν πάντα παράγοντες απειλών που αναζητούν οποιαδήποτε ευπάθεια προς εκμετάλλευση.

Οι χάκερ μπορούν να εγκαταστήσουν μία κερκόπορτα σε μία συσκευή χρησιμοποιώντας κακόβουλο λογισμικό, εκμεταλλευόμενοι τα τρωτά σημεία του

λογισμικού ή ακόμα και εγκαθιστώντας απευθείας μία κερκόπορτα στο υλικό / λογισμικό της συσκευής. Μπορούν να χρησιμοποιήσουν κερκόπορτες για διάφορους λόγους, όπως:

- Επιτήρηση
- Κλοπή δεδομένων
- Cryptojacking
- Σαμποτάζ
- Επίθεση κακόβουλου λογισμικού

Κανείς δεν έχει ανοσία στο backdoor hacking και οι χάκερ εφευρίσκουν συνεχώς νέες μεθόδους και αρχεία κακόβουλου λογισμικού για να αποκτήσουν πρόσβαση στις συσκευές των χρηστών.

2.2.2. Η λειτουργία της κερκόπορτας

Οι κερκόπορτες μπορεί να διαφέρουν πολύ. Ορισμένες, για παράδειγμα, τοποθετούνται από νόμιμους προμηθευτές, ενώ άλλες εισάγονται ακούσια ως αποτέλεσμα σφαλμάτων προγραμματισμού. Οι προγραμματιστές χρησιμοποιούν μερικές φορές κερκόπορτες κατά τη διαδικασία ανάπτυξης, οι οποίες στη συνέχεια δεν αφαιρούνται από τον κώδικα παραγωγής.

Οι κερκόπορτες τοποθετούνται επίσης συνήθως μέσω κακόβουλου λογισμικού. Μια λειτουργική μονάδα κακόβουλου λογισμικού μπορεί να λειτουργεί ως κερκόπορτα η ίδια ή μπορεί να λειτουργήσει ως κερκόπορτα πρώτης γραμμής, πράγμα που σημαίνει ότι λειτουργεί ως πλατφόρμα για τη λήψη άλλων λειτουργικών μονάδων κακόβουλου λογισμικού που έχουν σχεδιαστεί για να εκτελούν την πραγματική επίθεση.

Οι αλγόριθμοι κρυπτογράφησης και τα πρωτόκολλα δικτύωσης μπορεί επίσης, τουλάχιστον δυνητικά, να περιέχουν κερκόπορτες. Για παράδειγμα, το 2016, πολλοί ερευνητές περιέγραψαν πώς οι πρώτοι αριθμοί που χρησιμοποιούνται σε αλγόριθμους κρυπτογράφησης θα μπορούσαν να δημιουργηθούν με τέτοιο τρόπο που θα μπορούσαν να επιτρέψουν σε έναν αντίπαλο να συνυπολογίσει τους πρώτους αριθμούς - και έτσι να σπάσει την κρυπτογράφηση - των αλγορίθμων κρυπτογράφησης που προηγουμένως θεωρούνταν ασφαλείς.

Προκειμένου οι εγκληματίες του κυβερνοχώρου να εγκαταστήσουν επιτυχώς μία κερκόπορτα σε συσκευή, πρέπει πρώτα να αποκτήσουν πρόσβαση στη συσκευή, είτε μέσω φυσικής πρόσβασης, είτε μέσω επίθεσης κακόβουλου λογισμικού είτε εκμεταλλευόμενοι μια ευπάθεια συστήματος. Μερικές από τις πιο κοινές ευπάθειες που στοχεύουν οι χάκερ είναι οι εξής:

- **Ανοιχτές θύρες:** Μια ανοιχτή θύρα σε ένα δίκτυο δέχεται κίνηση από απομακρυσμένες τοποθεσίες και ως αποτέλεσμα, μπορεί να δημιουργήσει ένα αδύναμο σημείο που μπορούν να εκμεταλλευτούν οι χάκερ. Οι χάκερ στοχεύουν συνήθως θύρες που δεν χρησιμοποιούνται — αυτό τους επιτρέπει να εγκαταστήσουν κερκόπορτες που έχουν πρόσβαση στη συσκευή του χρήστη χωρίς κανένα λογισμικό να ειδοποιεί για την εισβολή. Αυτό δεν θα πρέπει να είναι πρόβλημα για τους περισσότερους οικιακούς χρήστες, επειδή οι θύρες των δρομολογητών είναι κλειστές από προεπιλογή. Ωστόσο, οι γνώστες της τεχνολογίας και οι ιδιοκτήτες μικρών επιχειρήσεων πρέπει να είναι πολύ προσεκτικοί σχετικά με το ποιες θύρες διατηρούν ανοιχτές και τις ευπάθειες που δημιουργούν αυτές οι ανοιχτές θύρες. Πολλοί επαγγελματίες πληροφορικής χρησιμοποιούν το Shodan για να διατηρήσουν τα δίκτυά τους όσο το δυνατόν ασφαλέστερα.
- **Κρυφές / νόμιμες κερκόπορτες:** Οι κρυφές κερκόπορτες εγκαθίστανται σκόπιμα από τους προγραμματιστές λογισμικού για την παροχή απομακρυσμένης πρόσβασης για την εκτέλεση νόμιμων λειτουργιών, όπως η υποστήριξη πελατών ή η επίλυση προβλημάτων του λογισμικού. Ωστόσο, οι περισσότεροι αξιόπιστοι κατασκευαστές περιλαμβάνουν μόνο κερκόπορτες με ακραίες προστασίες ασφαλείας για την αποφυγή κακών παιχνιδιών — όμως μια έκθεση του 2020 έδειξε ότι ο Κινέζος κατασκευαστής Xiongmai είχε αποστείλει χιλιάδες κάμερες, DVR, NVR και άλλα προϊόντα με backdoors υλικού που θα μπορούσαν να επιτρέψουν σε οποιονδήποτε (για παράδειγμα CCP), για απομακρυσμένη πρόσβαση, λειτουργία και παρακολούθηση των χρηστών στις συσκευές τους.
- **Αδύναμοι κωδικοί πρόσβασης:** Ένας αδύναμος κωδικός πρόσβασης μπορεί να δώσει στους χάκερ άμεση πρόσβαση στους λογαριασμούς — αφού οι χάκερ έχουν σπάσει έναν λογαριασμό, είναι πολύ εύκολο για αυτούς να έχουν πρόσβαση στους άλλους λογαριασμούς και τις συσκευές. Ένα καλό

παράδειγμα του τρόπου με τον οποίο οι χάκερ αξιοποιούν τους προεπιλεγμένους κωδικούς πρόσβασης είναι το botnet Mirai του 2016, που επηρέασε 2.5 εκατομμύρια συσκευές IoT σε όλο τον κόσμο. Το Mirai σχεδιάστηκε για να σαρώνει το Διαδίκτυο για συσκευές IoT χρησιμοποιώντας προεπιλεγμένους κωδικούς πρόσβασης, χακάροντας αυτές τις συσκευές και υποδουλώνοντάς τις στο botnet, απλά εισάγοντας τους κατάλληλους κωδικούς πρόσβασης.

- Μη ενημερωμένο λογισμικό: Οι χάκερ χρησιμοποιούν επιθέσεις για να εγκαταστήσουν κακόβουλο λογισμικό (συμπεριλαμβανομένων των κερκόπορτων) σε συσκευές χρηστών. Αλλά αν διατηρείται ενημερωμένο όλο το λογισμικό, πιθανότατα ο χρήστης δεν θα πέσει θύμα μιας επίθεσης εκμετάλλευσης.
- Αδύναμα τείχη προστασίας: Οι περισσότερες επιθέσεις κακόβουλου λογισμικού εξαρτώνται από σφάλμα του ίδιου του χρήστη — με διαφημίσεις του τύπου ‘ θέλετε να λάβετε ένα δωρεάν πρόγραμμα ή torrent την πιο πρόσφατη ταινία της Marvel ή κάνετε κλικ σε έναν σύνδεσμο για μια μεγάλη πώληση ’ και, στη συνέχεια, ξαφνικά ο χρήστης καταλήγει να δίνει τις πληροφορίες του ή να εγκαθιστά ένα αρχείο κακόβουλου λογισμικού.

Πραγματοποιούνται στοχευμένες επιθέσεις που εκμεταλλεύονται ευπάθειες του λογισμικού (συνήθως σε λογισμικό που αντιμετωπίζει τον ιστό, όπως προγράμματα περιήγησης, Adobe Flash, Java, κ.λπ.), προκειμένου να παρέχουν στους χάκερ πρόσβαση στο σύστημά σας. Υπάρχουν κακόβουλοι ιστότοποι και διαφημίσεις που σαρώνουν τον υπολογιστή για ευπάθειες λογισμικού και τα χρησιμοποιούν για να κλέψουν τα δεδομένα, να διαρρήξουν το δίκτυο ή να εγκαταστήσουν μια κερκόπορτα στη συσκευή (Martens, 2022).

Το 2014, μια προσέγγιση για τη δημιουργία τυχαίων αριθμών που ονομάζεται Dual_EC_DRBG (Dual Elliptic Curve Deterministic Random Bit Generator), βρέθηκε να έχει ένα σφάλμα σε αυτήν που έκανε τους τυχαίους αριθμούς που προέκυψαν κάπως προβλέψιμους. Η συναίνεση της κοινότητας ασφαλείας ήταν ότι η NSA επέτρεψε τη χρήση του προτύπου, παρόλο που γνώριζε ότι υπήρχε μια αδυναμία, ώστε να μπορούν να το χρησιμοποιήσουν ως κερκόπορτα.

Στην ασφάλεια του κυβερνοχώρου, μια κερκόπορτα είναι οτιδήποτε μπορεί να επιτρέψει σε έναν εξωτερικό χρήστη να εισέλθει στη συσκευή χωρίς τη γνώση ή την άδειά του. Οι κερκόπορτες μπορούν να εγκατασταθούν σε δύο διαφορετικά μέρη του συστήματος:

- Firmware / hardware: Φυσικές αλλαγές που παρέχουν απομακρυσμένη πρόσβαση στη συσκευή σας.
- Software: Αρχεία κακόβουλου λογισμικού που κρύβουν τα ίχνη τους, ώστε το λειτουργικό σύστημα του χρήστη να μην γνωρίζει ότι κάποιος άλλος έχει πρόσβαση στη συσκευή του.

Η κερκόπορτα μπορεί να εγκατασταθεί από προγραμματιστές λογισμικού καθώς και υλικού για σκοπούς απομακρυσμένης τεχνικής υποστήριξης, αλλά στις περισσότερες περιπτώσεις, οι κερκόπορτες εγκαθίστανται είτε από εγκληματίες του κυβερνοχώρου, είτε από παρεμβατικές κυβερνήσεις, για να τους βοηθήσουν να αποκτήσουν πρόσβαση σε μια συσκευή, ένα δίκτυο ή μια εφαρμογή λογισμικού.

Οποιοδήποτε κακόβουλο λογισμικό παρέχει πρόσβαση στους χάκερ, μπορεί να θεωρηθεί ως κερκόπορτα και σε αυτό περιλαμβάνονται (Martens, 2022):

- Rootkits
- Trojans
- Spyware
- Cryptojackers
- Keyloggers
- Worms
- Ransomware

Μερικά παραδείγματα των διαφορετικών ειδών κερκόπορτων που χρησιμοποιούνται συχνά, είναι τα εξής:

1. Trojans: Τα Trojans είναι αρχεία κακόβουλου λογισμικού που προσποιούνται ότι είναι νόμιμα αρχεία, για να αποκτήσουν πρόσβαση στη συσκευή. Μόλις ο χρήστης κάνει κλικ στο "Να επιτρέπεται η εισαγωγή – προγράμματος - εδώ να κάνει αλλαγές στη συσκευή σας;" κουμπί στον υπολογιστή του, ο Trojan μπορεί στη συνέχεια να εγκατασταθεί στη συσκευή. Οι Trojan κερκόπορτες μπορούν να

επιτρέψουν στους χρήστες να έχουν πρόσβαση στα αρχεία και τα προγράμματα του χρήστη ή να εγκαταστήσουν πιο σοβαρά αρχεία κακόβουλου λογισμικού στη συσκευή.

2. Rootkits: Τα Rootkits είναι προηγμένες απειλές κακόβουλου λογισμικού που μπορούν να αποκρύψουν τις δραστηριότητές τους από ένα λειτουργικό σύστημα, έτσι ώστε το λειτουργικό σύστημα να παρέχει δικαιώματα ασφαλείας (root access) στο rootkit. Τα Rootkits μπορούν να επιτρέψουν σε έναν χάκερ να αποκτήσει απομακρυσμένη πρόσβαση στη συσκευή, να αλλάξει τα αρχεία που εμπεριέχονται σε αυτό, να παρατηρήσει τη δραστηριότητα του χρήστη και να σαμποτάρει το σύστημά του. Τα Rootkits μπορούν να λάβουν τη μορφή είτε λογισμικού, είτε ακόμα και φυσικά τροποποιημένα τσιπ υπολογιστή.
3. Hardware backdoors: Οι κερκόπορτες υλικού είναι τροποποιημένα τσιπ υπολογιστή ή άλλο firmware / hardware, που παρέχουν πρόσβαση σε μη χρήστες σε μια συσκευή. Αυτό μπορεί να περιλαμβάνει τηλέφωνα, συσκευές IoT όπως θερμοστάτες και συστήματα οικιακής ασφάλειας, δρομολογητές και υπολογιστές. Οι κερκόπορτες υλικού μπορούν να επικοινωνούν δεδομένα του χρήστη, να παρέχουν απομακρυσμένη πρόσβαση ή να χρησιμοποιούνται για επιτήρηση. Οι κερκόπορτες υλικού μπορούν να αποσταλούν μαζί με προϊόντα (είτε από έναν αδιάστακτο κατασκευαστή, είτε για κάποιο καλό σκοπό), αλλά μπορούν επίσης να εγκατασταθούν φυσικά σε περίπτωση κλοπής μιας συσκευής.
4. Cryptographic backdoors: Οι κρυπτογραφικές κερκόπορτες είναι ουσιαστικά ένα «κύριο κλειδί» που μπορεί να ξεκλειδώσει κάθε κομμάτι κρυπτογραφημένων δεδομένων που χρησιμοποιεί ένα συγκεκριμένο πρωτόκολλο κρυπτογράφησης. Τα πρότυπα κρυπτογράφησης όπως το AES χρησιμοποιούν κρυπτογράφηση από άκρο σε άκρο, έτσι ώστε μόνο τα μέρη που έχουν ανταλλάξει ένα τυχαία δημιουργημένο κρυπτογραφικό κλειδί να μπορούν να αποκρυπτογραφήσουν τις πληροφορίες που μοιράζονται. Οι κερκόπορτες είναι ένας τρόπος να διακοπεί αυτή η ασφαλής συνομιλία, χειραγωγώντας τα πολύπλοκα μαθηματικά ενός συγκεκριμένου κρυπτογραφικού πρωτοκόλλου για να δοθεί σε έναν εξωτερικό χρήστη πρόσβαση σε όλα τα κρυπτογραφημένα δεδομένα που μοιράζονται μεταξύ των μερών.

2.2.3. Ανίχνευση και πρόληψη

Οι κερκόπορτες μπορεί να είναι πολύ δύσκολο να εντοπιστούν και οι μέθοδοι ανίχνευσης ποικίλλουν σημαντικά ανάλογα με το λειτουργικό σύστημα του υπολογιστή. Σε ορισμένες περιπτώσεις, το λογισμικό προστασίας από κακόβουλο λογισμικό μπορεί να είναι ικανό να ανιχνεύει λογισμικό κερκόπορτας. Σε άλλες περιπτώσεις, οι επαγγελματίες ασφάλειας μπορεί να χρειαστεί να χρησιμοποιήσουν εξειδικευμένα εργαλεία για τον εντοπισμό κερκόπορτων ή να χρησιμοποιήσουν ένα εργαλείο παρακολούθησης πρωτοκόλλου για να επιθεωρήσουν πακέτα δικτύου.

Υπάρχουν πολλές διαφορετικές στρατηγικές για την αποφυγή επιθέσεων από κερκόπορτα. Αρχικά, οι οργανισμοί πρέπει να τηρούν τις βέλτιστες πρακτικές ασφάλειας, όπως η αποφυγή μη αξιόπιστου λογισμικού και η διασφάλιση ότι κάθε συσκευή προστατεύεται από τείχος προστασίας. Τα τείχη προστασίας εφαρμογών μπορούν επίσης να βοηθήσουν στην αποτροπή επιθέσεων κερκόπορτας, καθώς περιορίζουν την κίνηση που μπορεί να ρέει σε ανοιχτές θύρες. Είναι επίσης σημαντικό να παρακολουθείται η κυκλοφορία του δικτύου για υπογραφές που μπορεί να υποδηλώνουν την παρουσία μιας κερκόπορτας.

Οι κερκόπορτες είναι δύσκολο να εντοπιστούν. Οι καθημερινοί χρήστες δεν μπορούν να ανακαλύψουν μια κερκόπορτα απλώς ανοίγοντας τη διαχείριση εργασιών. Ωστόσο, υπάρχουν μερικά εύκολα βήματα που μπορεί να παρακολουθεί κάποιος για να διατηρεί τη συσκευή ασφαλή από επιθέσεις ιών κερκόπορτων, όπως:

1. Χρήση ενός προγράμματος Antivirus

Να γίνεται πάντα χρήση προηγμένου λογισμικού προστασίας από ιούς που μπορεί να εντοπίσει και να αποτρέψει ένα ευρύ φάσμα κακόβουλου λογισμικού, συμπεριλαμβανομένων trojans, cryptojackers, spyware και rootkits. Ένα πρόγραμμα προστασίας από ιούς θα εντοπίσει τους ιούς της κερκόπορτας και θα τους εξαλείψει πριν μολύνουν τον υπολογιστή. Το καλό λογισμικό προστασίας από ιούς όπως το Norton 360 περιλαμβάνει επίσης εργαλεία, όπως παρακολούθηση Wi-Fi, προηγμένο τείχος προστασίας, προστασία ιστού και παρακολούθηση απορρήτου μικροφώνου και κάμερας web, για να διασφαλιστεί ότι ο χρήστης είναι όσο το δυνατόν πιο ασφαλείς στο διαδίκτυο.

Για παράδειγμα, το Norton 360 διαθέτει προστασία σε πραγματικό χρόνο (η οποία μπορεί να αποτρέψει την εγκατάσταση backdoor malware στον υπολογιστή), ένα τείχος προστασίας (το οποίο μπορεί να εντοπίσει και να αποτρέψει την ανεπιθύμητη κυκλοφορία ιστού), έναν διαχειριστή κωδικών πρόσβασης (που μπορεί να αποτρέψει την καταστροφή των λογαριασμών), έναν σαρωτή ευπάθειας (που μπορεί να πει εάν ο χρήστης χρειάζεται ενημερώσεις λογισμικού) και παρακολούθηση σκοτεινού ιστού (που μπορεί να προειδοποιήσει εάν τα δεδομένα έχουν παραβιαστεί).

2. Λήψη με προσοχή

Οι κερκόπορτες συχνά συνοδεύονται από φαινομενικά νόμιμο ελεύθερο λογισμικό, αρχεία και εφαρμογές. Κατά τη λήψη οποιουδήποτε αρχείου από το Διαδίκτυο, πρέπει να γίνεται έλεγχος αν πραγματοποιείται η λήψη μόνο του αρχείου που επιθυμεί ο χρήστης ή αν έρχονται κάποια δυσάρεστα block κατά την λήψη. Ακόμη και ένα αρχείο που συμπεριφέρεται όπως το αρχείο που γίνεται λήψη μπορεί να είναι trojan. Πρέπει να πραγματοποιείται πάντα λήψη από επίσημους ιστότοπους, να αποφεύγονται οι πειρατικοί ιστότοποι και να εγκαθίσταται ένα πρόγραμμα προστασίας από ιούς με προστασία σε πραγματικό χρόνο, που μπορεί να επισημαίνει αρχεία κακόβουλου λογισμικού προτού καν τα κατεβάσετε στο σύστημα.

3. Χρήση ενός τείχους προστασίας

Τα τείχη προστασίας είναι απαραίτητα για την προστασία κατά της κερκόπορτας — παρακολουθούν όλη την εισερχόμενη και εξερχόμενη κίνηση στη συσκευή. Εάν κάποιος εκτός του εγκεκριμένου δικτύου προσπαθεί να εισέλθει στη συσκευή του χρήστη, το τείχος προστασίας θα τον αποκλείσει και εάν μια εφαρμογή στη συσκευή προσπαθεί να στείλει δεδομένα σε μια άγνωστη τοποθεσία δικτύου, το τείχος προστασίας θα αποκλείσει και αυτήν την εφαρμογή.

Τα προηγμένα τείχη προστασίας μπορούν να ανιχνεύσουν μη εξουσιοδοτημένη κίνηση στην κερκόπορτα ακόμα και όταν ο εντοπισμός κακόβουλου λογισμικού της συσκευής έχει εξαπατηθεί. Τα Windows και το macOS έχουν και τα δύο αρκετά αξιοπρεπή ενσωματωμένα τείχη προστασίας, αλλά δεν είναι αρκετά καλά. Υπάρχουν μερικά προγράμματα προστασίας από ιούς με καλά τείχη προστασίας (το McAfee έχει εξαιρετική προστασία δικτύου) και μπορεί επίσης να γίνει αγορά ενός έξυπνου τείχους

προστασίας, το οποίο είναι μια φυσική συσκευή που συνδέεται στο δρομολογητή σας για να διατηρείται το δίκτυό σας όσο το δυνατόν ασφαλέστερο.

4. Χρήση ενός διαχειριστή κωδικών πρόσβασης

Οι διαχειριστές κωδικών πρόσβασης δημιουργούν και αποθηκεύουν πληροφορίες σύνδεσης για όλους τους λογαριασμούς και βοηθούν ακόμη και την αυτόματη σύνδεση σε αυτούς. Όλες αυτές οι πληροφορίες είναι κρυπτογραφημένες με ασφάλεια, χρησιμοποιώντας κρυπτογράφηση AES 256-bit και κλειδώνονται πίσω από έναν κύριο κωδικό πρόσβασης. Οι προηγμένοι διαχειριστές κωδικών πρόσβασης, όπως το Dashlane, μπορούν ακόμη και να βελτιώσουν την ασφάλεια του θησαυροφυλακίου κωδικών πρόσβασης χρησιμοποιώντας βιομετρικά εργαλεία σύνδεσης ή εργαλεία 2FA, όπως γεννήτριες TOTP και διακριτικά USB. Επειδή δημιουργούν τυχαίους, σύνθετους κωδικούς πρόσβασης, οι διαχειριστές κωδικών πρόσβασης καθιστούν πολύ πιο δύσκολο για τους χάκερ να εισέλθουν ή να εξαπλωθούν σε όλο το δίκτυο σε περίπτωση που εγκατασταθεί μια κερκόπορτα στο σύστημα.

5. Παρακολούθηση για Ενημερώσεις / Επιδιορθώσεις ασφαλείας

Οι επιθέσεις Zero-day είναι αρκετά σπάνιες και οι περισσότεροι χάκερ απλώς ανακυκλώνουν τα ίδια exploits και το ίδιο κακόβουλο λογισμικό, επειδή είναι φθηνό, πιο εύκολο για αυτούς να το υλοποιήσουν κι επιπλέον είναι αποτελεσματικό και λειτουργεί σχεδόν σε κάθε περίπτωση. Ένας στους τρεις επαγγελματίες πληροφορικής (με ποσοστό 34%) στην Ευρώπη, παραδέχτηκε ότι η εταιρεία του είχε παραβιαστεί ως αποτέλεσμα μιας μη επιδιορθωμένης ευπάθειας.

Οι προγραμματιστές λογισμικού δημοσιεύουν συχνά νέες ενημερώσεις κώδικα για να διορθώσουν τα τρωτά σημεία στο λογισμικό τους και δεν είναι δύσκολο να εγκαταστήσουν αυτές τις ενημερώσεις. Πολλά προγράμματα περιλαμβάνουν ακόμη και μια επιλογή αυτόματης ενημέρωσης. Εάν είναι χρήστης macOS ή Windows, μπορεί να γίνει μετάβαση στις ρυθμίσεις και να ενεργοποιηθούν "Αυτόματες Ενημερώσεις" — είναι ιδιαίτερα σημαντικό να διατηρείται το λειτουργικό σύστημα ενημερωμένο, επειδή οι κερκόπορτες εξαρτώνται από την εξαπάτηση του λειτουργικού συστήματος.

