Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακή Διατριβή
Στην Ασφάλεια Υπολογιστών και Δικτύων

Η Επιρροή Του Είδους Μίας Εφαρμογής Στις Συσκευές Android Σε Επίπεδο Ασφάλειας

Αγγέλα Ιωάννου

Επιβλέπων Καθηγητής
Δρ. Σταύρος Σιαηλής

Μάιος 2017

Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Η Επιρροή Του Είδους Μίας Εφαρμογής Στις Συσκευές Android Σε Επίπεδο Ασφάλειας

Αγγέλα Ιωάννου

Επιβλέπων Καθηγητής
Δρ. Σταύρος Σιαηλής

Η παρούσα μεταπτυχιακή διατριβή υποβλήθηκε
προς μερική εκπλήρωση των απαιτήσεων για απόκτηση

μεταπτυχιακού τίτλου σπουδών
στην Ασφάλεια Υπολογιστών και Δικτύων

από τη Σχολή Θετικών και Εφαρμοσμένων Επιστημών
του Ανοικτού Πανεπιστημίου Κύπρου

Μάιος 2017

Περίληψη

Το Android έχει γίνει το πιο δημοφιλές λειτουργικό σύστημα smartphone. Η Google λέει ότι 1.3 εκατομμύρια Android συσκευές θα ενεργοποιούνται κάθε μέρα [09]. Ταυτόχρονα αφήνει πίσω του, τους ανταγωνιστές του κατέχοντας περισσότερο από το 78% της συνολικής αγοράς το 2013 [49]. Σύμφωνα με τη IDC-International Data Corporation, το Android OS κατέχει το 82.8% της συνολικής αγοράς το 2ο τρίμηνο του 2015 [49]. Η πλατφόρμα smartphone της Google Android είναι πασιφανές πως έχει ξεπεράσει τα Symbian και iOS. Την ίδια στιγμή με τη δημοτικότητα του, έχει γίνει κύριος στόχος των επιτιθέμενων. Αυτή η γρήγορη αύξηση της υιοθέτησης των Android έχει σαν αποτέλεσμα την σημαντική αύξηση του αριθμού των κακόβουλων λογισμικών σε σύγκριση με τα προηγούμενα χρόνια. Υπάρχουν ήδη πολλά antimalware προγράμματα που έχουν σχεδιαστεί για την αποτελεσματική προστασία των ευαίσθητων δεδομένων των χρηστών στα κινητά συστήματα από επιθέσεις.

Η παρούσα μελέτη έχει ως απώτερο σκοπό και στόχο τη μελέτη των δικαιωμάτων που συνήθως χρησιμοποιούν διάφορες εφαρμογές Android είτε αυτές προέρχονται από το Google Play Store είτε από Third Party Markets. Επιπλέον μελετάται η χρήση της μνήμης RAM και της CPU που γίνεται από τις διάφορες εφαρμογές ανάλογα με την προέλευσης τους ώστε να διαπιστώσουμε την επιρροή που δέχεται μία συσκευή, την ίδια στιγμή μελετάται και η επιρροή που δέχονται οι Android εκδόσεις: KITKAT, Lollipop και Marshmallow ώστε να αντιληφθούμε την εξέλιξη της ποιότητας και της ασφάλειας που σημειώνουν οι εκδόσεις Android σύμφωνα με τις ανάγκες που προκύπτουν. Τέλος μελετάται το Network Traffic που δημιουργείται κατά την χρήση των εν λόγω εφαρμογών, μέσα από αυτή την ανάλυση του αριθμού των TCP και HTTP ports που χρησιμοποιούν οι εφαρμογές. Καταδεικνύεται λοιπόν ότι η προέλευση μίας εφαρμογής παίζει σημαντικό ρόλο στην ασφάλειας μίας συσκευής Android.

Summary

The most of the android applications has a positive impact in our everyday life, but there some few illegal android’s applications that could destroy our device even can steal personal information which is already saved in the phone or in a personal computer. The purpose of my project is to find the differences between the illegal and the legal androids applications. I studied 25 applications that are coming from illegal and legal androids specifically, I was recording the permissions of each application that is needed to accept by its user. Also, I tested the applications in KitKat, Lollipop and Marshmallow versions in respect to RAM, CPU and Network Traffic. My results will help android developers to create a safe environment for the future devices.

Ευχαριστίες

Με την ολοκλήρωση των μεταπτυχιακών μου σπουδών θα ήθελα να ευχαριστήσω και να αφιερώσω την μεταπτυχιακή μου διατριβή στους γονείς μου Ανδρέα Ιωάννου και Ευτυχία Ιωάννου για όλα τα εφόδια που μου προσέφεραν αλλά και για την συνεχή ενθάρρυνση τους. Ένα μεγάλο ευχαριστώ στα αδέλφια μου Φίλιππο Ιωάννου και Έλενα Ιωάννου που με το δικό τους τρόπο συνέβαλαν στην ολοκλήρωση αυτής της εργασίας.

ΠΕΡΙΕΧΟΜΕΝΑ

Κεφάλαιο 1
Εισαγωγή

Κοινή διαπίστωση όλων πως η εξέλιξη της τεχνολογίας αποτελεί αναπόσπαστο κομμάτι της καθημερινότητας μας. Αντιληπτό γίνεται πως ο κόσμος των δικτύων σε συνδυασμό με την εξέλιξη της επιστήμης των υπολογιστών αποτελούν το έναυσμα για τη δημιουργία των έξυπνων συσκευών όπως τα smartphones. Στις μέρες μας και εμείς σαν μέλη αυτής της κοινωνίας της τεχνολογίας πραγματοποιούμε πολλές προσωπικές και επιχειρηματικές συναλλαγές μέσω αυτών των συσκευών οι οποίες παρουσιάζουν κοινές δυνατότητες και χαρακτηριστικά με έναν υπολογιστή.