2.2.4. Διάσημες επιθέσεις σε κερκόπορτα

Υπήρξαν αρκετές επιθέσεις με κερκόπορτα υψηλού προφίλ που έχουν συμβεί τις τελευταίες δεκαετίες. Στα τέλη του 2020, μια εταιρεία κυβερνοασφάλειας, η οποία ονομάζεται FireEye, ανακάλυψε μια εξαιρετικά σοβαρή κερκόπορτα κρυμμένη σε ενημερώσεις για το λογισμικό διαχείρισης δικτύου Orion της SolarWinds. Οι επιτιθέμενοι, οι οποίοι πιστεύεται ότι προέρχονται από το επίπεδο του έθνους - κράτους, χρησιμοποίησαν το SolarWinds για να διευκολύνουν μια επίθεση island hopping, που εγκατέστησε κακόβουλο λογισμικό σε δίκτυα πελατών Orion, προκειμένου να συγκεντρώσει πληροφορίες. Η Υπηρεσία Κυβερνοασφάλειας και Ασφάλειας Υποδομών των Ηνωμένων Πολιτειών (CISA) πιστεύει ότι η επίθεση ξεκίνησε ήδη από τον Μάρτιο του 2020 και ότι στην πραγματικότητα δεν στοχοποιήθηκαν όλοι οι παραβιασμένοι οργανισμοί από τον εισβολέα για επακόλουθες ενέργειες.

Στις αρχές του 2021, μια ολλανδική εταιρεία κυβερνοασφάλειας ανακάλυψε έναν μυστικό λογαριασμό backdoor με σκληρό κώδικα σε τείχη προστασίας Zyxel και ελεγκτές σημείων πρόσβασης (AP). Ο μυστικός λογαριασμός επέτρεπε στους εισβολείς να δώσουν στους εαυτούς τους διαχειριστικά προνόμια, συμπεριλαμβανομένης της δυνατότητας αλλαγής των ρυθμίσεων του τείχους προστασίας και της παρακολούθησης της κυκλοφορίας. Το backdoor εκμεταλλεύτηκε μια ευπάθεια στα διαπιστευτήρια που χρησιμοποιούνται για την ενημέρωση του υλικολογισμικού του τείχους προστασίας και του ελεγκτή AP.

Μια άλλη αξιοσημείωτη επίθεση ονομαζόταν Back Orifice. Το Back Orifice, το οποίο δημιουργήθηκε το 1999 από μια ομάδα χάκερ που αυτοαποκαλούνταν Cult of the Dead Cow, εκμεταλλεύτηκε τις ευπάθειες στο λειτουργικό σύστημα (OS) των Windows, για να εγκαταστήσει κερκόπορτες που επέτρεπαν τον απομακρυσμένο έλεγχο των υπολογιστών με Windows (Posey, 2021).

Τα backdoors δεν βασίζονται πάντα σε λογισμικό, ούτε δημιουργούνται πάντα από απατεώνες ομάδες χάκερ. Το 2013, το γερμανικό ειδησεογραφικό πρακτορείο Der Spiegel ανέφερε ότι η μονάδα Επιχειρήσεων Προσαρμοσμένης Πρόσβασης της NSA διατηρούσε έναν κατάλογο κερκόπορτων για εμφύτευση σε τείχη προστασίας, δρομολογητές και άλλες συσκευές που θα χρησιμοποιηθούν στο εξωτερικό. Η NSA

φέρεται να ενσωμάτωσε επίσης δυνατότητες backdoor σε μεμονωμένα στοιχεία υλικού, όπως σκληρούς δίσκους, ακόμη και καλώδια USB.

2.3. Κέντρο διαχείρισης δικτύου Botnet (Command and Control Center)

Ένας διακομιστής εντολών και ελέγχου (διακομιστής C&C), είναι ένας υπολογιστής που εκδίδει οδηγίες σε ψηφιακές συσκευές που έχουν μολυνθεί με rootkits ή άλλους τύπους κακόβουλου λογισμικού, όπως ransomware. Οι διακομιστές C&C μπορούν να χρησιμοποιηθούν για τη δημιουργία ισχυρών δικτύων μολυσμένων συσκευών ικανών να πραγματοποιούν επιθέσεις κατανομής άρνησης υπηρεσίας (DDoS), να κλέβουν δεδομένα, να διαγράφουν δεδομένα ή να κρυπτογραφούν δεδομένα, προκειμένου να πραγματοποιήσουν ένα σχέδιο εκβιασμού.

Στο παρελθόν, ένας διακομιστής C&C ήταν συχνά υπό τον φυσικό έλεγχο ενός εισβολέα και μπορούσε να παραμείνει ενεργός για αρκετά χρόνια. Σήμερα, οι διακομιστές C&C έχουν γενικά μικρή διάρκεια ζωής. Συχνά κατοικούν σε νόμιμες υπηρεσίες cloud και χρησιμοποιούν αυτοματοποιημένους αλγόριθμους δημιουργίας τομέα (DGAs), για να καταστήσουν πιο δύσκολο τον εντοπισμό τους από τις αρχές επιβολής του νόμου και τους κυνηγούς κακόβουλου λογισμικού λευκού καπέλου.

Ένα κακόβουλο δίκτυο υπό τον έλεγχο ενός διακομιστή C&C ονομάζεται botnet και οι κόμβοι δικτύου που ανήκουν στο botnet αναφέρονται μερικές φορές ως ζόμπι. Σε ένα παραδοσιακό botnet, τα ρομπότ μολύνονται με έναν δούρειο ίππο και χρησιμοποιούν το Internet Relay Chat (IRC) για να επικοινωνούν με έναν κεντρικό διακομιστή C&C. Αυτά τα botnets χρησιμοποιούνταν συχνά για τη διανομή ανεπιθύμητου, κακόβουλου λογισμικού και τη συλλογή καταχρηστικών πληροφοριών, όπως αριθμούς πιστωτικών καρτών.

Οι δημοφιλείς τοπολογίες botnet περιλαμβάνουν:

- Τοπολογία αστεριών (Star topology): τα bots είναι οργανωμένα γύρω από έναν κεντρικό διακομιστή.
- Τοπολογία πολλών διακομιστών (Multi-server topology): υπάρχουν πολλοί διακομιστές C&C για πλεονασμό.

- Ιεραρχική τοπολογία (Hierarchical topology): πολλαπλοί διακομιστές C&C είναι οργανωμένοι σε κλιμακωτές ομάδες.
- Τυχαία τοπολογία (Random topology): οι υπολογιστές που έχουν επιλεγεί επικοινωνούν ως ομότιμο botnet (P2P botnet).

Δεδομένου ότι η επικοινωνία IRC χρησιμοποιείται συνήθως για την εντολή botnet, συχνά προφυλάσσεται. Αυτό έχει παρακινήσει την κίνηση για πιο κρυφούς τρόπους των διακομιστών C&C να εκδίδουν εντολές. Τα εναλλακτικά κανάλια που χρησιμοποιούνται για την εντολή botnet περιλαμβάνουν εικόνες JPG, αρχεία Microsoft Word και αναρτήσεις από εικονικούς λογαριασμούς LinkedIn ή Twitter.

2.4. Drive by Download Attack

Οι επιθέσεις Drive by Download αναφέρονται συγκεκριμένα σε κακόβουλα προγράμματα που εγκαθίστανται στις συσκευές — χωρίς τη συγκατάθεση του χρήστη. Αυτό περιλαμβάνει επίσης ακούσιες λήψεις οποιωνδήποτε αρχείων ή ομαδοποιημένου λογισμικού σε μια συσκευή υπολογιστή. Καλυμμένες σε όλες τις γωνίες του ιστού, αυτές οι επιθέσεις προκαλούν ακόμη και απολύτως νόμιμες τοποθεσίες να διαδώσουν αυτήν την απειλή. Οι δύο κύριες παραλλαγές των επιθέσεων Drive by Download είναι οι εξής:

1. Μη κακόβουλα δυνητικά ανεπιθύμητα προγράμματα ή εφαρμογές (PUP / PUA).
2. Επιθέσεις φορτωμένες με κακόβουλο λογισμικό.

Ενώ το πρώτο είναι καθαρό και ασφαλές, μπορεί να είναι adware στη χειρότερη περίπτωση. Οι ειδικοί στον τομέα της κυβερνοασφάλειας χρησιμοποιούν το τελευταίο ως κινητήρια δύναμη με τον ορισμό λήψης.

Μια επίθεση Drive by Download αναφέρεται στην ακούσια λήψη κακόβουλου κώδικα στον υπολογιστή ή την κινητή συσκευή που αφήνει το χρήστη ανοιχτό σε μια κυβερνοεπίθεση. Δεν χρειάζεται να κάνει κλικ σε τίποτα, να πατήσει λήψη ή να ανοίξει ένα κακόβουλο συνημμένο email για να μολυνθεί.

Μια λήψη μέσω Drive μπορεί να εκμεταλλευτεί μια εφαρμογή, λειτουργικό σύστημα ή πρόγραμμα περιήγησης ιστού που περιέχει ελαττώματα ασφαλείας, λόγω ανεπιτυχών ενημερώσεων ή έλλειψης ενημερώσεων. Σε αντίθεση με πολλούς άλλους τύπους κυβερνοεπίθεσης, το drive-by δεν βασίζεται στον χρήστη για να κάνει οτιδήποτε για να ενεργοποιήσει ενεργά την επίθεση.

Το Drive by Downloads έχουν σχεδιαστεί για να παραβιάζουν τη συσκευή για ένα ή περισσότερα από τα ακόλουθα:

- Παραβίαση τη συσκευής: Για τη δημιουργία botnet, και τη μόλυνση άλλων συσκευών.
- Κατασκοπία της δραστηριότητας του χρήστη: για υποκλοπή δεδομένων, όπως τα διαδικτυακά διαπιστευτήρια, τις οικονομικές πληροφορίες ή την ταυτότητα του χρήστη.
- Καταστροφή δεδομένων: Για τη δημιουργία προβλημάτων καθώς και προσωπικής βλάβης.

Κάθε χρήστης, χωρίς κατάλληλο λογισμικό ασφαλείας ή επιδιορθώσεις για τα τρωτά σημεία, μπορεί να γίνει θύμα μιας επίθεσης, μέσω λήψης. Υπάρχουν δύο κύριοι τρόποι με τους οποίους η κακόβουλη κίνηση μέσω λήψεων εισέρχεται στις συσκευές:

1. Εξουσιοδοτημένο χωρίς ο χρήστης να γνωρίζει τις πλήρεις συνέπειες: Μια ενέργεια μπορεί να οδηγήσει σε μόλυνση, όπως κάνοντας κλικ σε έναν σύνδεσμο σε μια παραπλανητική ψεύτικη ειδοποίηση ασφαλείας ή λήψη ενός Trojan.
2. Εντελώς μη εξουσιοδοτημένος χωρίς καμία ειδοποίηση: Επίσκεψη σε ιστότοπο και μόλυνση χωρίς καμία προτροπή ή περαιτέρω ενέργεια. Αυτές οι λήψεις μπορούν να είναι οπουδήποτε, ακόμη και σε νόμιμους ιστότοπους.

Εξουσιοδοτημένες λήψεις με κρυφά ωφέλιμα φορτία

Η εξουσιοδοτημένη μονάδα δίσκου από λήψεις είναι βελτιστοποιημένη και μπορεί ακόμη και να εντοπιστεί πριν από την επίθεση:

- Ο χάκερ δημιουργεί ένα διάνυσμα για παράδοση κακόβουλου λογισμικού — διαδικτυακά μηνύματα, διαφημίσεις, νόμιμες λήψεις προγραμμάτων.

- Η αλληλεπίδραση του χρήστη με το διάνυσμα — κάνοντας κλικ σε έναν παραπλανητικό σύνδεσμο, λήψη λογισμικού κ.λπ..
- Εγκαταστάσεις κακόβουλο λογισμικού στη συσκευή — αποτυγχάνοντας να εξαιρεθεί από επιπλέον λογισμικό ή φτάνοντας σε ιστότοπο που έχει μολυνθεί από κακόβουλο λογισμικό.
- Ο χάκερ εισέρχεται με επιτυχία στη συσκευή — το κακόβουλο λογισμικό αναλαμβάνει τον ανεπιθύμητο έλεγχο των δεδομένων.

Το λογισμικό ή οι ιστότοποι μπορεί να φαίνονται αβλαβή, αλλά έχουν καταστραφεί από κακόβουλο λογισμικό. Στην πραγματικότητα, ένας από τους μεγαλύτερους κινδύνους είναι η ευκολία προσέλκυσης επισκεπτών σε ιστότοπους ή εφαρμογές που φαίνονται αθώες.

Το Bundleware είναι μια δημοφιλής "εξουσιοδοτημένη" μέθοδος, που χρησιμοποιεί δευτερεύοντα προγράμματα που συνδέονται με το πραγματικό επιθυμητό πρόγραμμα λήψης. Μερικές φορές ονομάζονται barnacles, αυτά τα δυνητικά ανεπιθύμητα προγράμματα / εφαρμογές (PUPs / PUA) μπορεί να αποκρύψουν κακόβουλο λογισμικό ή να είναι οι ίδιες κακόβουλο λογισμικό. Αυτό το λογισμικό εμφανίζεται συνήθως ως επιλογή εξαίρεσης κατά τη λήψη δωρεάν λογισμικού ή κοινόχρηστου λογισμικού.

Το ηλεκτρονικό "ψάρεμα" μπορεί επίσης να χρησιμοποιηθεί για να δολώσει σε μια λήψη μέσω οδήγησης. Τα αναδυόμενα παράθυρα ή τα δόλια ηλεκτρονικά μηνύματα παρουσιάζονται ως γνώριμοι και έμπιστοι οργανισμοί. Μπορεί να φαίνεται μια ψεύτικη ειδοποίηση παραβίασης ασφαλείας από το πρόγραμμα περιήγησης στον ιστό ή ένα ψεύτικο email παραβίασης δεδομένων από την τράπεζα. Αυτά αποσκοπούν στην τρομοκρατία του χρήστη να κάνει κλικ σε έναν σύνδεσμο ή να κάνει λήψη ενός συνημμένου — και να μολυνθεί.

Μη εξουσιοδοτημένες λήψεις χωρίς τη γνώση του χρήστη

Μια μη εξουσιοδοτημένη μονάδα δίσκου κατά τη λήψη λειτουργεί με απλό τρόπο, παρόλο που έχει πολλές φάσεις:

- Ο χάκερ υπονομεύει μια ιστοσελίδα — συνδέοντας ένα κακόβουλο στοιχείο σε ένα ελάττωμα ασφαλείας.

- Ενεργοποιείται το στοιχείο — ο χρήστης επισκέπτεται τη σελίδα και βρίσκει τα κενά ασφαλείας της συσκευής του.
- Το στοιχείο κατεβάζει κακόβουλο λογισμικό στη συσκευή — μέσω της ασφάλειας που εκμεταλλεύεται ο χρήστης.
- Το κακόβουλο λογισμικό κάνει τη δουλειά του — αφήνει τον χάκερ να διακόψει, να ελέγξει ή να κλέψει από τη συσκευή.

Όπως σημειώθηκε παραπάνω, μη εξουσιοδοτημένος κακόβουλος κώδικας διανέμεται απευθείας από παραβιασμένους ιστότοπους. Ωστόσο, ο κώδικας τοποθετείται εκεί από χάκερ. Οι μέθοδοί τους βασίζονται σε φυσικά ελαττώματα της ψηφιακής τεχνολογίας, καθώς και σε μη ασφαλείς πρακτικές ασφαλείας.

Τα κιτ εκμετάλλευσης είναι το λογισμικό που χρησιμοποιείται για την παραβίαση των ευάλωτων διακομιστών ιστού και των συσκευών. Αυτά τα κιτ εντοπίζουν τρωτά σημεία λογισμικού σε μηχανήματα και προγράμματα περιήγησης ιστού για να προσδιορίσουν ποια συστήματα είναι εύκολο να αποκτηθούν. Τα κιτ εκμετάλλευσης συχνά χρησιμοποιούν μικρά κομμάτια κώδικα, που έχουν σχεδιαστεί για να ξεφεύγουν από απλές άμυνες και να περνούν σε μεγάλο βαθμό απαρατήρητα. Ο κώδικας διατηρείται απλός για να εστιάσει σε μια εργασία: να επικοινωνεί με έναν άλλο υπολογιστή για να εισαγάγει τον υπόλοιπο κώδικα που χρειάζεται για πρόσβαση σε μια φορητή συσκευή ή υπολογιστή.

Οι ευπάθειες ασφαλείας που ανακαλύπτονται από τα κιτ εκμετάλλευσης είναι αναπόφευκτες στην ψηφιακή εποχή. Κανένα λογισμικό ή υλικό δεν μπορεί ποτέ να γίνει τέλειο. Ακριβώς όπως ένα κτίριο μπορεί να καταστραφεί με προσεκτική μελέτη και σχεδιασμό, έτσι και κάθε λογισμικό, δίκτυο ή άλλη ψηφιακή υποδομή. Τα τρωτά σημεία παρουσιάζονται σε μερικές κοινές μορφές. Αυτές είναι:

- Zero-day exploits: Σφάλματα ασφαλείας χωρίς γνωστές επιδιορθώσεις ή ενημερώσεις κώδικα.
- Γνωστές εκμεταλλεύσεις: Ζητήματα ασφαλείας που έχουν γνωστές αλλά απεγκατεστημένες επιδιορθώσεις.

Οι εκμεταλλεύσεις Zero-day είναι δύσκολο να αποφευχθούν, αλλά τα γνωστά ζητήματα χειραγωγούνται καθαρά λόγω κακών πρακτικών ασφαλείας στον

κυβερνοχώρο. Εάν οι χρήστες τελικού σημείου και οι διαχειριστές ιστού αποτύχουν να ενημερώσουν όλο το λογισμικό τους έγκαιρα, θα καθυστερήσουν επίσης τις ζωτικής σημασίας ενημερώσεις ασφαλείας — και σε αυτό βασίζονται οι χάκερ. Το Drive by Downloads είναι μακράν μία από τις πιο δύσκολες απειλές για αποτροπή. Χωρίς την κατάλληλη προσοχή στη λεπτομέρεια και ισχυρή ασφάλεια σε όλα τα σημεία επαφής, η δραστηριότητα στον ιστό γίνεται πολύ πιο επικίνδυνη.

2.4.1. Αποτροπή από τους ιδιοκτήτες των ιστοτόπων

Ο ιδιοκτήτης ιστότοπου αποτελεί την πρώτη γραμμή άμυνας μεταξύ των χάκερ που στοχεύουν τους χρήστες του. Για την αποτροπή τέτοιων επιθέσεων είναι απαραίτητα τα εξής:

1. Να διατηρεί ενημερωμένα όλα τα στοιχεία του ιστότοπου. Αυτό περιλαμβάνει οποιαδήποτε θέματα, πρόσθετα ή οποιαδήποτε άλλη υποδομή. Κάθε ενημέρωση έχει πιθανώς νέες επιδιορθώσεις ασφαλείας για να κρατήσει έξω τους χάκερ.
2. Να καταργήσει τυχόν παλιά ή μη υποστηριζόμενα στοιχεία του ιστότοπου. Χωρίς τακτικές ενημερώσεις κώδικα ασφαλείας, το παλιό λογισμικό είναι ιδανικό για απάτες για μελέτη και εκμετάλλευση.
3. Να χρησιμοποιεί ισχυρούς κωδικούς πρόσβασης και ονόματα χρήστη για τους λογαριασμούς διαχειριστή. Οι επιθέσεις ωμής βίας δίνουν στους χάκερ μια σχεδόν άμεση διάρρηξη για προεπιλεγμένους κωδικούς πρόσβασης ή αδύναμους κωδικούς όπως "password1234". Να χρησιμοποιεί ένα πρόγραμμα δημιουργίας κωδικών πρόσβασης μαζί με έναν διαχειριστή κωδικών πρόσβασης για να παραμείνει ασφαλής.
4. Να εγκαταστήσει προστατευτικό λογισμικό ασφάλειας ιστού στον ιστότοπο. Το λογισμικό παρακολούθησης θα βοηθήσει να παρακολουθεί τυχόν κακόβουλες αλλαγές στον κώδικα υποστήριξης του ιστότοπου.
5. Η χρήση της διαφήμισης μπορεί να επηρεάσει τους χρήστες. Οι διαφημίσεις είναι ένα δημοφιλές διάνυσμα για κίνηση κατά λήψεις. Να είναι βέβαιος ότι οι χρήστες δεν λαμβάνουν προτεινόμενες ύποπτες διαφημίσεις.

2.4.2. Αποτροπή από τους χρήστες των ιστοτόπων

Ο χρήστης θα πρέπει να βασίζεται περισσότερο στις διάφορες δυνατότητες ασφαλείας που προσφέρονται στο λογισμικό του. Μπορεί να ακολουθήσει αυτές τις συμβουλές για να προετοιμάσει τον εαυτό του και το λογισμικό του έναντι μιας επίθεσης Drive by Download:

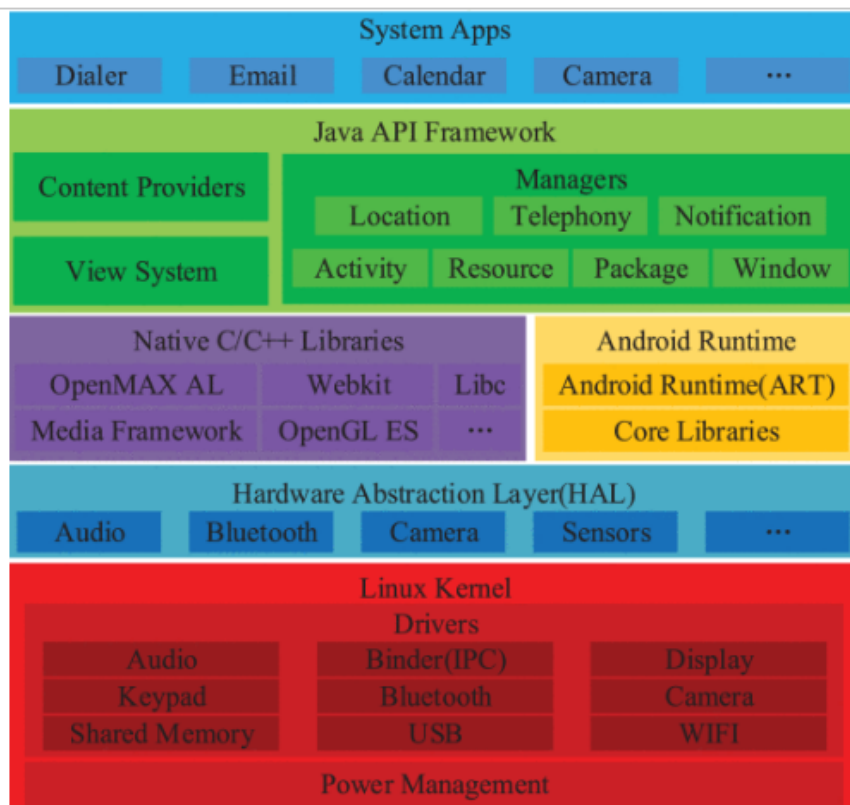
1. Να χρησιμοποιεί μόνο τον λογαριασμό διαχειριστή του υπολογιστή του για εγκαταστάσεις προγραμμάτων. Τα δικαιώματα διαχειριστή είναι απαραίτητα για την εγκατάσταση μέσω λήψεων χωρίς τη συγκατάθεσή του. Εφόσον αυτή η ρύθμιση είναι προεπιλεγμένη στον κύριο λογαριασμό του, ενδείκνυται η χρήση ενός δευτερεύοντα λογαριασμού μη διαχειριστή για καθημερινή χρήση.
2. Να διατηρεί το πρόγραμμα περιήγησης και το λειτουργικό σύστημα ενημερωμένα. Οι νέες ενημερώσεις κώδικα σφραγίζουν κενά στην άμυνά τους όπου θα μπορούσε να εισχωρήσει ο κώδικας μονάδας-από-λήψης.
3. Να είναι προσεκτικός με το να κρατάει πάρα πολλά περιττά προγράμματα και εφαρμογές. Όσο περισσότερα πρόσθετα έχει στη συσκευή του, τόσο πιο ευάλωτος είναι σε μόλυνση. Να διατηρεί μόνο το λογισμικό που εμπιστεύεται και χρησιμοποιεί συχνά. Επίσης, να καταργήσει τυχόν παλαιότερες εφαρμογές που δεν λαμβάνουν πλέον ενημερώσεις.
4. Να χρησιμοποιεί μια λύση λογισμικού ασφάλειας Διαδικτύου σε όλες τις συσκευές του. Τα προϊόντα όπως το Kaspersky Security Cloud, διατηρούν αυτόματα ενημερωμένους τους ορισμούς κακόβουλου λογισμικού του, για να εντοπίζουν τις πιο πρόσφατες απειλές. Μπορούν επίσης να σαρώσουν ιστότοπους προληπτικά, για να αποκλείσουν γνωστούς παραβιασμένους ιστότοπους.
5. Να αποφεύγει πάντα ιστότοπους που ενδέχεται να περιέχουν κακόβουλο κώδικα. Οι ιστότοποι που προσφέρουν κοινή χρήση αρχείων ή περιεχόμενο για ενηλίκους, είναι κοινά σημεία μόλυνσης. Να επισκέπτεται μόνο κεντρικούς ιστότοπους που χρησιμοποιεί συνήθως ή τουλάχιστον καθιερωμένους ιστότοπους, για να βελτιώσει τις πιθανότητές του να παραμείνει καθαρός.

6. Να διαβάσει προσεκτικά και να εξετάσει τα αναδυόμενα παράθυρα ασφαλείας στον ιστό πριν κάνει κλικ. Οι απατεώνες χρησιμοποιούν παραπλανητικές αναδυόμενες διαφημίσεις σε προγράμματα περιήγησης για υπολογιστές και κινητά που μοιάζουν με νόμιμες ειδοποιήσεις. Για να αποφύγει τη σύνδεση με έναν ιστότοπο επίθεσης, να προσέξει για τυπογραφικά λάθη, περίεργες γραμματικές και κοκκώδεις εικόνες.
7. Να χρησιμοποιεί ένα πρόγραμμα αποκλεισμού διαφημίσεων. Οι επιθέσεις λήψης Drive-by χρησιμοποιούν συχνά διαφημίσεις στο διαδίκτυο για τη μεταφόρτωση μολύνσεων. Η χρήση ενός προγράμματος αποκλεισμού διαφημίσεων μπορεί να τον βοηθήσει να μειώσετε την έκθεσή σας σε αυτόν τον τύπο επίθεσης.

3. Εφαρμογές Android

3.1. Αρχιτεκτονική συστήματος Android

Βασισμένο στον πυρήνα Linux, το λειτουργικό σύστημα Android υιοθετεί μια στοίβα λογισμικού για να δημιουργήσει την ιεραρχική αρχιτεκτονική του συστήματος. Η Google παρέχει την κλασική πολυεπίπεδη αρχιτεκτονική του συστήματος Android (Σχήμα 1), η οποία είναι διατεταγμένη, από κάτω προς τα πάνω, ως εξής: πυρήνας Linux, στρώμα αφαίρεσης υλικού, εγγενείς βιβλιοθήκες C++ / C και περιβάλλον Android Runtime, πλαίσιο Java API και το επίπεδο εφαρμογής. Κάθε επίπεδο περιέχει πολλές υπομονάδες και υποσυστήματα. Ο χώρος του πυρήνα στο κάτω μέρος της στοίβας Android έχει τον πυρήνα Linux ως τον ακρογωνιαίο λίθο του, ενώ ο χώρος χρήστη στην κορυφή του συστήματος Android αποτελείται από εγγενείς βιβλιοθήκες C++ / C, Android Runtime και το πλαίσιο Java API. Οι χώροι πυρήνα και χρήστη συνδέονται με κλήσεις συστήματος. Τα προγράμματα χώρου χρήστη είναι κυρίως γραμμένα σε C++ ή Java. Μέσω της εγγενούς διεπαφής Java, το επίπεδο Java και το εγγενές επίπεδο του χώρου χρήστη συνδέονται με το υπόλοιπο σύστημα Android.



Σχήμα 1 Αρχιτεκτονική πλατφόρμας Android (developers, 2021)

3.2. Μηχανισμοί ασφαλείας Android

Γενικά, το Android είναι ένα λειτουργικό σύστημα διαχωρισμένο με προνόμια. Το σύστημα επιτυγχάνει λειτουργίες συστήματος υψηλού επιπέδου εκτελώντας ένα σύνολο υπηρεσιών συστήματος μέσω ενός μηχανισμού επικοινωνίας μεταξύ διεργασιών, γνωστό ως Binder. Το σύστημα Android απομονώνει τις εφαρμογές που εκτελούνται χρησιμοποιώντας τα μοναδικά αναγνωριστικά συστήματος τους (UID Linux). Οι εφαρμογές Android έχουν πολύ λίγα δικαιώματα από προεπιλογή. Πρέπει να αποκτήσουν λεπτομερείς άδειες για να αλληλεπιδρούν με υπηρεσίες συστήματος, συσκευές υλικού και άλλες εφαρμογές. Τα δικαιώματα που απαιτούνται από μια εφαρμογή Android ορίζονται στο αντίστοιχο αρχείο μανιφέστου (Androidmanifest.xml) και χορηγούνται κατά την εγκατάσταση ή κατά την εκτέλεση της εφαρμογής. Το σύστημα Android χρησιμοποιεί UID για να διακρίνει τα δικαιώματα που χορηγούνται σε κάθε εφαρμογή, εκτελεί αυτά τα δικαιώματα, ενώ η εφαρμογή εκτελείται και περιορίζει περαιτέρω τα δικαιώματα κάθε διεργασίας με το SELINUX (Elenkov, 2015).

Κατά τη διαδικασία ενημέρωσης και επανάληψης του συστήματος, οι προγραμματιστές Android έχουν αφιερώσει μεγάλη προσοχή στη βελτίωση των λειτουργιών ασφαλείας. Για παράδειγμα, το Android Q, το οποίο κυκλοφόρησε το 2019, έχει πολλά νέα χαρακτηριστικά ασφαλείας, όπως κρυπτογράφηση βάσει αρχείων, έλεγχο πρόσβασης για ευαίσθητες πληροφορίες, έλεγχο πρόσβασης για κάμερα φόντου / μικρόφωνο, λειτουργία κλειδώματος, κρυπτογραφημένο αντίγραφο ασφαλείας και μηχανισμό, που ονομάζεται Google Play Protect. Το Android Q προστατεύει το απόρρητο και την ασφάλεια των χρηστών από πολλαπλές προοπτικές. Διαθέτει επίσης βελτιωμένο μηχανισμό ελέγχου αδειών, δίνει στους χρήστες περισσότερο έλεγχο σχετικά με την αποκάλυψη της τοποθεσίας τους, απαγορεύει στις εφαρμογές παρασκηνίου να ξεκινούν δραστηριότητες, περιορίζει την πρόσβαση εφαρμογών σε μη επαναρυθμιζόμενα αναγνωριστικά συσκευής (όπως IMEI και σειριακός αριθμός) και επιτρέπει την τυχαιοποίηση διευθύνσεων MAC από προεπιλογή. Ωστόσο, το κακόβουλο λογισμικό εξακολουθεί να αποτελεί πρόβλημα.

3.3. Ταξινόμηση κακόβουλου λογισμικού Android

Το κακόβουλο λογισμικό είναι ένας τύπος εφαρμογής που περιέχει κακόβουλο εκτελέσιμο κώδικα που μπορεί να καταστρέψει τις κανονικές ή προκαθορισμένες υπηρεσίες και λειτουργίες ενός συστήματος ή άλλης εφαρμογής. Η αναφορά των G. Suarez-Tangil, J. E. Tapiador, P. Peris-Lopez and A. Ribagorda (2014), ταξινομεί το κακόβουλο λογισμικό σε έξυπνες συσκευές από τρεις προοπτικές: στόχους και συμπεριφορά επίθεσης, διαδρομές διανομής και μόλυνσης και τρόπους απόκτησης προνομίων. Οι στόχοι και η συμπεριφορά επίθεσης μπορεί να περιλαμβάνουν, αλλά δεν περιορίζονται σε, απάτη και κακή χρήση υπηρεσιών, ανεπιθύμητη αλληλογραφία, κατασκοπεία, κλοπή δεδομένων και δολιοφθορά. Οι οδοί διανομής και μόλυνσης περιλαμβάνουν μεταξύ άλλων την αγορά λογισμικού, εφαρμογές, προγράμματα περιήγησης ιστού, SMS, δίκτυο και υπολογιστές. Οι μέθοδοι απόκτησης προνομίων περιλαμβάνουν χειραγώγηση χρήστη και τεχνική εκμετάλλευση.

Με την ταχεία ανάπτυξη του διαδικτύου για κινητά και των έξυπνων συσκευών, το κακόβουλο λογισμικό που στοχεύει την πλατφόρμα για κινητές συσκευές Android, έχει εμφανιστεί σε διάφορες μορφές. Αναφερόμενοι στην παραδοσιακή έννοιά του, το κακόβουλο λογισμικό Android μπορεί να χωριστεί στα εξής: trojan, backdoor, worm, botnet, spyware, επιθετικό adware και ransomware. Οι Felt et al (2011) ταξινόμησε το κακόβουλο λογισμικό Android σύμφωνα με τα ανθρώπινα συμπεριφορικά κίνητρα, συμπεριλαμβανομένης της καινοτομίας και της διασκέδασης, της πώλησης πληροφοριών χρήστη, της κλοπής διαπιστευτηρίων χρήστη, της πραγματοποίησης κλήσεων υψηλής τιμής, της αποστολής μηνυμάτων SMS, των ανεπιθύμητων μηνυμάτων SMS, της βελτιστοποίησης μηχανών αναζήτησης και της λήψης λύτρων. Οι Zhou και Jiang (2012) ταξινόμησαν και ανέλυσαν το κακόβουλο λογισμικό Android από την άποψη της εγκατάστασης, της ενεργοποίησης, των κακόβουλων ωφέλιμων φορτίων και της κατάχρησης αδειών. Στις ερευνητικές εκθέσεις της, η Google χρησιμοποιεί πολύ συντηρητικές λέξεις για να περιγράψει κακόβουλο λογισμικό, όπως δυνητικά επιβλαβείς εφαρμογές (PHA). Τα PHA στην αγορά εφαρμογών Google ταξινομούνται ως απάτη κλικ, απάτη μέσω SMS, λογισμικό κατασκοπείας, απάτη με διόδια, trojans, εχθρικά προγράμματα λήψης, backdoors, phishing, κλιμάκωση προνομίων και λογισμικό υποκλοπής εμπορίου.

4. Προσεγγίσεις ανίχνευσης κακόβουλου λογισμικού Android με βάση τη μηχανική εκμάθηση. Εισαγωγή στην μηχανική μάθηση

Η μηχανική μάθηση είναι ένας κλάδος της έρευνας και εφαρμογής της τεχνητής νοημοσύνης και αρκετοί ερευνητές έχουν παράσχει ορισμούς της μηχανικής μάθησης. Σύμφωνα με τον Schapire (2013), η μηχανική μάθηση αποτελείται από μια σειρά τεχνικών για την αυτοματοποίηση των προβλέψεων που βασίζονται σε προηγούμενες παρατηρήσεις. Με βάση μια αναλογία μεταξύ των αλγορίθμων μηχανικής μάθησης και των εργασιών που εκτελούνται από τον ανθρώπινο εγκέφαλο, η μηχανική μάθηση μπορεί να χωριστεί χονδρικά σε πέντε παραδείγματα με διαφορετικές θεωρητικές ιδέες: συμβολιστές, συνδετικοί, εξελικτικοί, Bayesians και αναλογιστές. Κάθε κατηγορία μηχανικής μάθησης έχει τους δικούς της ερευνητικούς τομείς και τους αντίστοιχους αλγόριθμους, σύμφωνα με τις αντίστοιχες θεμελιώδεις έννοιες. Μια άλλη αντιπροσωπευτική και ευρέως χρησιμοποιούμενη ταξινόμηση προσεγγίσεων μηχανικής μάθησης βασίζεται στη μέθοδο μάθησης, η οποία τυπικά χωρίζεται σε εποπτευόμενη μάθηση, χωρίς επίβλεψη μάθηση, ημι-εποπτευόμενη μάθηση και ενισχυτική μάθηση (Das & Behera, 2017).

Η εποπτευόμενη μάθηση χρησιμοποιεί ένα επισημασμένο σύνολο δεδομένων δειγμάτων ή περιπτώσεων για την εκπαίδευση του προγνωστικού μοντέλου, το οποίο χρησιμοποιείται συχνά για την επίλυση προβλημάτων ταξινόμησης ή παλινδρόμησης. Όταν η πρόβλεψη της εξόδου είναι μια συνεχής μεταβλητή, είναι πρόβλημα παλινδρόμησης και όταν η πρόβλεψη είναι διακριτή, είναι πρόβλημα ταξινόμησης. Η μάθηση χωρίς επίβλεψη δεν απαιτεί ειδικά επισημασμένα σύνολα δεδομένων για την εκπαίδευση του μοντέλου πρόβλεψης. Ο σκοπός αυτού του είδους μηχανικής μάθησης είναι να ανακαλύψει την εσωτερική δομή ή τα χαρακτηριστικά διανομής των ίδιων των συνόλων δεδομένων και συχνά εφαρμόζεται σε προβλήματα όπως η ομαδοποίηση δεδομένων και η μείωση της διάστασης χαρακτηριστικών.

Η ημι-εποπτευόμενη μάθηση συνδυάζει στοιχεία εποπτευόμενης μάθησης και μάθησης χωρίς επίβλεψη, χρησιμοποιώντας δεδομένα τόσο με ετικέτα, όσο και χωρίς ετικέτα. Η βασική ιδέα της ημι-εποπτευόμενης μάθησης είναι να επιτρέψει σε έναν

εκπαιδευόμενο να επισημάνει τα μη επισημασμένα δείγματα δεδομένων χρησιμοποιώντας ένα μοντέλο της διανομής δεδομένων. Αυτός ο τύπος μηχανικής εκμάθησης χρησιμοποιείται κυρίως σε σενάρια όπου υπάρχει μόνο ένας μικρός αριθμός δεδομένων με ετικέτα στο σύνολο δεδομένων.

Στην ενισχυτική μάθηση, δεν υπάρχουν δεδομένα με ετικέτα όπως στην εποπτευόμενη μάθηση. Η ενισχυτική μάθηση προχωρά ως ένας κύκλος πρόβλεψης και αξιολόγησης, όπου τα δεδομένα εισόδου μεταφέρονται απευθείας στο μοντέλο, οδηγώντας στη δυναμική προσαρμογή των παραμέτρων του μοντέλου. Το μοντέλο μάθησης και τα δεδομένα εκπαίδευσης συνάγονται με τη λήψη ανατροφοδότησης από το περιβάλλον για την ενημέρωση των παραμέτρων του μοντέλου. Οι κοινές εφαρμογές αυτού του τύπου μηχανικής μάθησης περιλαμβάνουν δυναμικά συστήματα και έλεγχο ρομπότ.

Ένα τυπικό έργο που χρησιμοποιεί μεθόδους μηχανικής μάθησης για την επίλυση προβλημάτων πραγματικού κόσμου, αποτελείται από τις ακόλουθες κύριες διαδικασίες (Marsland, 2015):

1. Περίληψη του προβλήματος που πρέπει να λυθεί: Καθορίστε εάν πρόκειται για πρόβλημα ταξινόμησης, πρόβλημα παλινδρόμησης, πρόβλημα ομαδοποίησης ή κάτι άλλο.
2. Απόκτηση και ανάλυση δειγμάτων δεδομένων: Τα δεδομένα που λαμβάνονται πρέπει να είναι αντιπροσωπευτικά του τομέα του προβλήματος. Θα πρέπει να έχει επαρκή όγκο και να μην είναι υπερβολικά λοξό.
3. Προεπεξεργασία δεδομένων: Λαμβάνοντας υπόψη τους περιορισμούς και τα πιθανά σφάλματα στα δείγματα δεδομένων, η βελτιστοποίηση πραγματοποιείται μέσω της προεπεξεργασίας. Αυτό περιλαμβάνει καθαρισμό δεδομένων, κανονικοποίηση, διακριτοποίηση, παραγοντοποίηση, επεξεργασία τιμών που λείπουν, τμηματοποίηση δεδομένων και άλλες μεθόδους.
4. Επιλογή χαρακτηριστικών: Αυτό περιλαμβάνει την επιλογή των πιο σημαντικών χαρακτηριστικών και την απόρριψη ασήμαντων χαρακτηριστικών χρησιμοποιώντας σχετικές τεχνικές ανάλυσης εγκυρότητας χαρακτηριστικών, όπως η ανάλυση συσχέτισης, το τεστ χ -τετράγωνο, η μέση αμοιβαία πληροφορία, η υπό συνθήκη εντροπία και η μεταγενέστερη πιθανότητα.

5. Επιλογή και εκπαίδευση μοντέλου: Επιλέξτε το μοντέλο σύμφωνα με τα δεδομένα και το πρόβλημα που θα λυθεί και χρησιμοποιήστε τα δεδομένα εκπαίδευσης για να λάβετε τις παραμέτρους του μοντέλου.
6. Αξιολόγηση και βελτιστοποίηση μοντέλου: Χρησιμοποιήστε δεδομένα δοκιμής για να αξιολογήσετε το μοντέλο όσον αφορά την ακρίβεια, την ταχύτητα εκπαίδευσης, την πολυπλοκότητα του χώρου, την αξιοπιστία, τη φορητότητα και τη γενίκευση και βελτιστοποιήστε το μοντέλο με τις κατάλληλες μεθόδους.
7. Χρησιμοποιήστε το νέο σύνολο δεδομένων για να κάνετε προβλέψεις και να λύσετε πρακτικά προβλήματα.
8. Αξιολόγηση της μεθόδου μηχανικής μάθησης για την εκμάθηση της απόδοσής της (π.χ. ακρίβεια, ειδικότητα) στο νέο σύνολο δεδομένων.

Ο παραδοσιακός εντοπισμός κακόβουλου λογισμικού Android βασίζεται σε μια βιβλιοθήκη λειτουργιών κακόβουλου κώδικα, η οποία πρέπει να ενημερώνεται με την πάροδο του χρόνου, για να διασφαλιστεί η ακρίβεια των αποτελεσμάτων ανίχνευσης. Τα τελευταία χρόνια, οι ερευνητές έχουν εφαρμόσει ευρέως τεχνικές μηχανικής εκμάθησης για τον εντοπισμό κακόβουλου λογισμικού μέσω μοντέλων εκπαίδευσης σε μεγάλο αριθμό λειτουργιών, για να επιτύχουν την ικανότητα ανίχνευσης νέου κακόβουλου λογισμικού. Η τρέχουσα θεωρητική και πρακτική εργασία επικεντρώνεται κυρίως στην απόκτηση δειγμάτων, την προεπεξεργασία δεδομένων, την επιλογή χαρακτηριστικών, τα μοντέλα και τη θεωρία μηχανικής μάθησης, την αξιολόγηση της αποτελεσματικότητας ανίχνευσης και άλλες πτυχές.

5. Κοινά μοντέλα και αλγόριθμοι μηχανικής μάθησης

5.1. Εισαγωγή στα κοινά μοντέλα και αλγόριθμους

Σε αυτήν την ενότητα, ταξινομούνται πρώτα τα μοντέλα και οι αλγόριθμοι μηχανικής εκμάθησης που χρησιμοποιούνται συνήθως σύμφωνα με τη μέθοδο εκμάθησής τους. Μέθοδοι όπως τα νευρωνικά δίκτυα, η μάθηση συνόλου και η διαδικτυακή μάθηση μπορούν να εφαρμοστούν με διαφορετικούς τρόπους, όπως με την εφαρμογή της εποπτευόμενης μάθησης και της μάθησης χωρίς επίβλεψη, ανάλογα με τη συγκεκριμένη κατάσταση. Λαμβάνοντας ως παράδειγμα τα νευρωνικά δίκτυα, ο αλγόριθμος που βασίζεται στο perceptron υλοποιεί την εποπτευόμενη μάθηση, ενώ ο αλγόριθμος που βασίζεται στη μηχανή Boltzmann χρησιμοποιεί μάθηση χωρίς επίβλεψη. Για ένα άλλο παράδειγμα, η μάθηση συνόλου συνδυάζει πολλούς μαθητές σε ένα μοντέλο πρόβλεψης με στόχο τη βελτίωση της ακρίβειας της πρόβλεψης. Αυστηρά μιλώντας, η μάθηση συνόλου δεν είναι μια μέθοδος μάθησης, αλλά ένας τρόπος συνδυασμού των μαθητών.

Οι τεχνικές μηχανικής μάθησης μπορούν να εξεταστούν από διαφορετικές οπτικές γωνίες. Για παράδειγμα, μπορούν να χωριστούν σε ταξινόμηση, παλινδρόμηση, ομαδοποίηση και μείωση διαστάσεων, σύμφωνα με τους στόχους της εργασίας. Εναλλακτικά, η μάθηση μπορεί να γίνει μέσω της διαδικτυακής μάθησης και της ομαδικής μάθησης, ανάλογα με το εάν πραγματοποιείται σταδιακή μάθηση. Οι αλγόριθμοι μηχανικής μάθησης κατηγοριοποιούνται σύμφωνα με τη μέθοδο εκμάθησής τους, η οποία είναι η πιο συχνά χρησιμοποιούμενη ταξινόμηση. Ωστόσο, εξακολουθούν να υπάρχουν αντιπαραθέσεις. Για παράδειγμα, η βαθιά μάθηση μπορεί να θεωρηθεί ως μια πιο σύνθετη επέκταση των νευρωνικών δικτύων. Παρόλο που είναι δύσκολο να έχει κανείς μία καθολικά αποδεκτή κατανόηση της ταξινόμησης των μοντέλων και των αλγορίθμων μηχανικής μάθησης, παραθέτει τα πιο δημοφιλή μοντέλα και αλγόριθμους.

Με βάση την ταξινόμηση κοινών μοντέλων και αλγορίθμων, παρουσιάζονται εν συντομία αυτά που χρησιμοποιούνται συνήθως για τον εντοπισμό κακόβουλου λογισμικού Android:

1. Δένδρα αποφάσεων (DT): Ένα δέντρο αποφάσεων χρησιμοποιείται για τη λήψη αποφάσεων με βάση στοιχεία δεδομένων που διατηρούνται σε μια δομή που μοιάζει με δέντρο. Όπως σε ένα είδος τρόπου σκέψης και επεξεργασίας ανάλογο με εκείνον που υιοθετείται από τους ανθρώπους όταν αντιμετωπίζουν προβλήματα απόφασης, η βασική του διαδικασία ακολουθεί την απλή και διαισθητική στρατηγική «διαίρει και βασίλευε».
2. Naive Bayesian (NB): Οι μέθοδοι Naive Bayes είναι μια ομάδα εποπτευόμενων αλγορίθμων μάθησης που βασίζονται στο θεώρημα του Bayes. Αυτές οι μέθοδοι «αφελώς» υποθέτουν ότι κάθε ζεύγος χαρακτηριστικών είναι ανεξάρτητο το ένα από το άλλο, δηλαδή, κάθε χαρακτηριστικό επηρεάζει ανεξάρτητα τα αποτελέσματα πρόβλεψης.
3. Γραμμικό μοντέλο (LM): Το γραμμικό μοντέλο βασίζεται σε μια συνάρτηση που προβλέπει αποτελέσματα μέσω ενός γραμμικού συνδυασμού χαρακτηριστικών και η βέλτιστη λύση των παραμέτρων στη συνάρτηση επιλύεται χρησιμοποιώντας το σύνολο δεδομένων εκπαίδευσης. Όταν το γραμμικό μοντέλο εφαρμόζεται σε προβλήματα παλινδρόμησης, είναι γνωστό ως γραμμική παλινδρόμηση. Όταν χρησιμοποιείται ένα γραμμικό μοντέλο για την επίλυση προβλημάτων ταξινόμησης, μπορεί να υιοθετηθεί η λογιστική παλινδρόμηση. Λαμβάνοντας ως παράδειγμα τη δυαδική ταξινόμηση, η κύρια αρχή της μεθόδου λογιστικής παλινδρόμησης είναι η χρήση της λογιστικής συνάρτησης για τη μετατροπή της πραγματικής τιμής που δημιουργείται από το μοντέλο γραμμικής παλινδρόμησης σε τιμή 0 ή 1, που αντιστοιχεί στις δύο κατηγορίες που πρόκειται να προβλεφθούν.
4. Υποστήριξη διανυσματική μηχανή (SVM): Από τη σκοπιά της γεωμετρίας, η αρχή του SVM είναι να βρει ένα βέλτιστο διαχωριστικό υπερεπίπεδο που να πληροί τις απαιτήσεις ταξινόμησης, έτσι ώστε το υπερεπίπεδο να μπορεί να διαχωρίσει σημεία σε έναν n -διάστατο χώρο «όσο το δυνατόν περισσότερο» ενώ διασφαλίζοντας την ακρίβεια της ταξινόμησης. Για να ικανοποιηθεί η απαίτηση ακραίας τιμής του «όσο είναι δυνατόν», μπορεί να χρησιμοποιηθεί μια μέθοδος όπως ο πολλαπλασιαστής Lagrange.
5. K- Nearest Neighbor (KNN): Η στρατηγική του αλγόριθμου k -πλησιέστερου γείτονα είναι να βρει τα δείγματα με σήμανση k πιο κοντά στο δείγμα που πρόκειται να ταξινομηθεί στο χώρο του δείγματος. Εάν τα περισσότερα από τα k -πλησιέστερα δείγματα ανήκουν σε μια συγκεκριμένη κατηγορία, τότε το

δείγμα που θα ταξινομηθεί ανήκει επίσης σε αυτήν την κατηγορία. Η απόσταση μπορεί να μετρηθεί με μια ποικιλία μετρήσεων και η Ευκλείδεια απόσταση είναι η πιο συχνά χρησιμοποιούμενη. Αυτός ο αλγόριθμος κάνει μια ταξινόμηση σύμφωνα με την κατανομή των δεδομένων σε ένα σύνολο επισημασμένων δειγμάτων.