Η πλατφόρμα Android αποτελεί ίσως το πιο δημοφιλές λειτουργικό σύστημα στις κινητές συσκευές. Αξίζει να σημειωθεί πως το 37.23% των κινητών συσκευών ανά το παγκόσμιο έχουν λειτουργικό σύστημα Android. Διαθέτει χιλιάδες επίσημες και ανεπίσημες εφαρμογές. Την ίδια στιγμή όμως το λειτουργικό σύστημα Android κατέχει την πρωτιά στους ιούς στις κινητές συσκευές. Το κακόβουλο λογισμικό αποτελεί σοβαρό κίνδυνο για να πληγεί η εμπιστευτικότητα, η ακεραιότητα και η διαθεσιμότητα. Έτσι για να αντιμετωπιστούν ουσιαστικά τα κακόβουλα λογισμικά θα πρέπει να μελετηθούν οι τεχνοτροπίες τους και την ίδια στιγμή να κατανοηθούν οι στόχοι τους και ο τρόπος με τον οποίο μεταδίδονται. Κρίνεται λοιπόν επιτακτική ανάγκη η μελέτη της συμπεριφοράς του κακόβουλου λογισμικού σε λειτουργικά συστήματα Android

ώστε να μπορέσουμε να διατηρήσουμε την ασφάλεια των συσκευών μας και την ίδια στιγμή την δική μας [03]. Είναι γεγονός πως το Android είναι το πιο ευρέως χρησιμοποιούμενο λειτουργικό σύστημα, σχεδιασμένο για συσκευές όπως smartphones και tablets. Αξίζει να σημειωθεί πως οι συσκευές με Android σημειώνουν τα μεγαλύτερα ποσοστά πωλήσεων συγκριτικά με συσκευές Windows, IOS και MAC OS X [07]. Έτσι για τους εξεταστές ψηφιακών τεκμηρίων το λογισμικό Android αποτελεί ιδανική πρόκληση.

Το Android λειτουργικό σύστημα συνδυάζει χαρακτηριστικά όπως η αποτελεσματική κοινόχρηστη μνήμη, Multitasking, αναγνωριστικά χρήστη Unix(UIDs) και δικαιώματα αρχείων. Έτσι προκύπτει ένα μοντέλο ασφάλειας που μοιάζει με multi-user Server. Οι κινητές πλατφόρμες αποκτούν ολοένα και μεγαλύτερη σημασία και σημειώνουν περισσότερες και πιο σύνθετες απαιτήσεις που περιλαμβάνουν και κανόνες συμμόρφωσης [02]. Επιτρέπει την δημιουργία εφαρμογών που χρησιμοποιούν λειτουργίες του τηλεφώνου και την ίδια ακριβώς στιγμή προστατεύει τους χρήστες από πιθανά σφάλματα που μπορεί να προκύψουν, προστατεύοντας τους από κακόβουλο λογισμικό. Μέσω των χαρακτηριστικών που διαθέτει το Android προσπαθεί να εξαλείψει πιθανές επιπτώσεις που μπορεί να επιφέρει ένα κακόβουλο λογισμικό. Η ελαχιστοποίηση της έκτασης της ζημιάς, μπορεί να πραγματοποιηθεί μέσω της απαίτησης άδειας του χρήστη για προγράμματα που πιθανόν να αιτούνται ενέργειες που μπορεί να αποβούν ιδιαιτέρως επικίνδυνες. Μερικά παραδείγματα τέτοιου είδους ενεργειών μπορεί να είναι η εκτέλεση απευθείας κλήσεων, η αποκάλυψη ή παραβίαση προσωπικών δεδομένων του χρήστη, η καταστροφή δεδομένων, διευθύνσεων, επαφών email κ.ά. Ελαχιστοποιώντας τα δικαιώματα που χρησιμοποιεί μία εφαρμογή ταυτόχρονα ελαχιστοποιούνται και πιθανές συνέπειες που μπορεί να προκύψουν από τα τυχόν κενά ασφάλειας που πιθανότατα να υπάρχουν [64]. Τα δικαιώματα έχουν ένα επίπεδο προστασίας που ονομάζεται Protection Level, υπάρχουν έξι επίπεδα προστασίας (AndroidManifest Permission) των δικαιωμάτων αυτά είναι: Normal, Dangerous, Signature, SignatureOrSystem, System και Development.

Οι κινητές συσκευές κινδυνεύουν από ένα μοντέλο απειλών το οποίο περιλαμβάνει τρεις τύπους απειλών: spyware, grayware και malware. Τα τρία αυτά είδη διαφοροποιούνται μεταξύ τους ως προς τους φορείς επίθεσης που παρουσιάζουν διαφορετικά κίνητρα και απαιτούν διαφορετικούς μηχανισμούς άμυνας. Το κακόβουλο λογισμικό αποκτά πρόσβαση σε μία συσκευή με σκοπό την κλοπή δεδομένων, την πρόκληση ζημίας στη συσκευή ή και την ενόχληση του χρήστη. Πιθανότατα να εξαπατήσει τον χρήστη της συσκευής και να εγκαταστήσει μία κακόβουλη εφαρμογή ή να αποκτήσει μη εξουσιοδοτημένη απομακρυσμένη πρόσβαση εκμεταλλευόμενος μία ευπάθεια της συσκευής ή και του λογισμικού. Οι ιοί, τα Worms, τα Trojans

και τα Botnets αποτελούν μέλη της κατηγορίας του κακόβουλου λογισμικού. Η πιο συχνή μορφή κακόβουλου λογισμικού είναι πλαστές εφαρμογές στις οποίες ο δημιουργός εισάγει κακόβουλο κώδικα σε μερικά σημεία μίας αυθεντικής εφαρμογής.

Ποικιλία εργαλείων υπάρχει στη διάθεση μας για την ανάλυση κακόβουλου λογισμικού με την εφαρμογή Δυναμικής και Στατικής ανάλυσης. Η Στατική ανάλυση είναι η επισκόπηση του κώδικα του κακόβουλου λογισμικού και μπορεί να δώσει πληροφορίες για μία εφαρμογή. Η στατική ανάλυση δύναται να γίνει με τη χρήση Dex2jar, Apktool, , Wireshark, PEiD, TCPView, WinHex, Process Explorer, Winanalysis, Strings. Η Δυναμική ανάλυση είναι η ανάλυση της συμπεριφοράς που μελετά την εφαρμογή καθώς αυτή εκτελείται. Όσον αφορά τη δυναμική ανάλυση μπορεί να επιτευχθεί μέσω Droidbox, Android SDK, Wireshark, Ollydbg, IDA Pro. Ο τρόπος με τον οποίο γίνεται η ανάλυση κακόβουλου λογισμικού θα πρέπει να είναι πολύ προσεγμένος γιατί οι δημιουργοί του malware ενσωματώνουν μηχανισμούς που ελέγχουν εάν οι εφαρμογές τους δρουν σε περιβάλλοντα ανάλυσης.