6. Αλγόριθμος ομαδοποίησης K-means: Ο στόχος του αλγόριθμου ομαδοποίησης k-means είναι να ταξινομήσει παρόμοια αντικείμενα στο ίδιο σύμπλεγμα. Αυτός ο αλγόριθμος χωρίζει τις περιπτώσεις σε k διαφορετικά συμπλέγματα όπου ο αριθμός των συστάδων καθορίζεται από τον χρήστη. Το κέντρο μάζας κάθε συστάδας αντιπροσωπεύεται από το μέσο όρο όλων των αντικειμένων στο σύμπλεγμα. Κατά τον υπολογισμό της απόστασης μεταξύ κάθε σημείου και του κέντρου μάζας, υπάρχουν πολλές μετρικές διαθέσιμες, όπως η Ευκλείδεια απόσταση.
7. Νευρωνικά δίκτυα (NN) και Deep Learning (DL): Τα νευρωνικά δίκτυα προσομοιώνουν τον τρόπο με τον οποίο οι βιολογικοί νευρώνες αλληλεπιδρούν με τον πραγματικό κόσμο και χρονολογούνται από τη δεκαετία του 1940. Το 1943, ο νευροφυσιολόγος Warren McCulloch και ο λογικός Walter Pitts, εμπνευσμένοι από τη δομή των βιολογικών νευρώνων, πρότειναν για πρώτη φορά ένα αφηρημένο μοντέλο νευρώνων «M-P», του οποίου η τιμή εξόδου είναι «0» ή «1», που αντιστοιχεί στις δύο καταστάσεις. νευρωνικής αναστολής και διέγερσης σε ένα βιολογικό νευρωνικό δίκτυο.

Πολλοί από αυτούς τους νευρώνες μπορούν να συνδεθούν σε μια πολυεπίπεδη δομή για να σχηματίσουν ένα νευρωνικό δίκτυο. Από την άποψη του μακροσκοπικού μαθηματικού υπολογισμού, ένα νευρωνικό δίκτυο είναι ένα μαθηματικό μοντέλο με πολλές παραμέτρους. Τα μοντέλα μηχανικής μάθησης που βασίζονται σε νευρωνικά δίκτυα περιλαμβάνουν κυρίως δύο κατηγορίες. Μια κατηγορία βασίζεται στο μοντέλο perceptron [229]. Το μοντέλο πολυστρωματικού perceptron (MP) χρησιμοποιείται συνήθως και συνήθως χρησιμοποιεί τον αλγόριθμο της αντίστροφης διάδοσης σφαλμάτων (BP) [230] για να εκπαιδεύσει το δίκτυο σύμφωνα με την αναμενόμενη έξοδο. Η άλλη κατηγορία βασίζεται στη μηχανή Boltzmann [231], η οποία είναι ένα τυχαίο νευρωνικό δίκτυο του οποίου η έξοδος καθορίζεται σύμφωνα με μια κατανομή πιθανοτήτων. Τα μοντέλα

περιορισμένης μηχανής Boltzmann (RBM) [232] χρησιμοποιούνται συνήθως και συνήθως χρησιμοποιούν τον αλγόριθμο αντίθεσης απόκλισης (CD) [233] για να εκπαιδεύσουν το δίκτυο σύμφωνα με τα δεδομένα εκπαίδευσης.

Τα μοντέλα βαθιάς μάθησης αποτελούνται από δίκτυα πολλαπλών επιπέδων που χρησιμοποιούνται για την εκμάθηση αναπαραστάσεων δεδομένων με πολλαπλά επίπεδα αφαίρεσης [234]. Η βαθιά μάθηση προέκυψε από την ανάπτυξη των νευρωνικών δικτύων και διαφορετικές μελέτες έχουν διαφορετικές απόψεις για τη σχέση μεταξύ των δύο. Αναφ. [235] υποστηρίζει ότι η βαθιά μάθηση είναι ένας τύπος μηχανικής μάθησης που βασίζεται σε νευρωνικά δίκτυα, ενώ η Αναφ. [236] υποστηρίζει ότι τα νευρωνικά δίκτυα είναι η πιο συχνά χρησιμοποιούμενη μορφή βαθιάς μάθησης. Λόγω αυτής της διαμάχης, η παρούσα εργασία δεν θα επιβάλει διάκριση μεταξύ νευρωνικών δικτύων και βαθιάς μάθησης. Αντίθετα, τα δύο θα συνδυαστούν για τους σκοπούς της συζήτησης.

8. Εκμάθηση συνόλου: Η μάθηση συνόλου επιτυγχάνει καλύτερες επιδόσεις στη γενίκευση από οποιονδήποτε μεμονωμένο μαθητή, συνδυάζοντας πολλαπλά μοντέλα μάθησης. Ανάλογα με τον τρόπο που είναι διατεταγμένοι οι εκπαιδευόμενοι, οι μέθοδοι μάθησης συνόλου μπορούν να χωριστούν σε δύο κατηγορίες. Η μία είναι η μέθοδος σειριοποίησης, όπου οι εκπαιδευόμενοι λειτουργούν διαδοχικά και υπάρχει ισχυρή εξάρτηση μεταξύ των μεμονωμένων μαθητών, όπως στην ενίσχυση. Το άλλο είναι η παραλληλοποίηση, όπου οι εκπαιδευόμενοι λειτουργούν ταυτόχρονα χωρίς ισχυρή εξάρτηση μεταξύ των μεμονωμένων μαθητών, όπως στο bagging και στα τυχαία δάση.
9. Διαδικτυακή μάθηση: Η διαδικτυακή μάθηση δεν χρησιμοποιεί ολόκληρο το σετ εκπαίδευσης κάθε φορά. Αντίθετα, οι εισροές στο μοντέλο υποβάλλονται σε επεξεργασία σε παρτίδες, έτσι ώστε το μοντέλο πρόβλεψης να ενημερώνεται συνεχώς με νέα δεδομένα εκπαίδευσης. Η διαδικτυακή μάθηση είναι κατάλληλη για σενάρια όπου οι αλγόριθμοι πρέπει να προσαρμοστούν δυναμικά για να χωρέσουν νέα μοτίβα στα δεδομένα. Οι κοινώς χρησιμοποιούμενοι διαδικτυακοί αλγόριθμοι μάθησης περιλαμβάνουν την παθητική επιθετική μάθηση και την προσαρμοστική μάθηση τακτοποίησης.

5.2. Αξιολόγηση της αποτελεσματικότητας ανίχνευσης

Η αξιολόγηση της απόδοσης μοντέλων ή αλγορίθμων είναι ένα σημαντικό θέμα στον τομέα της μηχανικής μάθησης. Γενικά, η ανίχνευση κακόβουλου λογισμικού Android μελετάται ως τυπικό πρόβλημα δυαδικής ταξινόμησης. Οι μετρήσεις μπορούν να χρησιμοποιηθούν όχι μόνο για την αξιολόγηση της προγνωστικής ακρίβειας του ταξινομητή, αλλά και για τη βελτιστοποίηση του μοντέλου.

5.2.1. Διάρθρωση συνόλων δεδομένων

Το αρχικό σύνολο δεδομένων χωρίζεται σε ένα σετ εκπαίδευσης και σε ένα σύνολο δοκιμής. Το σετ εκπαίδευσης χρησιμοποιείται για την επιλογή του μοντέλου και τον συντονισμό των παραμέτρων, ενώ το σύνολο δοκιμής χρησιμοποιείται για την αξιολόγηση της απόδοσης του ταξινομητή. Προκειμένου να διασφαλιστεί η ακεραιότητα της αξιολόγησης, τα σετ εκπαίδευσης και δοκιμών θα πρέπει να είναι αμοιβαία αποκλειόμενα και η διανομή των δεδομένων θα πρέπει να συνάδει με το αρχικό σύνολο δεδομένων όσο το δυνατόν περισσότερο. Στην αξιολόγηση της αποτελεσματικότητας ανίχνευσης, μέθοδοι όπως η αναμονή, η διασταυρούμενη επικύρωση και η εκκίνηση χρησιμοποιούνται συχνά για την τμηματοποίηση του συνόλου δεδομένων σε σύνολα εκπαίδευσης και δοκιμών. Λαμβάνοντας ως παράδειγμα τη συνήθη διασταυρούμενη επικύρωση k-fold, τα κύρια βήματα είναι τα εξής:

1. Το αρχικό σύνολο δεδομένων χωρίζεται σε k αμοιβαία αποκλειόμενα υποσύνολα περίπου ίσου μεγέθους.
2. Ένα από τα υποσύνολα επιλέγεται ως το σύνολο δοκιμής και τα υπόλοιπα $(k-1)$ υποσύνολα αποτελούν το σύνολο εκπαίδευσης. Το σετ εκπαίδευσης χρησιμοποιείται για την επιλογή μοντέλου και τον συντονισμό παραμέτρων και, στη συνέχεια, ο λαμβανόμενος ταξινομητής αξιολογείται χρησιμοποιώντας το σύνολο δοκιμής.
3. Το βήμα 2 επαναλαμβάνεται k φορές για να ληφθούν τα αποτελέσματα αξιολόγησης απόδοσης για k ομάδες. Η μέση τιμή των k ομάδων αποτελεσμάτων λαμβάνεται ως η συνολική απόδοση του ταξινομητή.

5.2.2. Αξιολόγηση της απόδοσης του ταξινομητή

Υπάρχουν πολλές μετρήσεις απόδοσης για ταξινομητές. Ως τυπικό πρόβλημα δυαδικής ταξινόμησης, τα αποτελέσματα μιας πρόβλεψης για το εάν μια εφαρμογή Android περιέχει κακόβουλο λογισμικό, μπορούν να χωριστούν σε τέσσερις τύπους:

1. Αληθινό θετικό (TP): η εφαρμογή είναι κακόβουλη εφαρμογή και είχε προβλεφθεί σωστά ότι είναι κακόβουλη.
2. Ψευδές θετικό (FP): η εφαρμογή δεν είναι κακόβουλη εφαρμογή, αλλά εσφαλμένα είχε προβλεφθεί ότι είναι κακόβουλη.
3. Αληθινό αρνητικό (TN): η εφαρμογή δεν είναι κακόβουλη εφαρμογή και είχε προβλεφθεί σωστά ότι δεν είναι κακόβουλη.
4. Ψευδές αρνητικό (FN): η εφαρμογή είναι κακόβουλη εφαρμογή αλλά εσφαλμένα είχε προβλεφθεί ως μη κακόβουλη.

Τα παραπάνω τέσσερα αποτελέσματα είναι αμοιβαία αποκλειόμενα και επομένως το άθροισμά τους είναι ο συνολικός αριθμός δειγμάτων στη δοκιμή. Με βάση αυτές τις τέσσερις βασικές έννοιες, έχει προκύψει μια σειρά από μετρήσεις απόδοσης. Ορισμένες μετρήσεις που χρησιμοποιούνται συνήθως, είναι οι εξής:

1. Η ακρίβεια (Acc) αντιπροσωπεύει την αναλογία των σωστών προβλέψεων μεταξύ του συνολικού αριθμού των δειγμάτων στη δοκιμή. Η εξίσωση 1 δείχνει πώς υπολογίζεται η ακρίβεια.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

2. Ποσοστό σφάλματος (Err) αντιπροσωπεύει την αναλογία ψευδών προβλέψεων μεταξύ του συνολικού αριθμού δειγμάτων στη δοκιμή. Η εξίσωση 2 δείχνει πώς υπολογίζεται το ποσοστό σφάλματος.

$$Err = \frac{FP + FN}{TP + TN + FP + FN} \quad (2)$$

3. Η ακρίβεια (P) αντιπροσωπεύει την αναλογία όλων των δειγμάτων που προβλέφθηκαν σωστά ως θετικά μεταξύ όλων των δειγμάτων που προβλέπονται θετικά. Η εξίσωση 3 δείχνει πώς υπολογίζεται η ακρίβεια.

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

4. Η ανάκληση (R) αντιπροσωπεύει την αναλογία όλων των θετικών δειγμάτων που έχουν προβλεφθεί σωστά μεταξύ όλων των θετικών δειγμάτων. Η εξίσωση 4 δείχνει πώς υπολογίζεται η ανάκληση.

$$R = \frac{TP}{TP + FN}$$

6. Παραδείγματα προβλημάτων κακόβουλου λογισμικού (malware), λύση τους και παρουσίαση μεθοδολογιών

Πίνακας 2

Άρθρο 1: Catch them Alive: Ανίχνευση κακόβουλου λογισμικού με προσέγγιση μέσω της εγκληματολογίας μνήμης, πολλαπλής μάθησης και όρασης του υπολογιστή

No	Πηγή	Πρόβλημα	Λύση	Μεθοδολογία	Αποτελέσματα
1	(Bozkir, Tahillioglu, Aydos, & Kara, 2021)	Εντοπισμός και ανίχνευση κακόβουλου λογισμικού σε καλοήθη εκτελέσιμα αρχεία	Ανίχνευση κακόβουλου λογισμικού	Μελέτη δύο φάσεων, χρησιμοποιώντας δύο περιγραφείς εικόνων, GIST και HOG, για ανάλυση εικόνων ένδειξης μνήμης και χρήση του UMAP	UMAP: Η κατάλληλη μέθοδος Συνεισφέρει και στα δύο προβλήματα ανισορροπίας τάξης και οπτικοποίησης χαρακτηριστικών

Στη μελέτη των Bozkir, Tahillioglu, Aydos & Kara (2021), πραγματοποιήθηκε μια μελέτη δύο φάσεων για τον εντοπισμό και την ανίχνευση κακόβουλου λογισμικού σε καλοήθη εκτελέσιμα αρχεία, χρησιμοποιώντας δύο περιγραφείς εικόνων, συγκεκριμένα GIST και HOG, για ανάλυση εικόνων ένδειξης μνήμης. Επιπλέον, στη δεύτερη φάση, χρησιμοποιήθηκε τελευταίας τεχνολογίας πολλαπλή τεχνική μάθησης, που ονομάζεται UMAP, η οποία μπορεί να “καταλάβει” καλύτερα εάν μια ύποπτη διαδικασία στη μνήμη ενός υπολογιστή είναι κακόβουλη ή όχι. Τα πειράματα που πραγματοποιήθηκαν με το UMAP, δείχνουν ότι είναι η κατάλληλη μέθοδος που μπορεί να χρησιμοποιηθεί στον τομέα, για να συνεισφέρει και στα δύο πρόβλήματα ανισορροπίας τάξης και οπτικοποίησης χαρακτηριστικών.

Αυτή η προσέγγιση ακολουθεί την εγκληματολογία της ασταθούς μνήμης, για τη δημιουργία ενός ισχυρού συστήματος ενάντια σε κακόβουλο λογισμικό χωρίς αρχεία. Έτσι, έχουν χρησιμοποιηθεί αποθήκες μνήμης των διεργασιών ως κύρια πηγή πληροφοριών. Το σύνολο δεδομένων περιέχει περιπτώσεις για 10 οικογένειες κακόβουλου λογισμικού, συμπεριλαμβανομένων καλοήθων δειγμάτων. Επιπλέον, διερευνήθηκε ο αντίκτυπος των διάφορων σχημάτων απόδοσης εικόνων και διαπιστώθηκε ότι το πλάτος στήλης 4096 px αποδίδει τα καλύτερα αποτελέσματα. Τα αποτελέσματα έδειξαν ότι κατά τη μετατροπή των αρχείων μνήμης σε αρχικές εικόνες, έχοντας μεγαλύτερες στήλες (δηλαδή μεγαλύτερες τιμές πλάτους), έχουν ως αποτέλεσμα καλύτερες αναπαραστάσεις. Επιπλέον, η όψιμη συγχώνευση των GIST και τα χαρακτηριστικά HOG, αποφέρουν το καλύτερο αποτέλεσμα. Κατά συνέπεια, η προτεινόμενη προσέγγιση με ακρίβεια 96.39% δείχνει πολλά υποσχόμενα αποτελέσματα, προσφέροντας επίσης έναν εύλογο χρόνο ανίχνευσης, περίπου στα 3.56 δευτερόλεπτα (Bozkir, Tahillioglu, Aydos, & Kara, 2021).

Πίνακας 3

Άρθρο 2: Ένα νέο πλαίσιο για κακόβουλο λογισμικό που βασίζεται σε εικόνες

No	Πηγή	Πρόβλημα	Λύση	Μεθοδολογία	Αποτελέσματα
2	(Jian, Kuang, Ren, Ma, & Wang, 2021)	Κακόβουλο λογισμικό	Πλαίσιο οικογενειακής ταξινόμησης με βάση την τεχνολογία οπτικοποίησης και μετατροπή στο αντίστοιχο αρχείο bytes και asm με τεχνολογία αποσυναρμολόγησης	CLAHE Νευρωνικό δίκτυο SERLA	Ανωτερότητα της SERLA

Στην μελέτη των Jian, Kuang, Ren, Ma, & Wang (2021), προτείνεται ένας πλήρης εντοπισμός κακόβουλου λογισμικού με ένα πλαίσιο οικογενειακής ταξινόμησης, με βάση την τεχνολογία οπτικοποίησης, που μετατρέπεται στο αντίστοιχο αρχείο bytes και αρχείο asm, με τεχνολογία αποσυναρμολόγησης. Επιπλέον, τρεις πίνακες που δημιουργούνται από το δυαδικό αρχείο, το byte word vector και το opcode word vector, χρησιμοποιούνται αντίστοιχα ως τα τρία κανάλια

εικόνας RGB και συγχωνεύονται με τη σειρά. Τέλος, η τεχνολογία αύξησης δεδομένων CLAHE, χρησιμοποιείται για την αύξηση των εικόνων RGB τριών καναλιών και όλες οι επαυξημένες εικόνες χρησιμοποιούνται για την εκπαίδευση του, προτεινόμενου βάθους, νευρωνικού δικτύου SERLA. Συγκρίνεται με τέσσερα είδη ουδέτερων δικτύων που έχουν χρησιμοποιηθεί ευρέως στην αναγνώριση κακόβουλου λογισμικού.

Χρησιμοποιείται λοιπόν, ένα οπτικό πλαίσιο ανίχνευσης κακόβουλου λογισμικού με βάση τα βαθιά νευρωνικά δίκτυα. Πρώτα απ' όλα, το πλαίσιο χρησιμοποιεί την τεχνολογία *dissemination*, για να δημιουργήσει το αρχείο bytes και το αρχείο asm που αντιστοιχεί σε κάθε εκτελέσιμο αρχείο και χρησιμοποιεί έναν ειδικό πίνακα με μέθοδο παραγωγής, για τη δημιουργία τριών τετραγωνικών μητρώων 256×256 . Στη συνέχεια, οι τρεις πίνακες χρησιμοποιούνται αντίστοιχα ως τα τρία κανάλια μιας εικόνας RGB και συγχωνεύονται σε μια οπτικοποιημένη έγχρωμη εικόνα. Επιπλέον, η μέθοδος αύξησης δεδομένων CLAHE (Contrast Limited Adaptive Histogram Equalization), χρησιμοποιείται για την εκτέλεση προσαρμοστικής επεξεργασίας εξισορρόπησης ιστογράμματος σε κάθε εικόνα RGB. Σε συνδυασμό με τη χρήση της μεθόδου υπερδειγματοληψίας για εκπαίδευση, τελικά αποκτώνται τα νευρωνικά δίκτυα, δηλαδή τα εκπαιδευμένα μοντέλα που χρησιμοποιούνται για τον εντοπισμό κακόβουλου λογισμικού και την ταξινόμηση οικογένειας.

Τα πειραματικά αποτελέσματα ανίχνευσης κακόβουλου λογισμικού και των δύο και η οικογενειακή ταξινόμηση, δείχνουν ότι το μοντέλο έχει πολύ καλή εκτέλεση. Εξάλλου, συγκρίνοντας τις επιδόσεις του SERLA με τις σύγχρονες μεθόδους ταξινόμησης της οικογένειας κακόβουλου λογισμικού, η οποία χρησιμοποιεί το σύνολο δεδομένων BIG 2015, τα αποτελέσματα της αξιολόγησης αποδεικνύουν την ανωτερότητα της SERLA. Λόγω της επιρροής της μεθόδου CBOW, η ταχύτητα δημιουργίας εικόνας είναι αργή χωρίς τη χρήση πολλαπλών νημάτων. Η πολυπλοκότητα του δικτύου SERLA, κάνει το μοντέλο εκπαίδευσης να λειτουργεί σε μεγάλο χρονικό διάστημα, αν και η ταχύτητα πρόβλεψης ενός δείγματος λογισμικού είναι γρήγορη (Jian, Kuang, Ren, Ma, & Wang, 2021).

Πίνακας 4

Άρθρο 3: A survey on android malware detection techniques using machine learning algorithms

No	Πηγή	Πρόβλημα	Λύση	Μεθοδολογία	Αποτελέσματα
3	(Alqahtani, Zagrouba , & Almuhaideb, 2018)	Εντοπισμός κακόβουλου λογισμικού σε Android συσκευή	Συνδυασμοί αλγορίθμων	Εξέταση υπαρχόντων συστημάτων εντοπισμού κακόβουλου λογισμικού, που βασίζονται σε μηχανική μάθηση	OneR και J48: Ακρίβεια 100% με ψευδώς θετικό ποσοστό 83%. Αύξηση ρυθμού ανίχνευσης NB: Ακρίβεια 86,7% και ποσοστό ανίχνευσης 91,5% Multi NB: Ακρίβεια 96.88% και ποσοστό ανίχνευσης 97.76% Deep Neural Networks με QDBP: Ακρίβεια 99,63% και ποσοστό ανίχνευσης 83%

Η μελέτη των Alqahtani Zagrouba & Almuhaideb (2018) αποτελεί μία βιβλιογραφική ανασκόπηση κι ερευνά, για το κατά πόσο ο συνδυασμός λογισμικών έχει αποφέρει υψηλή προστασία στα λογισμικά Android. Η χρήση Deep Neural Networks στο TSDNN μαζί με το QDBP, έδειξε υψηλή ακρίβεια με 99.63% και ποσοστό ανίχνευσης με 83%. Ο παραλληλισμός του TSDNN μαζί με το QDBP, δείχνει την αποτελεσματικότητα των αλγορίθμων και βελτιώνει την ανίχνευση κακόβουλου λογισμικού με μεγάλη ακρίβεια. Επίσης, ο παραλληλισμός ενός συνδυασμού αλγορίθμων OneR και J48, χρησιμοποιώντας το σχήμα AvgProb, αυξάνει το ρυθμό ανίχνευσης και έτσι αποδεικνύει την αποτελεσματικότητα των αλγορίθμων αυτών. Η χρήση NB, δείχνει ακρίβεια 86.7% και ποσοστό ανίχνευσης 91.5%, ενώ η χρήση του Multi NB, δείχνει καλά αποτελέσματα με ακρίβεια 96.88% και ποσοστό ανίχνευσης 97.76%.

Το κύριο μέλημα στην πλατφόρμα Android, όπου οι χρήστες του ολοένα και αυξάνονται, είναι η ανίχνευση του κακόβουλου λογισμικού. Αυτή η εργασία εξετάζει υπάρχοντα συστήματα εντοπισμού κακόβουλου λογισμικού που βασίζονται σε

μηχανική μάθηση και πλαίσια όπως «πλατφόρμα ασφάλειας για κινητά», «ευέλικτο πλαίσιο βασισμένο σε μηχανική μάθηση» και TSDNN μαζί με το QDBP. Τα περισσότερα χρησιμοποιούν SVM και NB αλγόριθμους. Η υψηλότερη ακρίβεια ήταν 100%, με 0.00% ψευδώς θετικό ποσοστό, που λήφθηκε από τους αλγόριθμους OneR και J48. Υπάρχει ποσοστό ανίχνευσης 83% και 90% αντίστοιχα. Σε μια κρίσιμη στιγμή, όπως για παράδειγμα τώρα, στην ανίχνευση κακόβουλου λογισμικού, είναι απαραίτητος ένας πολύ καλός αλγόριθμος μηχανής μάθησης, για να επιτευχθεί υψηλή ακρίβεια και ρυθμός ανίχνευσης. Καταλήγει όμως, ότι ενσωματώνοντας μία ή δύο καλές τεχνικές μαζί, θα αυξηθεί η αποτελεσματικότητα και ο ρυθμός ακρίβειας και ανίχνευσης, στο μέγιστο (Alqahtani, Zagrouba , & Almuhaideb, 2018).