Κρίνεται λοιπόν επιτακτική ανάγκη για μελέτη και ανάλυση της συμπεριφοράς του κακόβουλου λογισμικού σε λογισμικά Android μέσω των εφαρμογών, ώστε να εντοπιστούν οι αδυναμίες που είναι πιθανόν να εκμεταλλευτούν [44]. Η παρούσα μεταπτυχιακή διατριβή αποτελείται από πέντε κεφάλαια. Στο δεύτερο κεφάλαιο της εργασίας γίνεται μία σύντομη βιβλιογραφική ανασκόπηση στην ανάλυση κακόβουλου λογισμικού, παρουσιάζονται γενικές πληροφορίες για το Λειτουργικό σύστημα Android και αναλύονται οι κακόβουλες συμπεριφορές που μπορεί να το επηρεάσουν και πως μπορούν αυτές να ανιχνευθούν. Στο τρίτο κεφάλαιο περιγράφεται ο σχεδιασμός της πειραματικής διαδικασίας. Ανάλυση των πειραματικών δεδομένων γίνεται στο τέταρτο κεφάλαιο, όπου μελετώνται τα δικαιώματα που συνήθως χρησιμοποιούν διάφορες εφαρμογές Android ανάλογα με την προέλευση τους. Οι εφαρμογές που θα μελετηθούν προέρχονται από το Play Store και από Third Party Markets. Με τη διαδικασία μελέτης 25 όμοιων εφαρμογών από το Google Play Store που θεωρούνται καλοήθεις όπου οι δημιουργοί τους θεωρητικά δεν έχουν κανένα δόλιο σκοπό και από Third Party Stores που είναι κρακαρισμένες, μας δίνεται η ευκαιρία να αντιληφθούμε αν η προέλευση των εφαρμογών σημαίνει και κάτι για το είδος των εφαρμογών αν είναι καλοήθεις ή κακόβουλες και αν υπάρχει κάποιος πονηρός σκοπός από μέρους των κατασκευαστών τους. Στη συνέχεια αναλύεται η χρήση της μνήμης RAM και CPU που γίνεται από τις εφαρμογές ανάλογα με τη προέλευση της εφαρμογής, ώστε να αντιληφθούμε πώς αυτή επηρεάζει μία συσκευή, ταυτόχρονα μελετάται και η επιρροή που δέχονται οι Android εκδόσεις: KITKAT, Lollipop και Marshmallow που είναι κυρίαρχες στις Android συσκευές σήμερα. Με αυτό τον τρόπο κατανοούμε την εξέλιξη της

ποιότητας και της ασφάλειας που σημειώνουν οι εκδόσεις με το πέρασμα των ετών και αν οι κατασκευαστές δρουν για να αντιμετωπίσουν τους κινδύνους που προκύπτουν. Στο τελευταίο μέρος του κεφαλαίου, αναλύεται το Network traffic που σημειώνουν οι εφαρμογές μέσω της μελέτης των TCP και HTTP ports που χρησιμοποιούν οι εφαρμογές κατά τον χρόνο εκτέλεσης τους, παρακολουθώντας την συμπεριφορά τους μπορούμε να αντιληφθούμε τις προθέσεις τους. Τέλος στο πέμπτο κεφάλαιο γίνεται μία σύνοψη των πειραματικών ευρημάτων και παρατίθενται μελλοντικές προεκτάσεις της έρευνας που διεξήχθει.

Κεφάλαιο 2
Βιβλιογραφική Ανασκόπηση

Στο παρόν κεφάλαιο γίνεται μία σύντομη βιβλιογραφική ανασκόπηση στην ανάλυση κακόβουλου λογισμικού σε συσκευές Android. Επιπρόσθετα παρουσιάζονται ιστορικά στοιχεία αλλά κα χαρακτηριστικά του λειτουργικού συστήματος Android. Γίνεται μία αναφορά στα Android Markets αλλά και στις τέσσερις πρόσφατες εκδόσεις Android. Παρουσιάζεται το επίπεδο επιτήρησης από τα δημόσια θεσμικά όργανα. Στην συνέχεια ακολουθούν οι απειλές ασφάλειας που μπορεί να υπάρξουν, αναλύονται οι κακόβουλες συμπεριφορές που μπορεί να υπάρξουν στα Android αλλά και οι τεχνικές ανάλυσης κακόβουλου λογισμικού.

2.1 Ανασκόπηση στην Ανάλυση Κακόβουλου Λογισμικού σε συσκευές Android

Ο εντοπισμός, η κατηγοριοποίηση και εξέταση των κινητών malware ήταν ένα ενδιαφέρον πεδίο έρευνας από την αρχική εμφάνιση των κινητών πλατφορμών. Αρκετά χρόνια πριν από την έλευση των σύγχρονων κινητών πλατφορμών, όπως το iOS και το Android, ο Dagon et al. [16]

παρέχει μια ταξινόμηση των κινητών malware. Παρά το γεγονός ότι τα μοντέλα απειλών περιγράφηκαν για τις παλιές φορητές συσκευές, όπως PDAs.