Πίνακας 5

Άρθρο 4: Χρήση της εικόνας Rgb του κώδικα μηχανής για την ταξινόμηση του κακόβουλου λογισμικού

No	Πηγή	Πρόβλημα	Λύση	Μεθοδολογία	Αποτελέσματα			
4	(Chen, Jia, Zhao, & Zhang, 2020)	Ταξινόμηση κακόβουλου λογισμικού	Μετρείται το αληθές θετικό ποσοστό (TPR), το ψευδές θετικό ποσοστό (FPR) και η ακρίβεια (Accuracy)	Μέθοδος ταξινόμησης κακόβουλου λογισμικού, που βασίζεται σε εικόνα RGB και συνελκτικό νευρωνικό δίκτυο GIST 512 διαστάσεων και KNN ταξινόμησης	Experiment	Accuracy (%)	TPR (%)	FPR (%)
					1	97.25	97.45	3.922
					2	97.35	97.82	5.056
					3	97.44	96.70	4.402
					4	97.54	97.82	6.179
					5	97.73	98.25	6.180

Τα πειραματικά δεδομένα στην εργασία των Chen, Jia, Zhao, & Zhang (2020), προέρχονται από το κακόβουλο λογισμικό Classification Challenge στο Kaggle, στη Microsoft 2015. Έχει 10,868 δείγματα κακόβουλου λογισμικού και περιλαμβάνει 9 μεγάλα κακόβουλα προγράμματα με δυαδικά αρχεία και αρχεία αποσυναρμολόγησης. Για να αξιολογηθεί η ικανότητα του ταξινομητή, μετρείται το αληθές θετικό ποσοστό (TPR), το ψευδές θετικό ποσοστό (FPR) και η ακρίβεια (Accuracy). Το TPR, είναι το ποσοστό των δειγμάτων κακόβουλου λογισμικού που ταξινομείται σωστά. Το FPR, είναι το ποσοστό των δειγμάτων κακόβουλου λογισμικού που ταξινομούνται λάθος. Επαναλήφθηκε 5 φορές τυχαία δειγματοληψία και λήφθηκαν τα αποτελέσματα 5 πειραμάτων. Είναι γνωστό ότι η ακρίβεια της μεθόδου μας είναι μεταξύ 97% ~ 98%, και η μέση ακρίβεια είναι 97.462%, με την υψηλότερη ακρίβεια να είναι 97.73%.

Για σύγκριση, επιλέγεται η μέθοδος που προτείνεται από Nataraj et al. (2011), το οποίο υιοθετεί χαρακτηριστικά GIST κακόβουλου λογισμικού, εικόνες και ταξινόμηση KNN (K=3). Ο αλγόριθμος GIST 512 διαστάσεων και KNN, χρησιμοποιούνται για την ταξινόμηση των δειγμάτων. Ο αλγόριθμος επαναλαμβάνει 5 φορές για τυχαία δειγματοληψία. Στο τέλος, η μέση ακρίβεια του αλγόριθμου GIST-KNN στο δοκιμαστικό σύνολο 9 οικογενειών, είναι 95.595%. Αποκτά την καλύτερη ακρίβεια του αλγόριθμου GIST-KNN 95.974%. Σε σύγκριση με άλλες, η μέθοδος αυτή έχει πολύ μεγαλύτερη ακρίβεια. Με βάση αυτή τη μέθοδο λοιπόν, τα δείγματα μπορούν να ταξινομηθούν με ακρίβεια και η ανάλυση και η αποτελεσματικότητά τους μπορεί να βελτιωθεί.

Πίνακας 6

Άρθρο 5: Ταξινόμηση εικόνων κακόβουλου λογισμικού με μοντέλα συνελκτικού νευρωνικού δικτύου

No	Πηγή	Πρόβλημα	Λύση	Μεθοδολογία	Αποτελέσματα
5	(Bensaoud, Abudawood, & Kalita, 2020)	Αυξημένες απειλές κακόβουλου λογισμικού	Προσεγγίσεις για την αυτόματη ανίχνευση και ταξινόμηση κακόβουλου λογισμικού	Μέθοδος CNN για την ταξινόμηση κακόβουλου λογισμικού, θεωρώντας τα δυαδικά προγράμματα κακόβουλου λογισμικού ως εικόνες	Το μοντέλο Inception V3 με ακρίβεια δοκιμής 99.24%, η οποία είναι καλύτερη από την ακρίβεια 98.52% που επιτυγχάνεται με την τρέχουσα κατάσταση, στο σύστημα που ονομάζεται μοντέλο M - CNN

Λόγω των αυξανόμενων απειλών από κακόβουλο λογισμικό (malware), τόσο σε αριθμό όσο και σε πολυπλοκότητα, οι ερευνητές έχουν αναπτύξει πολλές προσεγγίσεις για την αυτόματη ανίχνευση και ταξινόμηση κακόβουλου λογισμικού, αντί να αναλύουν μεθόδους για αρχεία κακόβουλου λογισμικού με μη αυτόματο τρόπο με χρονοβόρα προσπάθεια. Την ίδια στιγμή, οι δημιουργοί κακόβουλου λογισμικού έχουν αναπτύξει τεχνικές, για να αποφύγουν τις τεχνικές ανίχνευσης που βασίζονται σε υπογραφές που χρησιμοποιούνται από εταιρείες antivirus. Πιο πρόσφατα, η βαθιά μάθηση χρησιμοποιήθηκε στην ταξινόμηση κακόβουλου λογισμικού, για την επίλυση αυτού του ζητήματος.

Πολλά προγράμματα προστασίας από ιούς βασίζονται σε τεχνικές βαθιάς εκμάθησης, για την προστασία των συσκευών από κακόβουλο λογισμικό. Οι αρχιτεκτονικές εκμάθησης έχουν επιτύχει καλές επιδόσεις, όταν χρησιμοποιείται με δυαδικά αρχεία Windows PE. Η μελέτη των Bensaoud, Abudawood, & Kalita (2020), παρουσίασε τη σύγκριση απόδοσης μεταξύ έξι ταξινομητών σε ένα σύνολο δεδομένων εικόνων κακόβουλου λογισμικού, που δημιουργήθηκαν από αρχεία PE.

Χρησιμοποίησαν τα μοντέλα από το ImageNet Large Scale Visual Recognition Challenge και άλλα τρία CNN μοντέλα, για την ταξινόμηση εικόνων κακόβουλου λογισμικού σε κλίμακα του γκρι. Τα αποτελέσματα υποδεικνύουν ότι το μοντέλο Inception-V3 υπερτερεί όλων των συγκριτικών εργασιών. Οι συγκρίσεις επίσης, δείχνουν ότι τα καταφέρει με ακρίβεια δοκιμής 99.24%, η οποία είναι καλύτερη από την ακρίβεια 98.52% που επιτυγχάνεται με την τρέχουσα κατάσταση στο σύστημα που ονομάζεται μοντέλο M-CNN.

Πίνακας 7

Άρθρο 6: Ισχυρότητα κακόβουλου λογισμικού Android που βασίζεται σε εικόνα - Ανίχνευση κάτω από αντίθετες επιθέσεις

No	Πηγή	Πρόβλημα	Λύση	Μεθοδολογία	Αποτελέσματα
6	(Darwaish, Naït-Abdesselam, Titouna, & Sattar, 2021)	Ανίχνευση κάτω από αντίθετες επιθέσεις. Καινούριες τεχνολογίες των χάκερ	Νέες τεχνικές ML	(α) Έξυπνος μετασχηματισμός του APK σε εικόνα RGB και ταξινόμηση CNN (β) Δημιουργία καινοφανών επιθέσεων αντιπάλου και ο αντίκτυπός τους στη προσέγγιση των ερευνητών, έναντι δύο άλλων προσεγγίσεων που εφαρμόστηκαν: (DroidDector και DeepClassifyDroid)	Αυτό το σύστημα απέφυγε κατά 13% επιθέσεις, ενώ το ποσοστό διαφυγής για DroidDetector και DeepClassifyDroid ήταν πάνω από 50%

Ο συναρπαστικός ρυθμός της καινοτομίας του smartphone και η τεράστια διάδοση στην καθημερινή ζωή, έχει επιβάλλει σοβαρές απειλές για την ασφάλεια. Η μεγαλύτερη αγορά Android ανοιχτού κώδικα είναι επίσης αιτία για δημιουργούς

κακόβουλου λογισμικού. Από την τελευταία δεκαετία, η μηχανική μάθηση (ML) έχει κερδίσει μεγάλη έλξη, έχει αναπτυχθεί με επιτυχία και προσφέρει αξεπέραστη ευελιξία για την ανίχνευση αυτοματοποιημένου κακόβουλου λογισμικού. Καθώς οι προσεγγίσεις που βασίζονται σε ML υιοθετούνται και αναπτύσσονται ευρέως, οι χάκερ δημιουργούν νέες τεχνολογίες για να αποφύγουν τους ταξινομητές, περνώντας συστήματα ανίχνευσης κακόβουλου λογισμικού Android. Οι ταξινομητές κακόβουλου λογισμικού Android παρουσιάζουν όλο και νεότερες τεχνολογίες για την καταπολέμηση των χάκερ.

Τα συστήματα ανίχνευσης κακόβουλου λογισμικού που βασίζονται στη μηχανική μάθηση, έχουν πετύχει απaráμιλλη αναγνώριση, για την προστασία των χρηστών smartphone από παράνομη πρόσβαση. Ωστόσο, τα συστήματα που βασίζονται στη μάθηση είναι ευάλωτα σε αντίθετες επιθέσεις. Οι δημιουργοί κακόβουλου λογισμικού μπορούν να προσθέσουν μη αντιληπτό υλικό (που προκαλεί διαταραχές) σε πραγματική είσοδο και έτσι μπορούν εύκολα να παραπλανήσουν τον ταξινομητή - στόχο.

Στην εργασία των Darwaish, Naït-Abdesselam, Titouna, & Sattar (2021), παρουσιάζεται ένα σύστημα ανίχνευσης κακόβουλου λογισμικού Android, το οποίο είναι ανθεκτικό σε αντίθετες επιθέσεις, σε σύγκριση με άλλες προσεγγίσεις που βασίζονται στη μηχανική μάθηση. Τα αποτελέσματα έδειξαν την ανθεκτικότητα και των τριών ανιχνευτών κακόβουλου λογισμικού Android, κάτω από δύο νέες αντίπαλες επιθέσεις. Το σύστημα αποδείχθηκε περισσότερο εύρωστο έναντι των αντίπαλων διαταραχών και το απέφυγε κατά 13%, ενώ το ποσοστό διαφυγής ήταν πάνω από 50% για το DroidDetector και το DeepClassifyDroid.

Πίνακας 8

Άρθρο 7: Ανίχνευση κακόβουλου λογισμικού Android βάσει δυναμικών δικαιωμάτων με χρήση τεχνικών μηχανικής εκμάθησης

No	Πηγή	Πρόβλημα	Λύση	Μεθοδολογία	Αποτελέσματα
7	(Mahindru & Singh, 2017)	Επιθέσεις κακόβουλου λογισμικού	Ταξινόμηση ML	Τεχνικές ταξινόμησης μηχανικής μάθησης Naive Bayes (NB), Decision Tree (J48), Random Forest (RF), Simple Logistic (SL) και k-star	Η ακρίβεια ταξινόμησης κακόβουλου λογισμικού των RF, J48 και SL είναι συγκρίσιμη. Η SL αποδίδει (οριακά) καλύτερα από τις άλλες τεχνικές

Το Android είναι μακράν το πιο ευρέως χρησιμοποιούμενο λειτουργικό σύστημα των κινητών τηλεφώνων. Ωστόσο, οι εφαρμογές που βασίζονται στο Android είναι εξαιρετικά ευάλωτες σε διάφορους τύπους επιθέσεων κακόβουλου λογισμικού, που αποδίδονται στον ανοιχτό χαρακτήρα τους και στην υψηλή δημοτικότητά τους στην αγορά. Το σφάλμα βρίσκεται στο μοντέλο άδειας των εφαρμογών Android. Αυτές οι εφαρμογές χρειάζονται έναν αριθμό ευαίσθητων αδειών κατά την εγκατάσταση και το χρόνο εκτέλεσής τους, γεγονός που επιτρέπει πιθανές παραβιάσεις ασφαλείας από κακόβουλο λογισμικό.

Η εργασία των Mahindru & Singh (2017), εξάγει ένα σύνολο 123 δυναμικών αδειών από 11,000 εφαρμογές Android σε ένα μεγαλύτερο (δημοσίως) διαθέσιμο σύνολο δεδομένων μέχρι σήμερα και αξιολογεί έναν αριθμό τεχνικών ταξινόμησης μηχανικής μάθησης, όπως Naive Bayes (NB), Decision Tree (J48), Random Forest (RF), Simple Logistic (SL) και k-star στο πρόσφατα σχεδιασμένο σύνολο δεδομένων, για τον εντοπισμό κακόβουλων εφαρμογών Android. Τα πειραματικά αποτελέσματα δείχνουν ότι, αν και η ακρίβεια ταξινόμησης κακόβουλου λογισμικού των RF, J48 και SL είναι συγκρίσιμη, η SL αποδίδει οριακά καλύτερα από τις άλλες τεχνικές.

Πίνακας 9

Άρθρο 8: AdDroid: Πλαίσιο μηχανικής μάθησης βάσει κανόνων για ανάλυση κακόβουλου λογισμικού Android

No	Πηγή	Πρόβλημα	Λύση	Μεθοδολογία	Αποτελέσματα
8	(Mehtab, και συν., 2020)	Κακόβουλη συμπεριφορά σε εφαρμογές Android	Πλαίσιο AdDroid για ανάλυση και εντοπισμό	Χρήση συνόλου δεδομένων που περιλαμβάνει 1,420 εφαρμογές Android, με 910 κακόβουλες και 510 καλοήθειες εφαρμογές	Ακρίβεια 99.11%, με ποσοστό Αληθές Θετικό (TP) 98.61% και Αληθές Αρνητικό (TN) 99.33%

Τα τελευταία χρόνια σημειώθηκε τεράστια πρόοδος στην ανάπτυξη κακόβουλου λογισμικού Android. Η κολοσσιαία εξάρτηση από εφαρμογές Android για την καθημερινή εργασία και η μαζική ανάπτυξή τους, υπαγορεύει έναν αυτοματοποιημένο μηχανισμό διάκρισης των κακόβουλων εφαρμογών από τις καλοήθειες. Ένας σημαντικός όγκος έρευνας έχει αφιερωθεί στην ανάλυση και τον μετριασμό αυτού του ολοένα και αυξανόμενου προβλήματος. Ωστόσο, οι εισβολείς χρησιμοποιούν πιο περίπλοκες τεχνικές για να αποφύγουν τον εντοπισμό τους.

Η εργασία του Mehtab et al. (2020) προτείνει ένα πλαίσιο μηχανικής μάθησης, το οποίο ονομάζεται AdDroid, για την ανάλυση και τον εντοπισμό κακόβουλης συμπεριφοράς σε εφαρμογές Android, που βασίζονται σε διάφορους συνδυασμούς τεχνουργημάτων που ονομάζονται κανόνες. Τα τεχνουργήματα αντιπροσωπεύουν ενέργειες μιας εφαρμογής Android, όπως σύνδεση στο Διαδίκτυο, ανέβασμα αρχείου σε απομακρυσμένο διακομιστή ή εγκατάσταση άλλου πακέτου στη συσκευή κ.λπ.. Το AdDroid χρησιμοποιεί μια τεχνική μηχανικής εκμάθησης, η οποία βασίζεται σε σύνολο όπου το Adaboost συνδυάζεται με παραδοσιακούς ταξινομητές, για να εκπαιδεύσει ένα μοντέλο που βασίζεται σε στατική ανάλυση εφαρμογών Android που είναι ικανό να αναγνωρίζει κακόβουλες εφαρμογές. Οι τεχνικές επιλογής χαρακτηριστικών και εξαγωγής, χρησιμοποιούνται για να λάβει τους πιο διακριτικούς κανόνες.

Το προτεινόμενο μοντέλο δημιουργείται χρησιμοποιώντας ένα σύνολο δεδομένων που περιλαμβάνει 1,420 εφαρμογές Android, με 910 κακόβουλες και 510 καλοήθειες εφαρμογές. Το προτεινόμενο σύστημα πέτυχε ακρίβεια 99,11%, με ποσοστό Αληθινό Θετικό (TP) 98.61% και Αληθινό Αρνητικό (TN) 99.33%. Τα υψηλά ποσοστά TP και TN αντικατοπτρίζουν την αποτελεσματικότητα τόσο στην κύρια, όσο και στην ελάσσονα κατηγορία. Δεδομένου ότι η προτεινόμενη λύση έχει εξαιρετικά χαμηλή υπολογιστική πολυπλοκότητα, καθιστά δυνατή την ανάλυση εφαρμογών σε πραγματικό χρόνο.

Πίνακας 10

Άρθρο 9: Ένα σύστημα ανίχνευσης κακόβουλου λογισμικού Android βασισμένο σε μηχανική μάθηση

No	Πηγή	Πρόβλημα	Λύση	Μεθοδολογία	Αποτελέσματα
9	(Wen & Yu, 2017)	Κακόβολο λογισμικό Android	Σύστημα ανίχνευσης, βασισμένο σε ML	PCA RELIEF PCA-RELIEF	Η μέθοδος PCA-RELIEF αποδίδει καλύτερα, με υψηλότερο ρυθμό ανίχνευσης και χαμηλότερο ρυθμό ανίχνευσης σφαλμάτων, σε σύγκριση με τις παραδοσιακές προσεγγίσεις ανίχνευσης

Η ευκολία της πλατφόρμας Android είναι κάτι που έχει παρακινήσει την ανάπτυξη κακόβουλου λογισμικού. Η παραδοσιακή μέθοδος που εντοπίζει το κακόβουλο λογισμικό με βάση την υπογραφή, δεν μπορεί να εντοπίσει άγνωστες εφαρμογές. Η εργασία των Wen & Yu (2017), χρησιμοποιεί ένα σύστημα μηχανικής μάθησης που είναι σε θέση να αναγνωρίζει κακόβουλο λογισμικό σε συσκευές Android. Σε αυτό το σύστημα εξάγονται χαρακτηριστικά που βασίζονται στη στατική, καθώς και τη δυναμική ανάλυση και στη συνέχεια σε μία νέα προσέγγιση επιλογής χαρακτηριστικών, βασισμένη στην αρχή ανάλυσης συστατικών (PCA). Το μοντέλο θα

κατασκευάζεται με μηχανή υποστήριξης διανυσμάτων (SVM) για ταξινόμηση. Τα πειραματικά αποτελέσματα έδειξαν ότι αυτό το σύστημα παρέχει μια αποτελεσματική μέθοδο για τον εντοπισμό κακόβουλου λογισμικού Android.

Πίνακας 11

Άρθρο 10: Ταξινόμηση κακόβουλου λογισμικού Android με τη βοήθεια μηχανικής μάθησης

No	Πηγή	Πρόβλημα	Λύση	Μεθοδολογία	Αποτελέσματα
10	(Milosevic, Dehghantanha, & Choo, 2017)	Κακόβουλο λογισμικό σε Android	Ταξινόμηση κακόβουλου λογισμικού Android με τη βοήθεια ML	Αλγόριθμοι C4.5 decision tree + random tree + random forests Logistic regression + C4.5 + SVM with SMO Random tree + Random Forest + SVM with SMO SVM with SMO + Logistic regression + Random forest SVM with SMO + Logistic regression + AdaBoostM1 with SVM base Logistic regression + JRip + Random Forests + C4.5 + SVM with SMO SVM with SMO + Logistic regression + Simple Logistic regression + AdaBoostM1 with SVM base	Η μέθοδος που βασίζεται στην άδεια, ήταν σε θέση να ταξινομήσει το κακόβουλο από το καλό λογισμικό στο 89% των περιπτώσεων, ενώ η απόδοση ταξινόμησης ανάλυσης πηγαίου κώδικα ήταν πάνω από 95%. Ποσοστά ακρίβειας 95.1% με χρήση SVM και 95.6% χρησιμοποιώντας τη μέθοδο εκμάθησης συνόλου. Είναι συγκρίσιμες με τις υπάρχουσες λύσεις τελευταίας τεχνολογίας

Η ευρεία υιοθέτηση συσκευών Android και η ικανότητά τους να έχουν πρόσβαση σε σημαντικές ιδιωτικές και εμπιστευτικές πληροφορίες, είχαν ως αποτέλεσμα αυτές οι συσκευές να στοχοποιούνται από προγραμματιστές κακόβουλου λογισμικού. Οι υπάρχουσες τεχνικές ανάλυσης κακόβουλου λογισμικού Android μπορούν να κατηγοριοποιηθούν ευρέως σε στατική και δυναμική ανάλυση. Στην εργασία των Milosevic, Dehghantanha, & Choo (2017), παρουσιάζονται δύο προσεγγίσεις εφαρμογών με τη βοήθεια μηχανικής μάθησης για στατική ανάλυση κακόβουλου λογισμικού Android. Η πρώτη προσέγγιση βασίζεται σε δικαιώματα και το άλλο βασίζεται σε ανάλυση πηγαίου κώδικα χρησιμοποιώντας ένα μοντέλο bag-of-words. Το μοντέλο βασίζεται σε άδειες που είναι υπολογιστικά φθηνές και εφαρμόζονται ως χαρακτηριστικό της εφαρμογής Android OWASP Seraphimdroid, που μπορούν να ληφθούν από το Google Play Store. Οι αξιολογήσεις και για τις δύο προσεγγίσεις, δείχνουν βαθμολογία 95.1% και μέτρηση 89% για τα μοντέλα ταξινόμησης βάσει πηγαίου κώδικα και βάσει αδειών, αντίστοιχα.

Πίνακας 12

Άρθρο 11: Ανίχνευση κακόβουλου λογισμικού Android μέσω μηχανικής εκμάθησης σε δομές πυρήνα

No	Πηγή	Πρόβλημα	Λύση	Μεθοδολογία	Αποτελέσματα
11	(Wang & Li, 2021)	Υπαρξη κακόβουλου λογισμικού σε πλατφόρμες Android	Ο εντοπισμός κακόβουλου λογισμικού σε πλατφόρμες Android	WBD	Ακρίβεια 94%-98% με ποσοστό ψευδώς θετικών 2%-7%

Με την εμφάνιση των έξυπνων τηλεφώνων, η δημοτικότητα των δωρεάν εφαρμογών Android έχει αυξηθεί ραγδαία. Αυτό οδήγησε στην ακούσια εγκατάσταση κακόβουλων εφαρμογών Android, οι οποίες παραβιάζουν το απόρρητο των χρηστών. Ο εντοπισμός κακόβουλου λογισμικού σε πλατφόρμες Android είναι επομένως μια

αυξανόμενη ανησυχία, λόγω της ανεπιθύμητης ομοιότητας μεταξύ κακόβουλης συμπεριφοράς και καλοήθους συμπεριφοράς, η οποία μπορεί να οδηγήσει σε αργοπορημένο εντοπισμό, επιτρέποντας να διατηρούνται για συγκριτικά μεγάλα χρονικά διαστήματα σε μολυσμένα τηλέφωνα.

Η εργασία των Wang & Li (2021), ακολουθεί μία μέθοδος που διαθέτει ανίχνευση βάσει βάρους (WBD), που έχει σχεδιαστεί για να κατανοεί και να κατηγοριοποιεί τα χαρακτηριστικά του κακόβουλου λογισμικού Android και των καλοήθων εφαρμογών. Επιπλέον, ο πράκτορας λογισμικού τους υλοποιείται για τη συλλογή και αποθήκευση δεδομένων, για αυτόματη σάρωση χιλιάδων καλοήθων και κακόβουλων εφαρμογών. Εξετάστηκαν 112 χαρακτηριστικά πυρήνα για την εκτέλεση της δομής δεδομένων εργασιών στο σύστημα Android και αξιολογήθηκε η ακρίβεια ανίχνευσης με έναν αριθμό συνόλων δεδομένων διαφόρων διαστάσεων. Βρέθηκε ότι η μνήμη και τα χαρακτηριστικά που σχετίζονται με το σήμα, συμβάλλουν σε πιο ακριβή ταξινόμηση από ό,τι οι σχετικές με το χρονοδιάγραμμα και άλλοι περιγραφείς καταστάσεων εργασιών. Αυτή η μέθοδος μπορεί να επιτύχει ακρίβεια 94% - 98% και ποσοστό ψευδώς θετικών 2% – 7%, ενώ εντοπίζονται εφαρμογές κακόβουλου λογισμικού με χαρακτηριστικά μειωμένων διαστάσεων, που συντομεύουν τις διαδικτυακές ανιχνεύσεις κακόβουλου λογισμικού και προχωρούν σε επιθεωρήσεις κακόβουλου λογισμικού εκτός σύνδεσης.

Πίνακας 13

Άρθρο 12: Μια προσέγγιση μηχανικής εκμάθησης για τον εντοπισμό κακόβουλου λογισμικού Android

No	Πηγή	Πρόβλημα	Λύση	Μεθοδολογία	Αποτελέσματα
12	(Sahs & Khan, 2012)	Κακόβουλο λογισμικό σε συσκευές Android	ML για τον εντοπισμό κακόβουλου λογισμικού σε συσκευές Android	One Class Support Vector Machine	Καλοήθη παραδείγματα από κακόβουλα λογισμικά

Με την πρόσφατη εμφάνιση των κινητών πλατφορμών, υπάρχει ένας αυξανόμενος κίνδυνος που σχετίζεται με το κακόβουλο λογισμικό, που στοχεύει σε κινητές συσκευές. Το πρόβλημα της ανίχνευσης τέτοιων λογισμικών, είναι οι μοναδικές προκλήσεις, λόγω των περιορισμένων πόρων και τα διαθέσιμα και περιορισμένα προνόμια που παρέχονται στον χρήστη. Σε αυτό το άρθρο, παρουσιάζεται ένα σύστημα που βασίζεται στη μηχανική εκμάθηση, για τον εντοπισμό κακόβουλου λογισμικού σε συσκευές Android. Το σύστημα εξάγει μια σειρά από χαρακτηριστικά και εκπαιδεύει ένα One Class Support Vector Machine χωρίς σύνδεση (εκτός συσκευής), προκειμένου να αξιοποιηθεί η υψηλότερη υπολογιστική ισχύ ενός διακομιστή ή συμπλέγματος διακομιστών (Sahs & Khan, 2012).

Πίνακας 14

Άρθρο 12: Μια ανασκόπηση των προσεγγίσεων εντοπισμού κακόβουλου λογισμικού Android με βάση τη μηχανική εκμάθηση

No	Πηγή	Πρόβλημα	Λύση	Μεθοδολογία	Αποτελέσματα
13	(LIU, και συν., 2020)	Κακόβουλο λογισμικό σε συσκευές Android	Εντοπισμός κακόβουλου λογισμικού Android	PCA	Η μηχανική μάθηση, έχει βελτιώσει σε μεγάλο βαθμό τις προοπτικές για τον εντοπισμό κακόβουλου λογισμικού Android

Πολλοί ερευνητές έχουν μελετήσει το πρόβλημα ανίχνευσης του κακόβουλου λογισμικού Android και έχουν προτείνει θεωρίες και μεθόδους από διαφορετικές οπτικές γωνίες. Η υπάρχουσα έρευνα προτείνει την μηχανική μάθηση ως έναν αποτελεσματικό και πολλά υποσχόμενο τρόπο εντοπισμού κακόβουλου λογισμικού Android. Παρουσιάζονται εν συντομία κάποια υπόβαθρα για εφαρμογές Android, συμπεριλαμβανομένου της αρχιτεκτονική συστήματος, του μηχανισμού ασφαλείας και την ταξινόμηση κακόβουλου λογισμικού Android. Στη συνέχεια, με τη λήψη

μηχανικής μάθησης ως επίκεντρο, αναλύεται και συνοψίζεται η κατάσταση της έρευνας από βασικές προοπτικές, όπως το δείγμα, η απόκτηση, η προεπεξεργασία δεδομένων, η επιλογή χαρακτηριστικών, τα μοντέλα μηχανικής μάθησης, οι αλγόριθμοι και η αξιολόγηση αποτελεσματικότητας ανίχνευσης (LIU, και συν., 2020).

Πίνακας 15

Άρθρο 14: Μια έρευνα σχετικά με τις τεχνικές ανίχνευσης κακόβουλου λογισμικού Android

No	Πηγή	Πρόβλημα	Λύση	Μεθοδολογία	Αποτελέσματα
14	(Alqahtani, Zagrouba, & Almuhaideb, 2019)	Κακόβουλο λογισμικό	Εντοπισμός κακόβουλου λογισμικού	SVM & NB αλγόριθμοι DNN TSDNN QDBP	Η υψηλότερη ακρίβεια ήταν 100% με 0,00% ψευδώς θετικό ποσοστό που λήφθηκε από τους OneR και J48. Υπάρχει ποσοστό ανίχνευσης 83% και 90% αντίστοιχα. Το σχήμα AvgProb απέκτησε υψηλότερη ακρίβεια και ρυθμό ανίχνευσης από το 90%

Οι χρήστες smartphone αυξάνονται ραγδαία με τα χρόνια και κυρίως αυτοί των Android. Η κυρία ανησυχία στην πλατφόρμα Android, η οποία ολοένα και αυξάνεται, είναι ο εντοπισμός κακόβουλου λογισμικού. Ανάμεσα στις διάφορες υπάρχουσες προσεγγίσεις του, οι αλγόριθμοι που βασίζονται στη μηχανική μάθηση έχουν επιτύχει υψηλή ακρίβεια. Επομένως, στις συσκευές Android πρέπει να παρέχεται η δυνατότητα ανίχνευσης κακόβουλου λογισμικού χρησιμοποιώντας αλγόριθμους μηχανικής εκμάθησης. Πολλοί ερευνητές πρότειναν διάφορα τέτοια συστήματα μηχανικής εκμάθησης όπως SVM, NB και DNN. Η εργασία των Alqahtani, Zagrouba, & Almuhaideb (2019), ερευνά την τελευταία λέξη της τεχνολογίας και πολλές τεχνικές, εστιάζοντας σε ταξινομητές που βασίζονται στη Μηχανική Μάθηση για τον εντοπισμό κακόβουλου λογισμικού σε συσκευές Android.

Πίνακας 16

Άρθρο 15: Ανάλυση της απόδοσης εντοπισμού κακόβουλου λογισμικού Android χρησιμοποιώντας ταξινομητές μηχανικής μάθησης

No	Πηγή	Πρόβλημα	Λύση	Μεθοδολογία	Αποτελέσματα
15	(Ham & Choi, 2013)	Ανίχνευση του κακόβουλου λογισμικού	Χρήση ταξινομητών ML	Random Forest SVM	Από την άποψη του TPR / FPR, η απόδοση του ταξινομητή Random Forest ήταν η πιο κατάλληλη

Πολλές μελέτες έχουν αξιολογήσει την ανίχνευση του κακόβουλου λογισμικού με την απόδοση των ταξινομητών μηχανικής μάθησης, μέσω συλλογής και ανάλυσης συμβάντων, κλήσεων συστήματος και πληροφοριών καταγραφής που δημιουργούνται σε κινητές συσκευές Android. Ωστόσο, η παρακολούθηση περιττών χαρακτηριστικών χωρίς κατανόηση της αρχιτεκτονικής Android και των χαρακτηριστικών κακόβουλου λογισμικού δημιουργούν κατανάλωση πόρων, γενικά έξοδα συσκευών Android και χαμηλή αναλογία εντοπισμού κακόβουλου λογισμικού.

Στο άρθρο των Ham & Choi (2013), προτείνονται νέα σύνολα χαρακτηριστικών που λύνουν το πρόβλημα προηγούμενων μελετών, όσον αφορά τον εντοπισμό κακόβουλου λογισμικού για κινητά και αναλύεται η απόδοση των ταξινομητών με χρήση της μηχανικής μάθησης. Αρχικά, εξετάζονται τα δομικά χαρακτηριστικά του Android και παρακολουθούνται οι δυνατότητες με τέσσερις διαφορετικούς τύπους ταξινομητών μηχανικής μάθησης, όπου εν τέλει γίνεται με ακρίβεια ο εντοπισμός του κακόβουλου λογισμικού. Τα πειραματικά αποτελέσματα έδειξαν ότι η απόδοση του ταξινομητή Random Forest ήταν η πιο κατάλληλη από την άποψη του TPR / FPR. Υπήρξε παρόμοια απόδοση στο TP (True Positive) δεδομένων κανονικού τύπου και FP (False Positive) δεδομένων τύπου κακόβουλου λογισμικού. Αντιθέτως, ο ταξινομητής SVM παρουσίασε πολύ χαμηλή απόδοση σε σύγκριση με αυτή του Random Forest. Εν κατακλείδι, διαπιστώθηκε ότι η απόδοσή του μπορεί να βελτιωθεί, συλλέγοντας περισσότερα σύνολα δεδομένων.

Πίνακας 17

Άρθρο 16: Ανίχνευση κακόβουλου λογισμικού Android με χρήση ταξινομητών παράλληλης μηχανικής εκμάθησης

No	Πηγή	Πρόβλημα	Λύση	Μεθοδολογία	Αποτελέσματα
16	(Yerima, Sezer, & Muttik, 2014)	Κακόβουλο λογισμικό	Παράλληλη προσέγγιση του κακόβουλου λογισμικού	TPR TNR FPR FNR ACC ERR AUC	Η προσέγγιση είναι ιδανική από άποψη απόδοσης, καθώς είναι οικονομικά αποδοτική για την ταξινόμηση μιας νέας εφαρμογής, επειδή: A) χρησιμοποιούνται στατικά χαρακτηριστικά εφαρμογής B) τα επιλεγμένα μοντέλα ταξινόμησης συστατικών, έχουν χαμηλή υπολογιστική απαίτηση κατά την απόφαση ταξινόμησης

Πρόσφατα, έχει εμφανιστεί νέα γενιά οικογενειών κακόβουλου λογισμικού Android με προηγμένες δυνατότητες διαφυγής, χρησιμοποιώντας συμβατικές μεθόδους, που τα κάνουν πολύ πιο δύσκολο να εντοπιστούν. Η εργασία των Yerima, Sezer, & Muttik (2014), προτείνει και διερευνά μια παράλληλη μηχανική μάθηση για να πετύχει την ανίχνευσή του. Η χρήση πραγματικών δειγμάτων κακόβουλου λογισμικού και καλοήθους εφαρμογών, αναπτύσσεται με ένα σύνθετο μοντέλο ταξινόμησης, δηλαδή παράλληλο συνδυασμό ετερογενών ταξινομητών. Η εμπειρική αξιολόγηση του μοντέλου, υπό διαφορετικά συνδυαστικά σχήματα, καταδεικνύει την αποτελεσματικότητά του και τις δυνατότητές του για βελτίωση της ακρίβειας ανίχνευσης. Το πιο σημαντικό, χρησιμοποιώντας διάφορους ταξινομητές με διαφορετικά χαρακτηριστικά, είναι ότι τα δυνατά σημεία μπορούν να αξιοποιηθούν όχι μόνο για ένα βελτιωμένο Android, αλλά και για ταχύτερη ανάλυση white box, δηλαδή των πιο ερμηνεύσιμων συστατικών των ταξινομητών.

Η παράλληλη προσέγγιση ταξινόμησης που προτείνεται, είναι βιώσιμη για το σύστημα, το οποίο παρέχει ένα συμπληρωματικό εργαλείο που όχι μόνο βελτιώνει δυνητικά την ανίχνευση κακόβουλου λογισμικού Android, αλλά και επιτρέπει την αξιοποίηση των δυνατοτήτων των διαφορετικών ταξινομητών. Για παράδειγμα, οι

ταξινομητές που βασίζονται σε κανόνες, μπορούν να παρέχουν αποτέλεσμα, εύκολα ερμηνεύσιμο από τον άνθρωπο, που μπορεί να είναι χρήσιμο, αφού οδηγεί σε περαιτέρω στάδια ανάλυσης. Η προτεινόμενη προσέγγιση είναι ιδανική και αποδοτική και από οικονομικής άποψης, επειδή: 1) χρησιμοποιούνται στατικά χαρακτηριστικά εφαρμογής και 2) κατά την ταξινόμηση, τα επιλεγμένα μοντέλα της έχουν χαμηλή υπολογιστική απαίτηση.

Πίνακας 18

Άρθρο 17: Εφαρμογή ταξινομητών μηχανικής μάθησης σε δυναμική ανίχνευση κακόβουλου λογισμικού Android σε κλίμακα

No	Πηγή	Πρόβλημα	Λύση	Μεθοδολογία	Αποτελέσματα
17	(Amos, Turner, & White, 2013)	Κακόβουλο λογισμικό έξυπνων τηλεφώνων	Ταξινομητές ML	Αξιολόγηση ταξινομητών: Random Forest Naïve Bayes Multilayer Perceptron Bayes net Logistic J48	Όλοι οι ταξινομητές, με κύριο τον Bayes net που σωστά ταξινομεί το υψηλότερο ποσοστό διανυσματικών χαρακτηριστικών, είχαν ψευδώς θετικά ποσοστά, τα οποία κυμαίνονται από 15.86% έως 33.79%

Οι ταξινομητές μηχανικής μάθησης είναι μια τρέχουσα μέθοδος για τον εντοπισμό κακόβουλων εφαρμογών σε συστήματα smartphone. Η εργασία των Amos, Turner, & White (2013), παρουσιάζει την αξιολόγηση ορισμένων υπαρχόντων ταξινομητών, χρησιμοποιώντας ένα σύνολο δεδομένων που περιέχει χιλιάδες πραγματικές (δηλαδή όχι συνθετικές) εφαρμογές. Παρουσιάζεται επίσης το πλαίσιο STREAM, το οποίο αναπτύχθηκε για να επιτρέπει την ταχεία επικύρωση μεγάλης κλίμακας σε κινητά, για ταξινομητές μηχανικής μάθησης προς δυναμική ανίχνευση κακόβουλου λογισμικού Android.

Πίνακας 19

Άρθρο 18: Μια ολοκληρωμένη έρευνα με χρήση τεχνικών μηχανικής μάθησης για την ανίχνευση κακόβουλου λογισμικού Android

No	Πηγή	Πρόβλημα	Λύση	Μεθοδολογία	Αποτελέσματα
18	(Kouliaridis & Kambourakis, 2021)	Επιθέσεις κακόβουλου λογισμικού	Εντοπισμός κακόβουλου λογισμικού	ML	Η πλειονότητα των προσεγγίσεων περιλαμβάνει ένα σύνολο διαφορετικών βασικών παραμέτρων, συμπεριλαμβανομένων: Το σύνολο δεδομένων Την ανάλυση (συλλογή χαρακτηριστικών) Τις μετρήσεις αξιολόγησης ανίχνευσης

Οι επιθέσεις κακόβουλου λογισμικού για κινητά αυξάνονται τόσο σε πολυπλοκότητα, όσο και σε διάχυση. Όσο η πλατφόρμα ανοιχτού κώδικα Android συνεχίζει να κυριαρχεί στην αγορά, οι συντάκτες κακόβουλου λογισμικού τη θεωρούν ως δικό τους προτιμώμενο στόχο. Σχεδόν αυστηρά, προηγμένη λύση ανίχνευσης κακόβουλου λογισμικού για κινητά, είναι η μηχανική. Ωστόσο, τα ευρήματα δείχνουν ξεκάθαρα ότι η πλειονότητα των υπαρχόντων έργων χρησιμοποιεί διαφορετικές μετρήσεις και μοντέλα, καθώς και διαφορετικά σύνολα δεδομένων και χαρακτηριστικά ταξινόμησης, που προκύπτουν από διαφορετικές τεχνικές ανάλυσης, π.χ. στατικές, δυναμικές ή υβριδικές. Αυτό περιπλέκει τη διασταυρούμενη σύγκριση των διαφόρων προτεινόμενων σχημάτων ανίχνευσης και εγείρει αμφιβολίες για τα προκύπτοντα αποτελέσματα.

Η εργασία των Kouliaridis & Kambourakis (2021), επιχειρεί να σχηματοποιήσει τις μέχρι τώρα προσεγγίσεις ανίχνευσης κακόβουλου λογισμικού που υποστηρίζονται από ML (Machine Learning) και τεχνικές με την οργάνωσή τους σε τέσσερις άξονες, δηλαδή: την ηλικία του επιλεγμένου συνόλου δεδομένων, το χρησιμοποιούμενο τύπο ανάλυσης, τις χρησιμοποιούμενες τεχνικές ML και τις

επιλεγμένες μετρήσεις απόδοσης. Εξάλλου, με βάση αυτούς τους άξονες, εισάγεται ένα συγκλίνον σχήμα, που μπορεί να καθοδηγήσει το μελλοντικό κακόβουλο λογισμικό Android, με τεχνικές ανίχνευσης και παρέχουν μια σταθερή βάση για τις πρακτικές μηχανικής μάθησης σε αυτόν τον τομέα. Από τα αποτελέσματα γίνεται φανερό ότι η πλειονότητα των προσεγγίσεων περιλαμβάνει ένα διαφορετικό σύνολο βασικών παραμέτρων, συμπεριλαμβανομένων το σύνολο δεδομένων, την ανάλυση (συλλογή χαρακτηριστικών) και τις μετρήσεις αξιολόγησης ανίχνευσης. Μετριάζοντας αυτό το ζήτημα, προτείνεται ένα σύστημα σύγκλισης τεσσάρων βημάτων, για να χρησιμεύσει ως βάση και εφαλτήριο, για μελλοντικές προσεγγίσεις εντοπισμού κακόβουλου λογισμικού Android που βασίζονται σε ML σε κινητά.

Πίνακας 20

Άρθρο 19: Μηχανική εκμάθηση για ανίχνευση κακόβουλου λογισμικού Android με χρήση αδειών και κλήσεων API

No	Πηγή	Πρόβλημα	Λύση	Μεθοδολογία	Αποτελέσματα
19	(Peiravian & Zhu, 2013)	Κακόβουλο λογισμικό	Εντοπισμός κακόβουλου λογισμικού	Αναλυτής εφαρμογών APK Χαρακτηρισμός κάθε εφαρμογής Σύγκριση κακόβουλου και μη λογισμικού Εκπαίδευση μοντέλων ταξινόμησης Μέθοδος ταξινόμησης SVM	Το δέντρο απόφασης (J48) έχει το χαμηλότερο ποσοστό ανίχνευσης Η μέθοδος Bagging έχει το υψηλότερο

Στην εργασία των Peiravian & Zhu (2013), προτείνεται μια εργασία πλαισίου εκμάθησης βασισμένη σε χαρακτηριστικά, για τον εντοπισμό κακόβουλου λογισμικού Android. Το πλαίσιο εξετάζει τα χαρακτηριστικά της εφαρμογής, συμπεριλαμβανομένων των αδειών και των κλήσεων API, για να χαρακτηρίσει τις

συμπεριφορές εφαρμογών Android. Αποτελείται από τέσσερα κύρια μέρη: i) το πρώτο στοιχείο είναι ένας αναλυτής εφαρμογών που αποσυμπίεζει το αρχείο APK μιας εφαρμογής και εξάγει `AndroidManifest.xml` και αρχεία `class`, τα οποία είναι απαραίτητα για το χαρακτηρισμό των εφαρμογών, ii) το δεύτερο στοιχείο στοχεύει στον χαρακτηρισμό κάθε εφαρμογής με βάση τα ζητούμενα δικαιώματα και τις κλήσεις API, iii) το τρίτο συστατικό, η γεννήτρια χαρακτηριστικών, εκτελεί τη σύγκριση των κορυφαίων και πιο συχνά χρησιμοποιούμενων κλήσεων API σε κακόβουλο λογισμικό και σε καλοήθεις εφαρμογές. Τα αποτελέσματα δείχνουν ξεκάθαρα μοτίβα / διαφορές κλήσης API, με χρήσεις: οι καλοήθεις εφαρμογές τείνουν να χρησιμοποιούν περισσότερες κατηγορίες API από τις εφαρμογές κακόβουλου λογισμικού. Ως αποτέλεσμα αυτής της διαδικασίας, κάθε εφαρμογή αντιπροσωπεύεται από μια μεμονωμένη παρουσία με δυαδική άδεια και λειτουργίες κλήσης API και μια ετικέτα κλάσης υποδεικνύει εάν η εφαρμογή είναι καλοήθης ή κακοήθης. iv) Το τελευταίο μέρος περιλαμβάνει την εκπαίδευση των μοντέλων ταξινόμησης από τα δεδομένα που συλλέχθηκαν. Επειδή το πλαίσιο έχει μετατρέψει τις εφαρμογές σε γενική μορφή, μπορεί να χρησιμοποιηθεί οποιοσδήποτε αλγόριθμος εκμάθησης, για την εξαγωγή μοντέλων ταξινόμησης από τα δεδομένα εκπαίδευσης. Σε αυτή την εργασία χρησιμοποιείται η μέθοδος ταξινόμησης SVM, Δέντρα Αποφάσεων και Bagging.

Τα αποτελέσματα δείχνουν ότι σαν μέθοδος, το δέντρο απόφασης (J48) έχει το χαμηλότερο ποσοστό ανίχνευσης, ενώ το υψηλότερο ποσοστό ανίχνευσης, σε κάθε περίπτωση, ανήκει στη μέθοδο Bagging. Το Bagging έχει πολύ καλή απόδοση σε δεδομένα. Επίσης, αποδεικνύεται ότι έχει την καλύτερη απόδοση στην ταξινόμηση όλων των δημιουργημένων συνόλων δεδομένων (Peiravian & Zhu, 2013).

Πίνακας 21

Άρθρο 20: Σημαντική αναγνώριση άδειας για Android που βασίζεται σε ανίχνευση κακόβουλου λογισμικού με μηχανική μάθηση

No	Πηγή	Πρόβλημα	Λύση	Μεθοδολογία	Αποτελέσματα
20	(Li, et al., 2018)	Ανησυχητικός ρυθμός ανάπτυξης των κακόβουλων εφαρμογών	Μία προσέγγιση ανίχνευσης κακόβουλου λογισμικού που μπορεί να είναι αποτελεσματική και αποδοτική	SigPID SVM MLDP	Το SigPID είναι το πιο αποτελεσματικό, ανιχνεύοντας το 93.62% του κακόβουλου λογισμικού στο σύνολο δεδομένων και το 91.4% από άγνωστα / νέα δείγματα κακόβουλου λογισμικού

Στην μελέτη του Li et al. (2018) εισάγεται το Σημαντικό Αναγνωριστικό Άδειας (SigPID), ένα σύστημα ανίχνευσης κακόβουλου λογισμικού που βασίζεται στη χρήση αδειών και στην ανάλυσή τους, για την αντιμετώπιση της ραγδαίας αύξησης του αριθμού των κακόβουλων λογισμικών Android. Αντί να εξαχθούν και να αναλυθούν όλα τα δικαιώματα Android, αναπτύσσονται τρία επίπεδα εξόρυξης δεδομένων αδειών, για τον εντοπισμό των πιο σημαντικών, που μπορούν να είναι αποτελεσματικές στη διάκριση μεταξύ καλοήθων και κακόβουλων εφαρμογών. Στη συνέχεια, το SigPID χρησιμοποιεί μεθόδους ταξινόμησης, που βασίζονται στη μηχανική μάθηση, διαφορετικών οικογενειών κακόβουλου λογισμικού και καλοηθών εφαρμογών. Σε σύγκριση με άλλες προσεγγίσεις τελευταίας τεχνολογίας, το SigPID είναι το πιο αποτελεσματικό, ανιχνεύοντας το 93,62% του κακόβουλου λογισμικού στο σύνολο δεδομένων και το 91,4% από άγνωστα / νέα δείγματα κακόβουλου λογισμικού.

Πίνακας 22

Άρθρο 21: Ναι, η μηχανική μάθηση μπορεί να είναι πιο ασφαλής! Μια μελέτη περίπτωσης για την ανίχνευση κακόβουλου λογισμικού Android

No	Πηγή	Πρόβλημα	Λύση	Μεθοδολογία	Αποτελέσματα
21	(Demontis, και συν., 2019)	Κακόβουλο λογισμικό	Εντοπισμός κακόβουλου λογισμικού	SVM MCS-SVM Sec-SVM	Η έκδοση του Sec-SVM περιορίζει στενά τη μίμηση καλοήθων δεδομένων κι έτσι αποδίδει το πιο ασφαλές σύστημα

Η εργασία των Demontiset al. (2019), παρουσιάζει μία πειραματική αξιολόγηση ενός αλγορίθμου ασφαλούς μάθησης (Sec-SVM), δοκιμάζοντάς τον σε διαφορετικά σενάρια. Γίνεται σύγκριση της προσέγγισης Sec-SVM με την τυπική υλοποίηση Drebin (σημειώνεται με SVM) και με μια τεχνική που προτάθηκε νωρίτερα, η οποία βελτιώνει την ασφάλεια των γραμμικών ταξινομητών, χρησιμοποιώντας ένα σύστημα πολλαπλών ταξινομητών (MCS). Για το σκοπό αυτό, οι πολλαπλοί γραμμικοί ταξινομητές μαθαίνουν με ομοιόμορφη δειγματοληψία (μια τεχνική γνωστή ως bagging [14]) και με τυχαία δειγματοληψία του συνόλου χαρακτηριστικών. Θεωρείται ότι η έκδοση του Sec-SVM περιορίζει σε μεγάλο βαθμό τη μίμηση καλοήθων δεδομένων και έτσι αποδίδει το πιο ασφαλές σύστημα.