Η πρώτη έρευνα για τη σημερινή γενιά των κινητών συσκευών χρονολογείται από το 2007. Όταν τα λειτουργικά συστήματα iOS, BlackBerry και Android άρχισαν να εμφανίζονται στην αγορά. Όταν η ασφάλεια έγινε μια σημαντική ανησυχία, πολλές έρευνες ακολούθησαν. Cheng et al., SmartSiren αναπτύξαν: ένα σύστημα ανίχνευσης εισβολής για το λειτουργικό σύστημα Symbian και Windows Mobile [12]. Ένα μέσο παρακολούθησης λειτουργεί με την κινητή συσκευή και συλλέγει και καταγράφει τις κλήσεις του συστήματος από τις εφαρμογές που τρέχουν στο τηλέφωνο. Αυτά τα δεδομένα διαβιβάζονται στους διακομιστές του ερευνητή μέσω του Διαδικτύου. Κάθε συσκευή καταγράφει με τον Proxy και παρέχει πληροφορίες, ώστε ο Proxy να μπορεί να διακρίνει από πού προέρχονται τα δεδομένα μεταξύ διαφορετικών συσκευών. Αυτά τα δεδομένα στη συνέχεια αναλύονται για να μάθουμε αν η συσκευή έχει μολυνθεί με κάποιο κακόβουλο λογισμικό που χρησιμοποιεί είτε SMS ή Bluetooth για να εξαπλωθεί.

Bose et al. πρότεινε επίσης ένα σύστημα ανίχνευσης malware συμπεριφοράς που βασίζεται σε Symbian OS το 2008 [06]. Οι γνωστές οικογένειες των υφιστάμενων κακόβουλων λογισμικών που χρησιμοποιείται για τη συλλογή δεδομένων σχετικά με συμβάντα συστήματος και κλήσεις API και ένα λογικό διάγραμμα ροής κατασκευάζεται και συλλαμβάνεται ως η υπογραφή της συγκεκριμένης σειράς. Στη συνέχεια, κάθε νέο σύστημα ή εφαρμογή συγκρίνεται με αυτή τη βάση δεδομένων των υπογραφών. Εάν βρεθεί μια αντιστοιχία, η αίτηση θεωρείται κακόβουλη.

Ένα άλλο παράδειγμα είναι η έρευνα του Shabtai et al. [55,56], η οποία παρέχει μια περιεκτική αξιολόγηση των μηχανισμών ασφαλείας που παρέχονται από το Android πλαίσιο, αλλά δεν μελετά διεξοδικά άλλες ερευνητικές προσπάθειες για την ανίχνευση και τον μετριασμό των θεμάτων ασφαλείας στην πλατφόρμα Android. Η έρευνα της Zhou et al. [78] αναλύει και χαρακτηρίζει ένα σύνολο 1.260 κακόβουλων λογισμικών Android. Αυτή η συλλογή του κακόβουλου λογισμικού, που ονομάζεται Malware Genome, χρησιμοποιείται στη συνέχεια από πολλούς άλλους ερευνητές για να αξιολογήσουν προτεινόμενες τεχνικές ανίχνευσης κακόβουλου λογισμικού τους

Το pBMDS ήταν ένα άλλο σύστημα που προτάθηκε από Xie et al. που συγκρίνει τις εισροές σε εφαρμογές αντί της ανάγνωσης και τη δημιουργία αρχείων καταγραφής της συμπεριφοράς κακόβουλου λογισμικού [69]. Η βάση του συστήματος αυτού ήταν ότι είσοδοι του χρήστη διαφέρουν από την αυτοματοποιημένη εισαγωγή. Αυτή η διαφορά στη συνέχεια

χρησιμοποιείται για να προσδιορίσει ανώμαλη συμπεριφορά που μπορεί να προκύψει από κακόβουλο λογισμικό της συσκευής.

Ένα άλλο σύστημα παρακολούθησης αναπτύχθηκε από Houmansadr [33]. Το σύστημα αυτό βασιζόταν σε σύννεφο (cloud-based) στο οποία μία emulated συσκευή θα τρέχει στο σύννεφο. Η κινητή συσκευή στέλνει όλη την κυκλοφορία του δικτύου μέσω μιας ελεγχόμενης μεσολάβησης, η οποία αναπαράγει την κυκλοφορία της συσκευής στον εξομοιωτή που τρέχει στο σύννεφο. Ένα σύστημα παρακολούθησης συνδέεται με τον εξομοιωτή όπου αναλύει αυτή την κίνηση και ειδοποιεί τον παράγοντα της παρακολούθησης που εκτελείται στην κινητή συσκευή σχετικά με την κακόβουλη δραστηριότητα, ώστε να λάβει προστατευτική δράση.

Εκτός από αυτά τα γενικά, ανεξάρτητα από την πλατφόρμα οι malware έρευνες, έχουν ένα μεγάλο αριθμό σχετικών ερευνών που περιγράφουν υποπεριοχές της ασφάλειας του Android, ασχολείται κυρίως με συγκεκριμένου τύπου ζητήματα ασφάλειας στην πλατφόρμα Android. Για παράδειγμα, Chin et al. [13] μελέτησαν προκλήσεις ασφαλείας στην Android επικοινωνία μεταξύ εφαρμογής και παρουσιάζονται διάφορες κατηγορίες των πιθανών επιθέσεων στις εφαρμογές. Οι Felt et al. [25] ανέλυσαν τη συμπεριφορά ενός συνόλου του κακόβουλου λογισμικού spread πάνω από iOS, Android και Symbian πλατφόρμες. Αξιολόγησαν επίσης την αποτελεσματικότητα των τεχνικών που εφαρμόζονται από τις επίσημες αγορές εφαρμογών, όπως η Apple AppStore και η Google PlayStore, για την πρόληψη και τον εντοπισμό αυτών των κακόβουλων προγραμμάτων.

Η έρευνα του Sadeghi et al. [51] παρέχει επίσης μια ανάλυση 20 ερευνητικών προσπαθειών που ανιχνεύουν και αναλύουν τα κινητά malware. Ενώ στο επίκεντρο είναι η έρευνα κακόβουλου λογισμικού για διαφορετικές πλατφόρμες κινητών, η περιοχή της ανάλυσης της ασφάλειας του Android δεν έχει μελετηθεί λεπτομερώς. Δεν αναλύει, μεταξύ άλλων, τις ιδιότητες των προσεγγίσεων για την ανίχνευση και την ανάλυση Android malware, ούτε οι τεχνικές για την ανίχνευση ευπάθειας Android.