Πίνακας 23

Άρθρο 22: Μία μέθοδος συνδυασμού γραφημάτων ροής ελέγχου και αλγόριθμων μηχανικής εκμάθησης, για την ανίχνευση κακόβουλου λογισμικού Android

No	Πηγή	Πρόβλημα	Λύση	Μεθοδολογία	Αποτελέσματα
22	(Ma, Ge, Liu, Zhao, & Ma, 2017)	Ανίχνευση κακόβουλου λογισμικού Android	Συνδυαστική μέθοδος	C4.5 DNN LSTM	Το μοντέλο ανίχνευσης επιτυγχάνει 98.98% ακρίβεια ανίχνευσης και έχει υψηλή ακρίβεια και σταθερότητα

Η ανίχνευση κακόβουλου λογισμικού Android, είναι μεγάλης σημασίας και αναγκαιότητας. Στην εργασία των Ma, Ge, Liu, Zhao, & Ma (2017), παρουσιάζεται μια συνδυαστική μέθοδος για τον εντοπισμό κακόβουλου λογισμικού σε Android, με βάση έναν αλγόριθμο μηχανικής μάθησης. Αρχικά, κατασκευάζεται το γράφημα ροής ελέγχου της εφαρμογής, για να ληφθούν οι πληροφορίες API. Με βάση τις πληροφορίες αυτές, κατασκευάζονται καινοτόμα Boolean, συχνότητες και χρονοσειρές με σύνολα δεδομένων. Με βάση αυτά τα τρία σύνολα δεδομένων, εξάγονται τρία μοντέλα ανίχνευσης κακόβουλου λογισμικού Android. Τα πειράματα διεξήχθησαν σε 10,010 καλοήθεις εφαρμογές και 10683 κακόβουλες εφαρμογές. Τα αποτελέσματα δείχνουν ότι το μοντέλο ανίχνευσης επιτυγχάνει 98.98% ακρίβεια ανίχνευσης και έχει υψηλή ακρίβεια και σταθερότητα.

Πίνακας 24

Άρθρο 23: Αποτελεσματικός εντοπισμός κακόβουλου λογισμικού Android με βάση τη χρήση των API ροής δεδομένων και τη μηχανική εκμάθηση

No	Πηγή	Πρόβλημα	Λύση	Μεθοδολογία	Αποτελέσματα
23	(Wu, Wang, Li, & Zhang, 2016)	Κακόβουλο λογισμικό Android	Σύστημα ανίχνευσης που παρέχει υψηλής ακρίβειας ταξινόμηση και αποτελεσματική ανάλυση ευαίσθητων δεδομένων μετάδοσης	Προσέγγιση μηχανικής μάθησης, που αξιοποιεί τη χρήση της εφαρμογής ροής δεδομένων με διεπαφές προγραμμάτων (API) ως χαρακτηριστικά ταξινόμησης	Το σύστημα μπορεί να επιτύχει ποσοστό ακρίβειας έως και 97.66% στον εντοπισμό άγνωστου κακόβουλου λογισμικού Android

Η μελέτη των Wu, Wang, Li, & Zhang (2016) προτείνει ένα σύστημα ανίχνευσης κακόβουλου λογισμικού Android, που παρέχει υψηλής ακρίβειας ταξινόμηση και αποτελεσματική ανάλυση μετάδοσης ευαίσθητων δεδομένων. Η μελέτη υιοθετεί μία προσέγγιση μηχανικής μάθησης, που αξιοποιεί τη χρήση της εφαρμογής ροής δεδομένων με διεπαφές προγραμμάτων (API) ως χαρακτηριστικά ταξινόμησης, για τον εντοπισμό κακόβουλου λογισμικού Android. Πραγματοποιείται μια διεξοδική ανάλυση για την εξαγωγή χαρακτηριστικών σε επίπεδο API, που σχετίζονται με τη ροή δεδομένων και τη βελτίωση της ταξινόμησης του μοντέλου k-nearest neighbour classification. Η λίστα API, που σχετίζεται με τη ροή δεδομένων, βελτιστοποιείται περαιτέρω μέσω της μηχανικής εκμάθησης, κάτι που επιτρέπει να βελτιωθεί σημαντικά η αποτελεσματικότητα της ανάλυσης μετάδοσης ευαίσθητων δεδομένων και η αναλυτική ακρίβεια.

Το προτεινόμενο σχήμα, αξιολογείται χρησιμοποιώντας 1,160 καλοήθη και 1,050 κακόβουλα δείγματα. Τα αποτελέσματα δείχνουν ότι το σύστημα μπορεί να επιτύχει ποσοστό ακρίβειας έως και 97.66% στον εντοπισμό άγνωστου κακόβουλου λογισμικού Android. Το πείραμα για την ανάλυση στατικής ροής δεδομένων, δείχνει ότι περισσότερο από το 85% των ευαίσθητων διαδρομών μετάδοσης δεδομένων, μπορεί να προσδιοριστεί χρησιμοποιώντας το βελτιωμένο υποσύνολο API, ενώ ο

χρόνος ανάλυσης μειώνεται κατά σχεδόν 40%. Η χρήση API λοιπόν, είναι μια έγκυρη δυνατότητα για τον εντοπισμό κακόβουλου λογισμικού Android.

Πίνακας 25

Άρθρο 24: EMULATOR vs REAL PHONE: Ανίχνευση κακόβουλου λογισμικού Android με χρήση μηχανικής εκμάθησης

No	Πηγή	Πρόβλημα	Λύση	Μεθοδολογία	Αποτελέσματα
24	(Alzaylaee, Yerima, & Sezer, 2017)	Ταχεία αύξηση του κακόβουλου λογισμικού	Αντίμετρα κατά της εξομοίωσης, για τον εντοπισμό κακόβουλου λογισμικού Android	ML Δυναμική ανάλυση	Η δυναμική ανάλυση αποδείχθηκε πιο αποτελεσματική βάσει συσκευής Αναλύθηκαν με επιτυχία περισσότερες από 24% εφαρμογές τηλεφώνου

Το λειτουργικό σύστημα Android έχει γίνει το πιο δημοφιλές λειτουργικό σύστημα για smartphone και tablet, οδηγώντας σε ταχεία αύξηση του κακόβουλου λογισμικού. Το εξελιγμένο κακόβουλο λογισμικό Android, χρησιμοποιεί τεχνικές αποφυγής εντοπισμού, για να κρύψει τις κακόβουλες δραστηριότητές του από τα εργαλεία ανάλυσης. Αυτές περιλαμβάνουν ένα ευρύ φάσμα τεχνικών κατά του εξομοιωτή, όπου τα προγράμματα κακόβουλου λογισμικού προσπαθούν να κρύψουν τις κακόβουλες δραστηριότητές τους, ανιχνεύοντας τον εξομοιωτή. Για αυτόν τον λόγο, τα αντίμετρα κατά της εξομοίωσης γίνονται όλο και πιο σημαντικά στον εντοπισμό κακόβουλου λογισμικού Android. Η ανάλυση και η ανίχνευση που βασίζονται σε πραγματικές συσκευές, μπορούν να ανακουφίσουν τα προβλήματα της αντι-εξομοίωσης, καθώς και να βελτιώσουν την αποτελεσματικότητα της δυναμικής ανάλυσης.

Ως εκ τούτου σε αυτό το άρθρο, παρουσιάζεται μια έρευνα ανίχνευσης κακόβουλου λογισμικού με βάση τη μηχανική μάθηση, χρησιμοποιώντας δυναμική ανάλυση σε πραγματικές συσκευές. Υλοποιείται ένα εργαλείο για την αυτόματη εξαγωγή δυναμικών χαρακτηριστικών από τηλέφωνα Android και μέσω πολλών πειραμάτων, πραγματοποιείται μια συγκριτική ανάλυση ανίχνευσης βάσει εξομοιωτή, έναντι ανίχνευσης βάσει συσκευής, μέσω αρκετών αλγορίθμων μηχανικής μάθησης. Η μελέτη δείχνει ότι πολλά χαρακτηριστικά θα μπορούσαν να εξαχθούν πιο αποτελεσματικά από τη δυναμική ανάλυση στη συσκευή, σε σύγκριση με τους εξομοιωτές. Διαπιστώθηκε επίσης, ότι περίπου 24% περισσότερες εφαρμογές αναλύθηκαν με επιτυχία στο τηλέφωνο. Επιπλέον, όλη η ανίχνευση, βάσει μηχανικής μάθησης που μελετήθηκε, απέδωσε καλύτερα όταν εφαρμόστηκε σε χαρακτηριστικά που εξάγονται από τη δυναμική ανάλυση στη συσκευή.

Πίνακας 26

Άρθρο 25: Ανίχνευση κακόβουλου λογισμικού Android με χρήση βελτιστοποιημένης επιλογής λειτουργιών και χρήση μηχανικής μάθησης βάσει γενετικού αλγόριθμου

No	Πηγή	Πρόβλημα	Λύση	Μεθοδολογία	Αποτελέσματα
25	(Fatima, Maurya, Dutta, Burget, & Masek, 2019)	Κακόβουλες εφαρμογές	Αποτελεσματική προσέγγιση βασισμένη σε ML	Γενετικός Αλγόριθμος	Η ακρίβεια ταξινόμησης άνω του 94%

Η πλατφόρμα Android, λόγω των χαρακτηριστικών ανοιχτού κώδικα και της υποστήριξης της Google, κατέχει το μεγαλύτερο μερίδιο αγοράς παγκοσμίως. Όντας το πιο δημοφιλές λειτουργικό σύστημα στον κόσμο, έχει τραβήξει την προσοχή των εγκληματιών του κυβερνοχώρου που λειτουργούν ιδιαίτερα μέσω της ευρείας διανομής κακόβουλων εφαρμογών. Η εργασία των Fatima, Maurya, Dutta, Burget, & Masek (2019), προτείνει μια αποτελεσματική προσέγγιση, βασισμένη στη μηχανική μάθηση για την ανίχνευση κακόβουλου λογισμικού Android, χρησιμοποιώντας τον εξελικτικό

Γενετικό Αλγόριθμο για διακριτική επιλογή χαρακτηριστικών. Επιλεγμένα χαρακτηριστικά από τον Γενετικό Αλγόριθμο, χρησιμοποιούνται για την εκπαίδευση των ταξινομητών μηχανικής μάθησης και συγκρίνεται η ικανότητά τους στον εντοπισμό κακόβουλου λογισμικού (πριν και μετά την επιλογή χαρακτηριστικών).

Τα αποτελέσματα του πειραματισμού, επικυρώνουν ότι ο Γενετικός Αλγόριθμος παρέχει το πιο βελτιστοποιημένο υποσύνολο χαρακτηριστικών, βοηθώντας στη μείωση της διάστασης χαρακτηριστικών σε λιγότερο από το μισό του αρχικού συνόλου τους. Μετά την επιλογή χαρακτηριστικών για τους ταξινομητές, που βασίζονται στη μηχανική μάθηση, η ακρίβεια ταξινόμησης διατηρείται άνω του 94% (ενώ εργάζεται στη μειωμένη διάσταση χαρακτηριστικών), με αποτέλεσμα να έχει θετικό αντίκτυπο στην υπολογιστική πολυπλοκότητα των ταξινομητών αυτών.

Πίνακας 27

Άρθρο 26: Αξιολόγηση μοντέλων μηχανικής εκμάθησης για ανίχνευση κακόβουλου λογισμικού Android: Μια μελέτη σύγκρισης

No	Πηγή	Πρόβλημα	Λύση	Μεθοδολογία	Αποτελέσματα
26	(Rana, Gudla, & Sung, 2018)	Κακόβουλο λογισμικό Android	Εντοπισμός κακόβουλου λογισμικού Android	Αλγόριθμοι ML	Ακρίβεια: SVM 94% Random Forest: 97.24%

Το Android είναι το πιο δημοφιλές λειτουργικό σύστημα για κινητά, με δισεκατομμύρια ενεργούς χρήστες παγκοσμίως, που προσέλκυσε διαφημιστές, χάκερ και εγκληματίες στον κυβερνοχώρο, για να αναπτύξουν κακόβουλο λογισμικό για διάφορους σκοπούς. Τα τελευταία χρόνια, έχουν διεξαχθεί εκτεταμένες έρευνες σχετικά με την ανάλυση και τον εντοπισμό κακόβουλου λογισμικού για συσκευές Android, ενώ το ίδιο επίσης, έχει εφαρμόσει διάφορους ελέγχους ασφαλείας για την αντιμετώπιση των προβλημάτων κακόβουλου λογισμικού, όπως το μοναδικό

αναγνωριστικό χρήστη (UID) για κάθε εφαρμογή, τα δικαιώματα του συστήματος και την πλατφόρμα διανομής Google Play.

Στο άρθρο των Rana, Gudla, & Sung (2018), εφαρμόζεται ένας ταξινομητής που βασίζεται σε στατική ανάλυση, που αξιολογεί και βελτιστοποιεί, με διαφορετικούς τύπους αλγορίθμων μηχανικής μάθησης, τον εντοπισμό κακόβουλου λογισμικού σε εφαρμογές που εκτελούνται στο λειτουργικό σύστημα Android. Στην αξιολόγηση, χρησιμοποιούνται 11,120 εφαρμογές, με 5,560 δείγματα κακόβουλου λογισμικού και 5,560 καλοήγη δείγματα του συνόλου δεδομένων DREBIN και η ακρίβεια που επιτυγχάνεται είναι μεγαλύτερη από 94%. Διαπιστώθηκε ότι ο ταξινομητής Random Forest ξεπερνά το καλύτερο αποτέλεσμα, που αναφέρθηκε προηγουμένως (περίπου 94% ακρίβεια, που λήφθηκε από την SVM), με ακρίβεια 97.24% και έτσι παρέχει μια ισχυρή βάση για τη δημιουργία αποτελεσματικών εργαλείων για τον εντοπισμό κακόβουλου λογισμικού Android. Επομένως, η μελέτη απέδειξε την αποτελεσματικότητα της χρήσης ταξινομητών της μηχανικής μάθησης.

Πίνακας 28

Άρθρο 27: Εφαρμογή αλγορίθμων μηχανικής μάθησης για ανίχνευση κακόβουλου λογισμικού Android

No	Πηγή	Πρόβλημα	Λύση	Μεθοδολογία	Αποτελέσματα
27	(Kakavand, Dabbagh, & Dehghantanha, 2018)	Κακόβουλες εφαρμογές	Ανίχνευση κακόβουλου λογισμικού	SVM KNN	Μέσο ποσοστό ακρίβειας: 79,08% και 80,50%. Μέσω πραγματικό θετικό ποσοστό: άνω του 67,00% και 80,00%

Στην εργασία των Kakavand, Dabbagh, & Dehghantanha (2018), δύο αλγόριθμοι Μηχανικής Μάθησης (ML), που ονομάζονται Υποστήριξη Διανυσματικής Μηχανής (SVM) και K- Nearest Neighbors (KNN), εφαρμόζονται και αξιολογούνται για την ταξινόμηση του συνόλου των χαρακτηριστικών είτε σε καλοήθεις, είτε σε κακόβουλες εφαρμογές, μέσω της εποπτευόμενης μαθησιακής διαδικασίας. Αυτή η εργασία περιλαμβάνει στατική ανάλυση εφαρμογών, η οποία ελέγχει την παρουσία και τη συχνότητα των λέξεων - κλειδιών στο αρχείο μανιφέστου των εφαρμογών Android και εξάγει τα σετ στατικών χαρακτηριστικών από ένα σύνολο δεδομένων 400 εφαρμογών, για να παράγει καλύτερα αποτελέσματα ανίχνευσης κακόβουλου λογισμικού. Η απόδοση ταξινόμησης των αλγορίθμων ML, μετρείται ως προς την ακρίβεια και τον πραγματικό θετικό ρυθμό και ερμηνεύεται για να προσδιοριστεί ποιος αλγόριθμος είναι πιο εφαρμόσιμος για τον εντοπισμό κακόβουλου λογισμικού Android. Τα πειραματικά αποτελέσματα για ένα σύνολο δεδομένων πραγματικού κακόβουλου λογισμικού και καλοήθων εφαρμογών, υποδεικνύουν το μέσο ποσοστό ακρίβειας 79,08% και 80,50%, με μέσο πραγματικό θετικό ποσοστό άνω του 67,00% και 80,00%, χρησιμοποιώντας SVM και KNN αντίστοιχα.

Πίνακας 29

Άρθρο 28: Μία έρευνα για την τεχνολογία στατικής ανίχνευσης κακόβουλου λογισμικού Android με βάση τη μηχανική μάθηση

No	Πηγή	Πρόβλημα	Λύση	Μεθοδολογία	Αποτελέσματα
28	(Wu, Zhu, & Liu, 2021)	Κακόβουλες εφαρμογές κλέβουν πληροφορίες απορρήτου των χρηστών, στέλνουν μηνύματα κειμένου για ενεργοποίηση αφαιρέσεων, εκμεταλλεύονται την κλιμάκωση των προνομίων για τον έλεγχο του συστήματος κ.α.	Μέθοδοι στατικής ανίχνευσης βάση μηχανικής μάθησης	Σύγκριση μεθόδων	Έχει τα εξής πλεονεκτήματα: Πληρότητα Ακρίβεια Μικρότερη εξάρτηση από τους ειδικούς στον εντοπισμό Έχει όμως και αδυναμίες

Με την ταχεία ανάπτυξη των συσκευών και εφαρμογών Android, το περιβάλλον Android αντιμετωπίζει πολλές απειλές ως προς την ασφάλεια. Κακόβουλες εφαρμογές που κλέβουν πληροφορίες απορρήτου των χρηστών, στέλνουν μηνύματα κειμένου για να ενεργοποιήσουν αφαιρέσεις, εκμεταλλεύονται την κλιμάκωση των προνομίων για τον έλεγχο του συστήματος κ.λπ. και προκαλούν σημαντική βλάβη στους τελικούς χρήστες. Για τον εντοπισμό του κακόβουλου λογισμικού Android, οι ερευνητές έχουν προτείνει διάφορες τεχνικές, μεταξύ των οποίων οι μέθοδοι που βασίζονται στη μηχανική μάθηση με στατικά χαρακτηριστικά των εφαρμογών ως διανύσματα εισόδου, έχουν εμφανή πλεονεκτήματα στην κάλυψη κώδικα, τη λειτουργική αποτελεσματικότητα και τον μαζικό εντοπισμό δειγμάτων.

Στο άρθρο των Wu, Zhu, & Liu (2021), διερευνήθηκε η δομή των εφαρμογών Android, αναλύθηκαν διάφορες πηγές στατικών χαρακτηριστικών, εξετάστηκαν οι μέθοδοι μηχανικής εκμάθησης για τον εντοπισμό κακόβουλου λογισμικού Android και μελετήθηκαν τα πλεονεκτήματα. Σε σύγκριση με άλλους τύπους τεχνολογίας ανίχνευσης κακόβουλων εφαρμογών Android, η μέθοδος στατικής ανίχνευσης που βασίζεται στη μηχανική εκμάθηση έχει πλεονεκτήματα στην πληρότητα, την ακρίβεια και τη μικρότερη εξάρτηση από τους ειδικούς στον εντοπισμό. Βέβαια, είναι σημαντικό να αναφερθεί ότι έχει και ορισμένες αδυναμίες, που όμως δεν είναι περισσότερες από τα δυνατά του σημεία.

Πίνακας 30

Άρθρο 29: Ανίχνευση κακόβουλου λογισμικού Android με βάση χρήσιμες κλήσεις API και μηχανική εκμάθηση

No	Πηγή	Πρόβλημα	Λύση	Μεθοδολογία	Αποτελέσματα
29	(Jung, και συν., 2018)	Κακόβουλο λογισμικό	Ακριβής εντοπισμός κακόβουλου λογισμικού	benign_API_list malicious_API_list	Ο ταξινομητής Random Forest με top50_benign_API_list μπορεί να επιτύχει υψηλή ακρίβεια 99.98%

Ο ακριβής εντοπισμός κακόβουλου λογισμικού μπορεί να ωφελήσει σημαντικά τους χρήστες Android, λαμβάνοντας υπόψη τον αυξανόμενο αριθμό εξελιγμένων κακόβουλων προγραμμάτων. Στη μελέτη του Jung et al. (2018), προτείνεται μία μεθοδολογία ανίχνευσης κακόβουλου λογισμικού που βασίζεται στη μηχανική μάθηση, η οποία προσδιορίζει το υποσύνολο των API Android, που είναι αποτελεσματικό ως δυνατότητα και ταξινομεί τις εφαρμογές Android ως καλοήθειες ή κακόβουλες εφαρμογές. Η προτεινόμενη μεθοδολογία κατασκευάζει αρχικά δύο ταξινομημένες λίστες δημοφιλών API Android. Το ένα είναι το `benign_API_list`, που περιέχει τα κορυφαία δημοφιλή API που χρησιμοποιούνται συνήθως σε καλοήθειες εφαρμογές και το άλλο `malicious_API_list`, που περιέχει τα κορυφαία δημοφιλή API που χρησιμοποιούνται συνήθως σε κακόβουλες εφαρμογές.

Παρατηρείται ότι το σύνολο των API στο `benign_API_list` είναι αρκετά διαφορετικό από το σύνολο των API στο `malicious_API_list`. Εφαρμόζεται ο ταξινομητής Random Forest σε ένα σύνολο δεδομένων 60,243 εφαρμογών, χρησιμοποιώντας κάθε λίστα ως χαρακτηριστικά του ταξινομητή. Για να αξιολογηθεί η προτεινόμενη μεθοδολογία, δημιουργείται `top50_benign_API_list` και `top50_malicious_API_list`, επιλέγοντας μόνο τα πρώτα 50 API σε κάθε λίστα κατάταξης. Η αξιολόγηση δείχνει ότι ο ταξινομητής Random Forest με `top50_benign_API_list`, είναι πιο ακριβής από αυτόν με `top50_malicious_API_list`. Ο ταξινομητής Random Forest με `top50_benign_API_list` μπορεί να επιτύχει υψηλή ακρίβεια 99.98%.

Πίνακας 31

Άρθρο 30: Ανίχνευση κακόβουλου λογισμικού Android με βάση λειτουργίες που βασίζονται σε εικόνες και τεχνικές μηχανικής εκμάθησης

No	Πηγή	Πρόβλημα	Λύση	Μεθοδολογία	Αποτελέσματα
30	(Ünver & Bakour, 2020)	Δείγματα κακόβουλου λογισμικού	Εντοπισμός δειγμάτων κακόβουλου λογισμικού στο περιβάλλον	Μετατροπή ορισμένων περιεχομένων του αρχείου APK σε εικόνες, σε κλίμακα του γκρι Random Forest k- Nearest Neighbors Δέντρο αποφάσεων Bagging AdaBoost Gradient Boost	Το προτεινόμενο μοντέλο υπερέχει των υπολοίπων όσον αφορά την ακρίβεια ταξινόμησης και τον υπολογιστικό χρόνο, όπου η ακρίβεια ταξινόμησής του έφτασε πάνω από 98%. Με μια τυπική επιβάρυνση χρόνου εκτέλεσης δεν υπερβαίνει τα 0.018 s κατά μέσο όρο για κάθε δείγμα

Στην εργασία των Ünver & Bakour (2020), έχει προταθεί ένα μοντέλο ταξινόμησης κακόβουλου λογισμικού, για τον εντοπισμό τους στο περιβάλλον Android. Το προτεινόμενο μοντέλο βασίζεται στη μετατροπή ορισμένων αρχείων από την πηγή των εφαρμογών Android σε εικόνες, σε κλίμακα του γκρι. Ορισμένα τοπικά και καθολικά χαρακτηριστικά που βασίζονται σε εικόνα, συμπεριλαμβανομένων τεσσάρων διαφορετικών τύπων τοπικών και τριών διαφορετικών τύπων καθολικών χαρακτηριστικών, έχουν εξαχθεί από τα κατασκευασμένα σύνολα δεδομένων εικόνας σε κλίμακα του γκρι και χρησιμοποιήθηκαν για την εκπαίδευση του προτεινόμενου μοντέλου. Απ' όσο είναι γνωστό, αυτός ο τύπος λειτουργιών, χρησιμοποιείται για πρώτη φορά στον τομέα εντοπισμού κακόβουλου λογισμικού Android.

Επιπλέον, ο αλγόριθμος bag of visual words, έχει χρησιμοποιηθεί για την κατασκευή ενός διανύσματος χαρακτηριστικών, από τους περιγραφείς του τοπικού χαρακτηριστικού που εξάγεται από κάθε εικόνα. Τα εξαγόμενα τοπικά και καθολικά χαρακτηριστικά έχουν χρησιμοποιηθεί για την εκπαίδευση πολλαπλών ταξινομητών

μηχανικής μάθησης, όπως Random Forest, k- Nearest Neighbors, Δέντρα αποφάσεων, Bagging, AdaBoost και Gradient Boost. Η προτεινόμενη μέθοδος απέκτησε πολύ υψηλή ακρίβεια ταξινόμησης, που έφτασε το 98.75%, με τυπικό υπολογιστικό χρόνο να μην υπερβαίνει τα 0.018 s για κάθε δείγμα. Τα αποτελέσματα του προτεινόμενου μοντέλου ξεπέρασαν τα αποτελέσματα όλων των συγκριτικών μοντέλων τελευταίας τεχνολογίας, όσον αφορά τόσο την ακρίβεια ταξινόμησης, όσο και τον υπολογιστικό χρόνο.