Κάθε μία από αυτές τις έρευνες επισκόπησης συγκεκριμένων τομέων, όπως είναι η ανάλυση του Android τρωτών σημείων μεταξύ των εφαρμογών ή οικογενειών κακόβουλου λογισμικού Android. Ωστόσο, κανένας από αυτούς δεν παρέχει μια ολοκληρωμένη επισκόπηση της υπάρχουσας έρευνας στον τομέα της ανάλυσης της ασφάλειας του Android ως προς τις διάφορες εκδόσεις του και ούτε πως αυτές επηρεάζονται από τα χαρακτηριστικά των εφαρμογών, όπως η Ram και η Cpu.

2.2.1 Ιστορικά στοιχεία για το λειτουργικό σύστημα Android

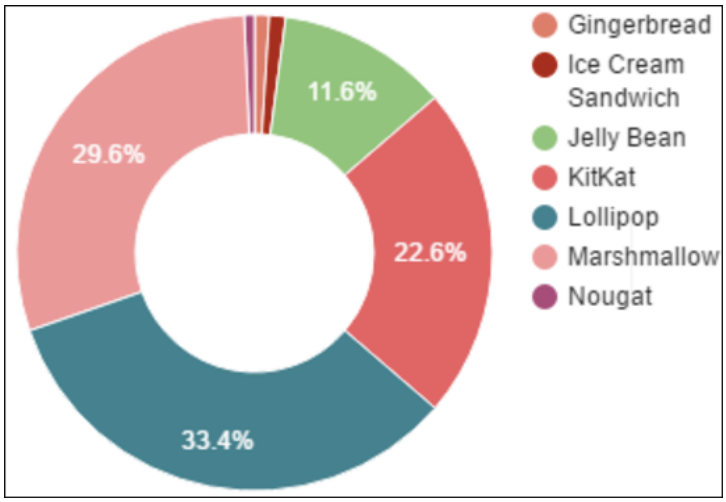
Τα Android αναπτύχθηκαν από μία μικρή εταιρεία την Android Inc. [80] , η οποία ιδρύθηκε στην Καλιφόρνια το 2003 με σκοπό την ανάπτυξη έξυπνων κινητών συσκευών οι οποίες θα γνωρίζουν με τον καλύτερο τρόπο την τοποθεσία και τις προτιμήσεις του ιδιοκτήτη τους. Τον Αύγουστο του 2005 η Google εξαγόρασε την Android Inc., πολλοί υπέθεσαν πως επιθυμεί να εισέλθει στην αγορά κινητής τηλεφωνίας [80]. Η Google ανέπτυξε μία πλατφόρμα κινητών συσκευών που τροφοδοτείται από τον πυρήνα Linux. Ακόμη προώθησε μία πλατφόρμα για κατασκευαστές και συνεχίζει τη πορεία της με την υπόσχεση να παρέχει ένα ευέλικτο σύστημα με δυνατότητες αναβάθμισης [10,15,81]. Στις 5 Νοεμβρίου 2007 ιδρύθηκε η Open Handset Alliance (OHA), μία κοινοπραξία 48 τηλεπικοινωνιακών εταιρειών λογισμικού καθώς και κατασκευής υλικού, οι οποίες στοχεύουν στην ανάπτυξη και εξέλιξη ανοικτών προτύπων στις συσκευές κινητής τηλεφωνίας. Επιπλέον συμπεριλαμβάνονται κατασκευάστριες εταιρείες όπως η HTC, η Sony, η Samsung καθώς και άλλοι φορείς όπως η Sprint και Nextel T-Mobile και κατασκευαστές chipset όπως Qualcomm και Texas Instruments [15,81]. Πλέον η Open Handset Alliance είναι μία κοινοπραξία 84 εταιριών στην οποία συμπεριλαμβάνονται η Google, η Dell, η Intel, η Nvidia, η Wind River Systems κ.ά. [81].

Από το 2008, το Android έχει υποστεί αρκετές αναβαθμίσεις οι οποίες έχουν βελτιώσει σταδιακά το λειτουργικό σύστημα με την προσθήκη νέων χαρακτηριστικών και διορθώσεων σφαλμάτων των προηγούμενων εκδόσεων. Κάθε σημαντική έκδοση ονομάζεται με αλφαβητική σειρά με το όνομα ενός επιδόρπιου ή ζαχαρωτού. Πιο κάτω παρουσιάζονται οι εκδόσεις του λειτουργικού συστήματος Android [82].

Code name	Version number	Initial release date	API level
Alpha	1.0	September 23, 2008	1
Beta	1.1	February 9, 2009	2
Cupcake	1.5	April 27, 2009	3
Donut	1.6	September 15, 2009	4
Eclair	2.0 – 2.1	October 26, 2009	5 – 7
Froyo	2.2 – 2.2.3	May 20, 2010	8
Gingerbread	2.3 – 2.3.7	December 6, 2010	9 – 10
Honeycomb	3.0 – 3.2.6	February 22, 2011	11 – 13
Ice Cream Sandwich	4.0 – 4.0.4	October 18, 2011	14 – 15
Jelly Bean	4.1 – 4.3.1	July 9, 2012	16 – 18
KitKat	4.4 – 4.4.4	October 31, 2013	19
Lollipop	5.0 – 5.1.1	November 12, 2014	21 – 22
Marshmallow	6.0 – 6.0.1	October 5, 2015	23
Nougat	**7.0 – 7.1.1**	**August 22, 2016**	**24 – 25**

Πίνακας 2.1: Εκδόσεις Android λειτουργικού συστήματος [82].

Σύμφωνα με την παγκόσμια κατανομή (Γράφημα 2.1) των εκδόσεων Android από τον Ιανουάριο του 2017, το 33.4% των συσκευών έχουν λειτουργικό σύστημα Lollipop και δεύτερο έρχεται το Marshmallow [82].



Γράφημα 2.1: Παγκόσμια κατανομή εκδόσεων Android μέχρι τον Ιανουάριο του 2017.

2.2.2 Χαρακτηριστικά των Android

Το περιβάλλον ανάδρασης του Android με τον εκάστοτε χρήστη βασίζεται στην άμεση χειραγώγηση, καθώς χρησιμοποιεί ως εντολές εισόδου στο λειτουργικό σύστημα , εντολές αφής

οι οποίες αντιστοιχούν σε πραγματικές δράσεις στην οθόνη του κινητού τηλεφώνου. Οι δράσεις αυτές αντιστοιχούν σε κινήσεις όπως σύρσιμο, τσίμπημα και αντίστροφο τσίμπημα προκειμένου να μετακινηθούν και να τοποθετούν επί της οθόνης διάφορα αντικείμενα όπως είναι εικονίδια ή ένα εικονικό πληκτρολόγιο. Ταυτόχρονα αυτές οι δράσεις μπορεί να συνοδεύονται από άλλες δυνατότητες του συστήματος όπως η δόνηση της συσκευής και η οπτική ανάδραση του χρήστη. Η αρχική οθόνη των Android είναι παρόμοια με την επιφάνεια των υπολογιστών. Στην πραγματικότητα αποτελεί ένα κόμβο ο οποίος περιλαμβάνει την πλοήγηση στο περιβάλλον Android καθώς και ποικιλία πληροφοριών.

Αξίζει να σημειωθεί πως στο Google Play αλλά και σε άλλα καταστήματα εφαρμογών υπάρχει μεγάλη ποικιλία διαθέσιμων εφαρμογών προς το χρήστη, ο οποίος μπορεί να τις κατεβάσει και στη συνέχεια να τις χρησιμοποιήσει. Είναι ολοφάνερο πως οι εφαρμογές αποτελούν ένα σημαντικό τμήμα των Android. Οι εφαρμογές αυτές επεκτείνουν τη λειτουργικότητα των συσκευών και έχουν γραφτεί σε γλώσσα προγραμματισμού Java, χρησιμοποιώντας το Kit ανάπτυξης λογισμικού Android (SDK). Το SDK αποτελείται από μία πλήρη σειρά εργαλείων ανάπτυξης, εντός των οποίων περιλαμβάνεται ένα πρόγραμμα εντοπισμού σφαλμάτων, βιβλιοθήκες λογισμικού, μία συσκευή η οποία εξομοιώνει το περιβάλλον του Android και που βασίζεται στο QEMU, σε τεκμηρίωση, δείγματα κώδικα και βοηθητικό υλικό που περιγράφει τον τρόπο λειτουργίας του αναφερόμενου λογισμικού ανάπτυξης (IDE) του Eclipse με χρήση Android Development Tool (ADT) plugin. Ενώ στη συνέχεια, το Δεκέμβριο του 2014 η Google έφερε στην κυκλοφορία το Android Studio. Υπάρχουν και άλλα εργαλεία ανάπτυξης τα οποία είναι διαθέσιμα συμπεριλαμβανομένου του Native Development Kit (NDK) για εφαρμογές ή επεκτάσεις σε άλλες γλώσσες προγραμματισμού όπως η C/C++, το Google App Inventor οπτικό περιβάλλον για αρχάριους προγραμματιστές. Το Android έχει μία μεγάλη ποικιλία εφαρμογών, οι οποίες μπορούν να αποκτηθούν από τους χρήστες με ένα απλό κατέβασμα και εγκατάστασης του APK αρχείου της εφαρμογής στη συσκευή, είτε με λήψη τους χρησιμοποιώντας ένα από τα υπάρχοντα online καταστήματα.

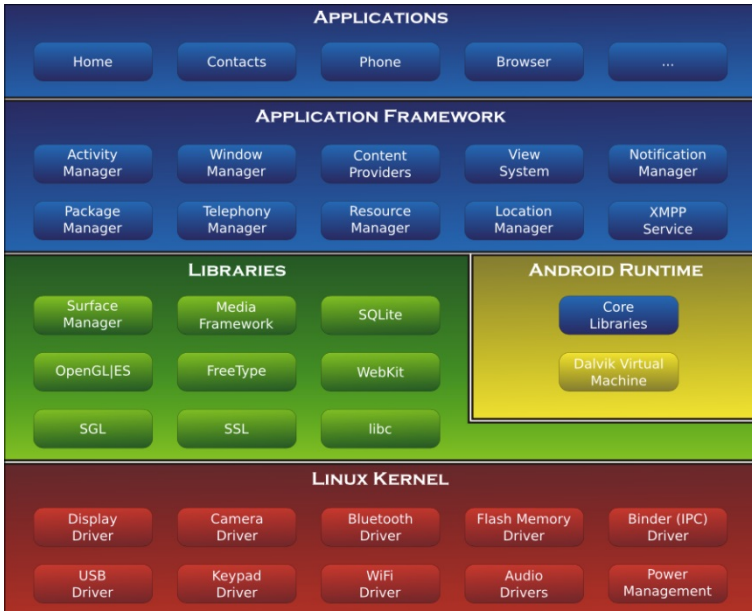
2.2.3 Αρχιτεκτονική του Android

Η πλατφόρμα του Android αποτελείται από μία στοίβα λογισμικού. Τα επίπεδα της στοίβας του Android από το υψηλότερο στο χαμηλότερο περιγράφονται παρακάτω [80]:

1. **Επίπεδο Εφαρμογών (Blue Layer-Application):**Στο επίπεδο αυτό περιλαμβάνεται ένα σύνολο από βασικές εφαρμογές, μερικές από τις οποίες είναι email client, πρόγραμμα sms, ημερολόγιο, χάρτες, επαφές, browser κ.ά. Όλες οι εφαρμογές είναι γραμμένες με χρήση της γλώσσας προγραμματισμού Java.
2. **Επίπεδο Πλαισίου Εφαρμογών (Blue Layer –Application Framework):**Βρίσκεται κάτω από το επίπεδο Εφαρμογών και αποτελείται από ένα σύνολο συστημάτων και υπηρεσιών:

- Ένα σύνολο από γραφικά στοιχεία (views) για την δημιουργία γραφικού περιβάλλοντος συμπεριλαμβανομένου λιστών (lists), πλεγμάτων (grids), κουτιών κειμένου (text boxes), κουμπιών (buttons) κ.ά.
- Ένα διαχειριστή περιεχομένου (Content Manager) ο οποίος επιτρέπει στις εφαρμογές την πρόσβαση σε δεδομένα άλλων εφαρμογών ή τον διαμοιρασμό των δικών τους δεδομένων με άλλες εφαρμογές.
- Ένα διαχειριστή πόρων (Resource Manager) για την πρόσβαση στους πόρους όπως εικόνες, strings, layout files.
- Ένα διαχειριστή ειδοποιήσεων (Notification Manager) ο οποίος επιτρέπει την προβολή ειδοποιήσεων στη μπάρα κατάστασης (status bar).
- Έναν διαχειριστή δραστηριοτήτων (Activity Manager) ο οποίος διαχειρίζεται τον κύκλο ζωής των εφαρμογών.