7. Σύγκριση των τρόπων με τους οποίους λύνεται το κάθε πρόβλημα κακόβουλου λογισμικού (malware) και αποτελέσματα

Στους παραπάνω Πίνακες, είδαμε ότι μελετήθηκαν πολλοί και διάφοροι μέθοδοι και μοντέλα, ώστε να βρεθεί το μέγιστο δυνατό αποτέλεσμα, όσον αφορά το ποσοστό ακρίβειας ανίχνευσης κακόβουλου λογισμικού (malware) σε Android εφαρμογές. Γενικά παρατηρήθηκε ότι στις περισσότερες περιπτώσεις, ο συνδυασμός δύο ή και περισσότερων μεθόδων / μοντέλων / αλγορίθμων, είχε καλύτερα αποτελέσματα σε σχέση με τη χρήση μίας / ενός.

Από τους πίνακες και τα παρακάτω γραφήματα προκύπτει και αποδεικνύεται, ότι η καλύτερη μέθοδος για αντιμετώπιση ενός κακόβουλου λογισμικού σε Android, είναι ο συνδυασμός των OneR και J48, με ακρίβεια 100% (το μέγιστο δυνατό) και ψευδές θετικό ποσοστό 0%. Μάλιστα, χρησιμοποιήθηκε σε δύο περιπτώσεις (Πίνακας 3 και 14) και βγήκε και στις δύο το ίδιο μέγιστο αποτέλεσμα. Πολύ μεγάλα ποσοστά εντοπισμού κακόβουλου λογισμικού είχαμε όμως και από τις μεθόδους M – CNN (Πίνακας 5), με ποσοστό ακρίβειας 98.52% και με Random Forest με top50_benign_API_list (Πίνακας 29), με ποσοστό ακρίβειας να φτάνει μόλις το 98.98% (πολύ κοντά στο μέγιστο δυνατό).

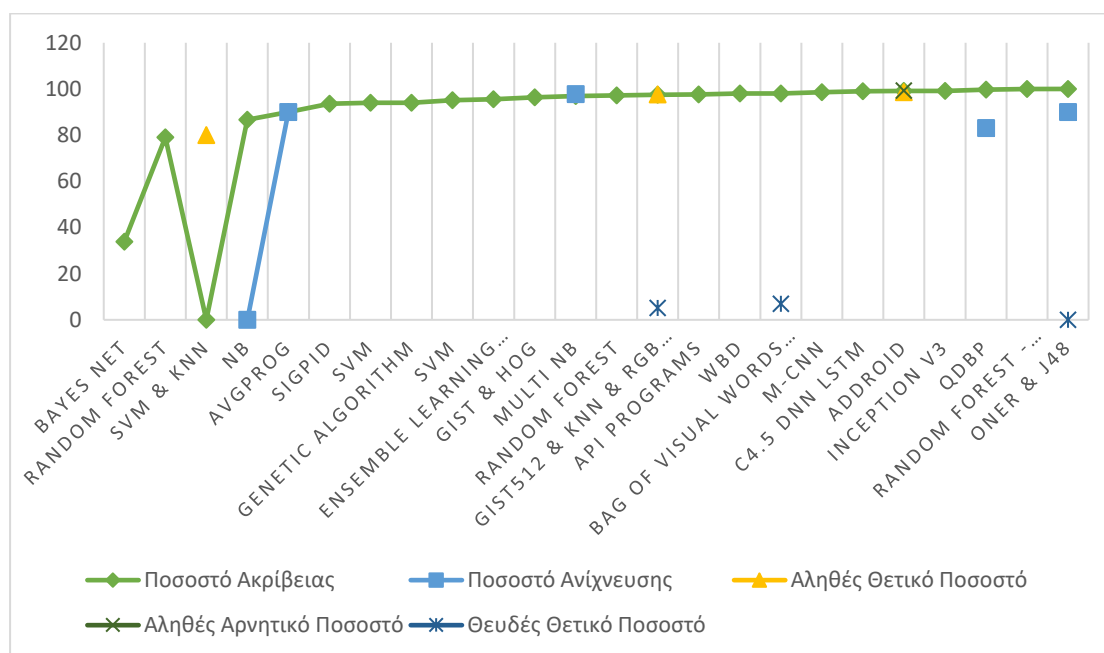
Αντιθέτως, η μέθοδος με τα χειρότερα αποτελέσματα στην εύρεση κακόβουλου λογισμικού, ήταν αυτή με τον συνδυασμό των Bayes net, Naïve Bayes, Random Forest, Logistic, J48 και Multilayer Perceptron (Πίνακας 17), με ακρίβεια να φτάνει μέχρι το 33.79%, ποσοστό πολύ μικρότερο από οποιουδήποτε άλλου μελετήθηκε.

Επίσης υπήρχαν περιπτώσεις χρήσης μεθόδων, όπου είχαν πολύ εύλογο υπολογιστικό χρόνο, κάτι που είναι μεγάλης σημασίας, όπως στις περιπτώσεις του Πίνακα 1 και 30. Χρησιμοποιήθηκαν η μέθοδος GIST & HOG ανάλυσης εικόνων και ο αλγόριθμος bag of visual words, αντίστοιχα.

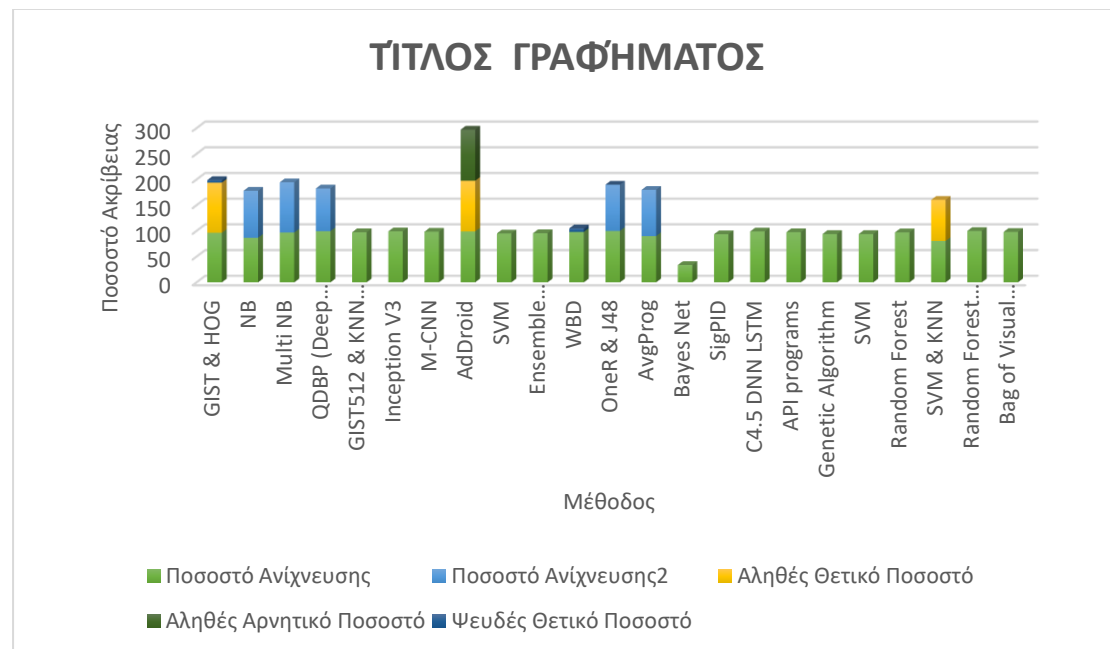
8. Γραφήματα

Παρακάτω παρουσιάζονται τα ποσοστά της ακρίβειας, της ανίχνευσης, των αληθών θετικών και αρνητικών, και των ψευδών θετικών, σε δύο γραφήματα διαφορετικών τύπων, ένα γραμμών και ένα στηλών, που δημιουργήθηκαν από δοκιμές και πειράματα διαφορετικών αλγορίθμων και μεθόδων που αναφέρθηκαν στους παραπάνω τριάντα πίνακες. Σε κάποιες περιπτώσεις έγιναν παραπάνω από μία δοκιμές, αλλά στα γραφήματα χρησιμοποιήθηκαν τα μεγαλύτερα ποσοστά με την μέγιστη ακρίβεια, ώστε να βρεθούν τα μεγαλύτερα και καλύτερα δυνατά αποτελέσματα. Επίσης, θα αναφερθούν και ποσοστά αποφυγής κακόβουλου λογισμικού που παρουσιάστηκαν σε κάποιες περιπτώσεις.

Γράφημα Γραμμών



Γράφημα Στηλών



Εκτός από της παραπάνω μεθόδους, πολύ καλά αποτελέσματα έδειξαν και οι εξής: η PCA – RELIEF (με υψηλό ποσοστό ανίχνευσης), η SERLA, η Sec – SVM, η Bagging (με υψηλό ποσοστό ακρίβειας), η SL η οποία βγήκε οριακά καλύτερη από τις υπόλοιπες που χρησιμοποιήθηκαν στο ίδιο πείραμα και η ML, η οποία χρησιμοποιεί δυναμική ανάλυση, που χρησιμοποιείται κυρίως στα τηλέφωνα και έχει 24% μεγαλύτερη επιτυχία από τις υπόλοιπες. Τέλος, η ML έχει την ικανότητα να ανιχνεύει και να αποφεύγει κακόβουλο λογισμικό με δύο διαφορετικές προσεγγίσεις: (α) με DroidDetector, με ποσοστό αποφυγής 50% και (β) με DeepClassifyDroid με ίδιο ποσοστό, δηλαδή 50%.

9. Συμπεράσματα

Με τη διάδοση του Διαδικτύου, του 5G και άλλων τεχνολογιών, οι κινητές έξυπνες συσκευές αναπτύσσονται γρήγορα και η κλίμακα των εφαρμογών Android που είναι εγκατεστημένες σε έξυπνα τερματικά, όπως κινητά τηλέφωνα και tablet, αυξάνεται επίσης. Ωστόσο, αυτό έχει ακολουθηθεί από αύξηση του κακόβουλου λογισμικού που στοχεύει σε αυτές τις πλατφόρμες. Με τη σειρά του, αυτό έχει προσελκύσει μεγάλη έρευνα για τον εντοπισμό εφαρμογών Android που επηρεάζονται από κακόβουλο λογισμικό. Η εισαγωγή μεθόδων τεχνητής νοημοσύνης, όπως η μηχανική μάθηση, έχει βελτιώσει σημαντικά τις προοπτικές για τον εντοπισμό κακόβουλου λογισμικού Android. Μέσα από την έρευνα της συλλεγόμενης βιβλιογραφίας, αυτή η εργασία παρέχει μια λεπτομερή ανασκόπηση των τρεχουσών προσεγγίσεων για τον εντοπισμό κακόβουλου λογισμικού Android, με έμφαση στη χρήση της μηχανικής μάθησης. Ο κύριος στόχος αυτής της εργασίας είναι να παρουσιάσει μια πλήρη εικόνα του εντοπισμού κακόβουλου λογισμικού Android με βάση τη μηχανική εκμάθηση μέσω της βιβλιογραφικής έρευνας από επίσημα επιστημονικά περιοδικά.

Παρουσιάστηκε εν συντομία το υπόβαθρο του κακόβουλου λογισμικού Android και δόθηκε μια ολοκληρωμένη ανασκόπηση των προσεγγίσεων που βασίζονται στη μηχανική μάθηση για τον εντοπισμό κακόβουλου λογισμικού Android, που είναι ταξινομημένες κατά προσέγγιση με τη σειρά του αγωγού ανάπτυξης μηχανικής μάθησης. Το βασικό περιεχόμενο συνοψίστηκε με τη μορφή πινάκων για καλύτερη κατανόηση και διευκόλυνση της συγκριτικής ανάλυσης.

Η ταξινόμηση κακόβουλου λογισμικού έχει γίνει μείζον θέμα για έρευνα και ανησυχία λόγω της συνεχιζόμενης ανάπτυξης του κακόβουλου κώδικα τα τελευταία χρόνια. Με στόχο το πρόβλημα και οι δυνατότητες που χρησιμοποιούνται στις τρέχουσες μεθόδους ταξινόμησης κακόβουλου κώδικα εξαρτώνται πάρα πολύ τόσο από την εμπειρία των ειδικών όσο και από την υψηλή πολυπλοκότητα που προκαλείται από την υψηλή διάσταση των χαρακτηριστικών. Η μηχανική μάθηση και πιο συγκεκριμένα η μέθοδος συνδυασμού των ταξινομητών OneR και J48, παρουσίασε στις περισσότερες περιπτώσεις την υψηλότερη ακρίβεια (100% ποσοστό ακρίβειας).

Η αντιμετώπιση του εντοπισμού κακόβουλου λογισμικού Android με χρήση τεχνικών μηχανικής εκμάθησης απαιτεί περαιτέρω έρευνα. Θα ήταν επιθυμητό να προταθούν νέα πλαίσια και αλγόριθμοι που είναι αρκετά γρήγοροι και ισχυροί για τον εντοπισμό του κακόβουλου λογισμικού. Με βάση τα αναθεωρημένα έγγραφα, ο παραλληλισμός ή η ενσωμάτωση μιας ή περισσότερων καλών τεχνικών μαζί, μπορεί να οδηγήσει σε ακριβή και αποτελεσματικό μηχανισμό.

Η εγκληματολογία μνήμης θα αποκτήσει μεγαλύτερη δημοτικότητα στο εγγύς μέλλον, λόγω της αύξησης του κακόβουλου λογισμικού χωρίς αρχεία και παρόμοιων απειλών. Ως μελλοντική εργασία, για να αυξηθεί η ακρίβεια και να αποκτηθεί μία πλατφόρμα ανάλυσης από άκρο σε άκρο, θα ήταν καλό να σχεδιαστεί σωστή διερεύνηση των ικανοτήτων των συνελκτικών νευρωνικών δικτύων, που είναι εξοπλισμένα με μηχανισμούς αυτοπροσοχής, για να αποκαλύψει πιο μεροληπτικές αναπαραστάσεις, καθώς και να επιτευχθούν καλύτερες δυνατότητες γενίκευσης. Μια άλλη ενδιαφέρουσα ιδέα που προβάλλεται, είναι το χαρακτηριστικό αντίστροφου μετασχηματισμού του UMAP, που μπορεί να επιτρέψει τη δημιουργία συνθετικών χαρακτηριστικών εικόνων, ανάλογα με την αποκωδικοποίηση των αυτόματων κωδικοποιητών.

Σε αυτό το άρθρο, προτάθηκε η χρήση αδειών και κλήσεων API εφαρμογών Android, για τον εντοπισμό κακόβουλου λογισμικού και κακόβουλων κωδικών, σε πλατφόρμα για κινητά που βασίζεται στο Android. Το προτεινόμενο πλαίσιο εξάγει δικαιώματα από εφαρμογές Android, άλλα και συνδυάζει τις κλήσεις API για να χαρακτηρίσει κάθε εφαρμογή ως ένα διάνυσμα χαρακτηριστικό υψηλής διάστασης. Με την εφαρμογή μεθόδων μάθησης στα συλλεγμένα σύνολα δεδομένων, μπορούμε να αντλήσουμε μοντέλα ταξινόμησης, για να χαρακτηριστούν οι εφαρμογές ως καλοήθεις ή κακοήθεις (κακόβουλο λογισμικό). Πειράματα σε δεδομένα πραγματικού κόσμου καταδεικνύουν την καλή απόδοση του πλαισίου για τον εντοπισμό κακόβουλου λογισμικού. Το πλαίσιο αυτό, σε σύγκριση με τις υπάρχουσες λύσεις, είναι τριπλό: (1) επικυρώνει ότι ο συνδυασμός δικαιωμάτων και κλήσεων API είναι αποτελεσματικός για τον εντοπισμό κακόβουλου λογισμικού, (2) περιλαμβάνει οποιαδήποτε δυναμική ανίχνευση των εφαρμογών Android και (3) γενικεύεται και ισχύει για όλα τα κινητά, για εντοπισμό κακόβουλου λογισμικού (Peiravian & Zhu, 2013).

10. Λέξεις Κλειδιά

ML: Machine Learning (Μηχανική Μάθηση)

Malware: Κακόβουλο Λογισμικό

KNN: K – Nearest Neighbor (Μέθοδος)

J48: Decision Tree (Μέθοδος)

TP / FP: True / False Positive

TPR / FPR: True Positive Rate / False Positive Rate

SVM: Υποστήριξη Διανυσματικής Μάθησης (Μέθοδος)

RF: Random Forest (Μέθοδος)

SL: Simple Logistic (Μέθοδος)

NB: Naïve Bayes (Μέθοδος)

11. Βιβλιογραφία

- Kouliaridis, V., & Kambourakis, G. (2021). A Comprehensive Survey on Machine Learning Techniques for Android Malware Detection. *Information*, 12. doi:<https://doi.org/10.3390/info12050185>
- Alqahtani, E., Zagrouba , R., & Almuhaideb, A. (2018). Android Malware Detection Using Machine Learning. *Sixth International Conference on Software Defined Systems (SDS)*.
- Alqahtani, E., Zagrouba, R., & Almuhaideb, A. (2019). A Survey on Android Malware Detection Techniques Using Machine Learning Algorithms. *Sixth International Conference on Software Defined Systems (SDS)*, σσ. 110-117.
- Alzaylaee, M., Yerima, S., & Sezer, S. (2017, March). EMULATOR vs REAL PHONE: Android Malware Detection Using Machine Learning. *IWSPA '17: Proceedings of the 3rd ACM on International Workshop on Security And Privacy Analytics*, σσ. 65–72. doi:<https://doi.org/10.1145/3041008.3041010>
- Amos, B., Turner, H., & White, J. (2013). Applying machine learning classifiers to dynamic Android malware detection at scale. *IEEE*, σσ. 1666-1671.
- Anderson, H. S., Kharkar, A., Filar, B., & Roth, P. (2017). Evading Machine Learning Malware Detection. *Black Hat*.
- Babak, B. R., Maslin, M., & Suhaimi, I. (2011). Evolution of Computer Virus Concealment and Anti-Virus Techniques: A Short Survey.
- Balaganesh, D., Chakrabarti, A., & Midhunchakkaravarthy, D. (2018). Smart Devices Threats. *Vulnerabilities and Malware Detection Approaches: A Survey*, 3(2).
- Bensaoud, A., Abudawaood, N., & Kalita, J. (2020). Classifying Malware Images with Convolutional Neural Network Models. *Computer Science* .
- Bozkir, A., Tahillioglu, E., Aydos, M., & Kara, I. (2021). Catch them alive: A malware detection approach through memory forensics, manifold learning and computer

vision. *computers & security*, 103.
doi:<https://doi.org/10.1016/j.cose.2020.102166>

Chen, J., Jia, X., Zhao, C., & Zhang, W. (2020). Using the Rgb Image of Machine Code to Classify the Malware. *IEEE 5th International Conference on Cloud Computing and Big Data Analytics*.

Darwaish, A., Naït-Abdesselam, F., Titouna, C., & Sattar, S. (2021). Robustness of Image-based Android Malware Detection Under Adversarial Attacks. *IEEE International Conference on Communications*. doi: 10.1109/ICC42927.2021.9500425

Das, K., & Behera, R. (2017). A survey on machine learning: Concept algorithms and applications. *Int. J. Innov. Res. Comput. Commun. Eng.*, 5(2), σσ. 1301-1309.

Demontis, A., Melis, M., Biggio, B., Maiorca, D., Arp, D., Rieck, K., . . . Roli, F. (2019, JULY/AUGUST). Yes, Machine Learning Can Be More Secure! A Case Study on Android Malware Detection. *IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING*, 16(4), σσ. 711-724.

developers. (2021). doi:<https://developer.android.com/guide/platform>

Egham. (2018, February 22). Gartner Says Worldwide Sales of Smartphones Recorded First Ever Decline During the Fourth Quarter of 2017. *Gartner*.

Elenkov, N. (2015). *Android Security Internals: An In-depth Guide to Android's Security Architecture*. San Francisco, CA, USA.

Fatima, A., Maurya, R., Dutta, M. K., Burget, R., & Masek, J. (2019). Android Malware Detection Using Genetic Algorithm based Optimized Feature Selection and Machine Learning. *42nd International Conference on Telecommunications and Signal Processing (TSP)*. doi:10.1109/TSP.2019.8769039

Ham, H.-S., & Choi, M.-J. (2013). Analysis of Android Malware Detection Performance using Machine Learning Classifiers. *ICTC*, σσ. 490-495.

- Jian, Y., Kuang, H., Ren, C., Ma, Z., & Wang, H. (2021). A novel framework for image-based malware detection with a deep neural network. *computers & security* , 109. doi:<https://doi.org/10.1016/j.cose.2021.102400>
- Jung, J., Kim, H., Shin, D., Lee, M., Lee, H., Cho, S.-j., & Suh, K. (2018, September). Android Malware Detection Based on Useful API Calls and Machine Learning. *IEEE First International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*. doi:10.1109/AIKE.2018.00041
- Kakavand, M., Dabbagh, M., & Dehghantanha, A. (2018, November). Application of Machine Learning Algorithms for Android Malware Detection. *CIIS 2018: Proceedings of the 2018 International Conference on Computational Intelligence and Intelligent Systems*, 32–36. doi:<https://doi.org/10.1145/3293475.3293489>
- Li, J., Sun, L., Yan, Q., Li, Z., Srisa-an, W., & Ye, H. (2018, JULY 7). Significant Permission Identification for Machine-Learning-Based Android Malware Detection. *IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS*, 14, 3216-3225.
- LIU, K., XU, S., XU, G., ZHANG, M., SUN, D., & LIU, H. (2020). A Review of Android Malware Detection Approaches Based on Machine Learning. *Review of Android Malware Detection Approaches Based on Machine Learning*, 8, 124579-124607.
- Ma, Z., Ge, H., Liu, Y., Zhao, M., & Ma, J. (2017). A Combination Method for Android Malware Detection Based on Control Flow Graphs and Machine Learning Algorithms. *Digital Object Identifier*.
- Mahindru, A., & Singh, P. (2017). Dynamic Permissions based Android Malware Detection using Machine Learning Techniques. *ISEC '17: Proceedings of the 10th Innovations in Software Engineering Conference*, 202–210. doi:<https://doi.org/10.1145/3021460.3021485>
- Marsland, S. (2015). *Machine Learning: An Algorithmic Perspective*. Boca Raton, FL, USA: CRC press.

- Martens, B. (2022). What Is a Backdoor & How to Prevent Backdoor Attacks. *Safety detectives*.
- Mehtab, A., Bin Shahid, W., Yaqoob, T., Amjad, M., Abbas, H., Afzal, H., & Saqib, M. (2020). AdDroid: Rule-Based Machine Learning Framework for Android Malware Analysis. *Mobile Networks and Applications*, 25, σσ. 180-192.
- Milosevic, N., Dehghantanha, A., & Choo, K.-K. (2017). Machine learning aided Android malware classification. *Computers and Electrical Engineering*, 61, σσ. 266–274.
- Peiravian, N., & Zhu, X. (2013). Machine Learning for Android Malware Detection Using Permission and API Calls. *IEEE 25th International Conference on Tools with Artificial Intelligence*, σσ. 300-305. doi:DOI 10.1109/ICTAI.2013.53
- Posey, B. (2021, January). Backdoor (computing). *Techtarget*.
- Rana, M., Gudla, C., & Sung, A. (2018, December). Evaluating Machine Learning Models for Android Malware Detection: A Comparison Study. *ICNCC 2018: Proceedings of the 2018 VII International Conference on Network, Communication and Computing*, σσ. 17–21. doi:https://doi.org/10.1145/3301326.3301390
- Rosencrance, L. (2021, September). Remote access. *Techtarget*.
- Sahs , J., & Khan, L. (2012). A Machine Learning Approach to Android Malware Detection. *European Intelligence and Security Informatics Conference*, σσ. 141-147. doi:DOI 10.1109/EISIC.2012.34
- Ünver, H., & Bakour, K. (2020). Android malware detection based on image-based features and machine learning techniques. *SN Applied Sciences*, 2.
- Wang, X., & Li, C. (2021). Android malware detection through machine learning on kernel task Android malware detection through machine learning on kernel task. *Neurocomputing*, 435, σσ. 126-150.

- Wen, L., & Yu, H. (2017). An Android Malware Detection System Based on Machine Learning. *AIP Conference Proceedings*, 1864. doi:<https://doi.org/10.1063/1.4992953>
- Wu, Q., Zhu, X., & Liu, B. (2021). A Survey of Android Malware Static Detection Technology Based on Machine Learning. *Mobile Information Systems*, 2021. doi:<https://doi.org/10.1155/2021/8896013>
- Wu, S., Wang, P., Li, X., & Zhang, Y. (2016). Effective detection of android malware based on the usage of data flow APIs and machine learning. *Information and Software Technology*, 75, σσ. 17–25.
- Yerima, S., Sezer, S., & Muttik , I. (2014). Android Malware Detection Using Parallel Machine Learning Classifiers. *Eighth International Conference on Next Generation Mobile Applications, Services and Technology*, σσ. 37-42.