3. **Επίπεδο βιβλιοθηκών (Green Layer -Libraries):**Το οποίο περιλαμβάνει ένα σύνολο από βιβλιοθήκες γραμμένες σε C/C++ οι οποίες χρησιμοποιούνται από διάφορα στοιχεία του συστήματος του Android. Οι δυνατότητες που προσφέρουν αυτές οι βιβλιοθήκες είναι προσβάσιμες στους προγραμματιστές δια μέσω του επιπέδου πλαισίου εφαρμογής.
4. **Επίπεδο Εκτέλεσης (Green Layer –Android Runtime):**Το οποίο αποτελείται από ένα σύνολο από βασικές βιβλιοθήκες και την Dalvik Virtual Machine.
5. **Ο Πυρήνας του Linux (Red Layer):**Το Android βασίζεται σε πυρήνα έκδοσης Linux για βασικές υπηρεσίες συστήματος όπως ασφάλεια, διαχείριση μνήμης, διαχείριση διεργασιών, στοίβα δικτύου και οδηγούς συσκευών. Ο πυρήνας λειτουργεί επίσης ως ένα ενδιάμεσο επίπεδο αφαίρεσης μεταξύ της στοίβας λογισμικού και του υλικού.



Εικόνα 2.1: Αρχιτεκτονική δομή του Android λειτουργικού συστήματος [83].

2.3 Android Markets

Τα επίσημα Android Markets φιλοξενούν εκατομμύρια εφαρμογών, εφαρμογές τις οποίες κατεβάζει μεγάλος αριθμός ανθρώπων καθημερινά [83]. Το Android προσφέρει ένα μοντέλο ανοικτής αγοράς όπου καμία εφαρμογή δεν επαληθεύεται από οποιοδήποτε εμπειρογνώμονα ασφάλειας και για αυτό καθίσταται εύκολος στόχος για τους προγραμματιστές ώστε να ενσωματώσουν κακόβουλο περιεχόμενο στις εφαρμογές του. Έτσι τα ευαίσθητα προσωπικά δεδομένα του χρήστη εύκολα μπορούν να τεθούν σε κίνδυνο και να μεταφερθούν σε άλλους χρήστες. Επιπρόσθετα η ύπαρξη third party αγορών εφαρμογών συμβάλλει στη διάδοση κακόβουλου λογισμικού για στα Android. Το επίσημο Android Market χρησιμοποιεί το Bouncer για τη προστασία του από κακόβουλα λογισμικά [41]. Οι προγραμματιστές κακόβουλου λογισμικού επωφελούνται των τρωτών σημείων των εφαρμογών με ανασυσκευασία των δημοφιλών εφαρμογών του Google Play και η διανομή αυτών σε third party αγορές. Τα κακόβουλα λογισμικά περιλαμβάνουν viruses, Trojan horses, adware, backdoors, spyware και άλλα κακόβουλα προγράμματα που σχεδιάστηκαν για να βλάψουν το λειτουργικό σύστημα και να κλέψει προσωπικές, οικονομικές ή επιχειρηματικές πληροφορίες.

Σε εργασία που διεξήχθει [59] ανέλυσαν περίπου 6.100 κακόβουλες εφαρμογές και τις ομαδοποιήσαν σε 53 οικογένειες κακόβουλου λογισμικού με τη βοήθεια του API VirusTotal [32]. Σχεδόν το 57% των αναλυθέντων malware οικογενειών προσπάθησαν να κλέψουν τα

προσωπικές πληροφορίες από το smartphone όπως το address book entries, το IMEI ή συντεταγμένες GPS. Επιπλέον, το ποσοστό αποστολής μηνυμάτων SMS είναι περίπου 45%. Τα πιο κοινά είναι η αποστολή αυτών των μηνυμάτων σε αριθμούς υψηλής χρέωσης για να βγάλουν λεφτά αμέσως.

2.4 Android Versions KITKAT, LOLIPOP, MARSHMALLOW και NOUGAT

Η πλατφόρμα Android είναι μία στοίβα λογισμικού που περιλαμβάνει λειτουργικό σύστημα, ενδιάμεσο λογισμικό και τις βασικές εφαρμογές. Η Google με την Open Handset Alliance συνεργάζονται για την ανάπτυξη και την απελευθέρωση του Android. Η τελευταία έκδοση αυτής της πλατφόρμας είναι το Nougat Android 7.1. Το Android 5.0 γνωστό και ως Lollipop να είναι ακόμα πιο ευέλικτο για να στοχεύει Smart watches, TV player, Car media Centre, συσκευές με 512 μνήμη RAM μπορεί να υποστηρίξουν αυτή τη νέα έκδοση.

2.4.1 Πίνακας χαρακτηριστικών Λειτουργικών Συστημάτων

Το Android έχει γίνει ιδιαίτερα δημοφιλές λόγω του ομαλού σχεδιασμού του, καλών αλληλεπιδράσεων, του φιλικού προς το χρήστη λειτουργικό σύστημα και της ελκυστικής εμφάνισης. Το Lollipop διαφέρει από πολλές απόψεις από τις προηγούμενες εκδόσεις του [82].

Στον παρακάτω πίνακα παρουσιάζονται τα χαρακτηριστικά του χρήστη και του προγραμματιστή για τις εκδόσεις του λειτουργικού συστήματος Android KITKAT, LOLIPOP,MARSHMALLOW και NOUGAT [82].

ANDROID KITKAT	Key user features +	Key developer features +
4.4	1. Screen recording	1) Public API for SMS management
4.4.1	2. New translucent system UI	2) Improved memory usage
4.4.2	3. Enhanced notification access	3) Security enhancements (SELinux enforcing mode, new cryptographic algorithms, VPN per user…)
4.4.3	4. System-wide setting for closed captioning	4) NFC Host Card Emulation (for wireless payment, loyalty programs.)
4.4.4	5. Performance improvements	5) Printing framework
	6. Bug fixes	6) Storage access framework

	7. Enhance the camera on the nexus 5 8. Security enhancements 9. Enable sprint spark band 26 and band 41 10. Fix Heartbleed/OpenSSL vulnerability	7) Hardware sensor batching 8) Full screen immersive mode 9) GLES2.0 Surface Flinger 10) Chromium Web View 11) Audio tunnelling to DSP 12) Audio monitoring 13) WIFI Tunnelled Direct Link Setup (TDLS) support 14) Tools for analysing memory use (procstats, on device memory status and profiling)
ANDROID Lollipop	**Key user features +**	**Key developer features +**
5.0, 5.0.1 5.0.2 5.2 5.1.1	1. New design (material design) 2. Speed improvement 3. Battery consumption improvement 4. Bug fixes, fix issues with video playback and password failures 5. Performance improvements 6. Multiple sim card support 7. Quick settings shortcuts to join Wi-Fi networks or control Bluetooth devices 8. Lock protection if lost or stolen 9. High definition voice call 10. Stability and performance enhancements	1) Several new API 2) Tracking battery consumption app
ANDROID Marshmallow	**Key user features +**	**Key developer features +**
6 6.0.1	1. New emojis 2. USB Type-C support 3. Fingerprint authentication support 4. Better battery life with “deep sleep” 5. Permissions dashboard 6. Android play 7. MIDI support 8. Google now improvements	1) Custom Chrome Tabs for better in app browser support 2) App permissions management update

ANDROID Nougat	Key user features +	Key developer features +
7.0 7.1	1. Daydream virtual reality mode 2. Night light 3. Storage manager improvements 4. Performance improvements for touch and display managements 5. Option to enable fingerprint swipe down gesture 6. Seamless system updates 7. Unicode 9.0 emoji 8. Better multitasking 9. Multi window mode (PIP, Freeform window) 10. Seamless system updates (with dual system partition) 11. Better performance and code size thanks to new JIT compiler	1) Shortcut manager APIs 2) Support Circular app icons 3) Keyboard image insertion 4) VR thread scheduling improvements 5) Enhanced wallpaper metadata 6) Multi-endpoint call support 7) Source type support for visual voicemail 8) Carrier config options to manage video telephony 9) Sustained performance mode (SPM) API 10) Vulkan 3d rendering API 11) Daydream virtual reality platform

Πίνακας 2.2: Χαρακτηριστικά του χρήστη και του προγραμματιστή για τις εκδόσεις του λειτουργικού συστήματος Android KITKAT, LOLIPOP,MARSHMALLOW και NOUGAT.

2.4.2 Διαφορές ανάμεσα στις εκδόσεις KitKat και Lollipop

1. Υλικός Σχεδιασμός: Το Android έχει ενισχυθεί πολύ από την άποψη του σχεδιασμού στην νέα έκδοση του 5.0. Η Google έχει εργαστεί για το σχεδιασμό του υλικό του εδώ και φαίνεται ότι τα έχει τελικά εγκαταστήσει στη Lollipop έκδοση.

3. Συνδεσιμότητα: Η συνδεσιμότητα πήρε μια διαφορετική διάσταση στο Android Lollipop με τη συμβατότητα του Android TV, στην τελευταία ενημέρωση. Η τελευταία έκδοση του Android 5.0 είναι στο ίδιο επίπεδο με όλες τις συσκευές, αρχής γενομένης από smartphones στο tablet. Με 4G, το Lollipop είναι αρκετά ικανό να προσαρμοστεί με τη νέα γενιά της επικοινωνίας. Η Google ενισχύει ακόμη τα χαρακτηριστικά Bluetooth και Wi-Fi στο Lollipop. Η ασύρματη σύνδεση Wi-Fi στο lollipop μπορεί να συνδεθεί μόνο με external και untrusted network αν και μόνο αν υπάρχει η σύνδεση. Αυτή είναι μια σημαντική εξέλιξη από το KitKat όπου η πραγματική σύνδεση δεν έχει επαληθευτεί. Σχεδόν 5.000 νέα API έχουν ενσωματωθεί στην έκδοση Android Lollipop και έχει κάνει το Android πιο ευέλικτο από KitKat.

4. Βελτιωμένη ασφάλεια: Το Android Lollipop έχει μερικά νέα χαρακτηριστικά ασφαλείας τα οποία είναι ιδιαίτερα χρήσιμα για την καλύτερη κρυπτογράφηση. Η κρυπτογράφηση έχει ενεργοποιηθεί για όλες τις συσκευές Lollipop. Έχει προστεθεί κάποια προστασία από κακόβουλο λογισμικό. Για αυτό το SELinux έχει εκτελεστεί για όλες τις εφαρμογές. Ένα μενού με νέες ρυθμίσεις έχει αντικαταστήσει το παλαιότερο του KitKat και πολλαπλά προφίλ χρηστών πλέον υποστηρίζονται σε μία μόνο συσκευή. Αυτά τα χαρακτηριστικά δεν ήταν σε KitKat και σίγουρα Lollipop είναι σε θέση να προσθέσει τέτοια μεγάλα πράγματα.

5. Επιδόσεις: Η Google έχει κάνει το Lollipop πιο προηγμένο από το KitKat, έτσι ώστε η κάθε Lollipop ενεργοποιημένη συσκευή να είναι καλύτερη από οποιαδήποτε άλλη. Προαιρετική το ART KitKat είναι τώρα η de-facto πρότυπο για το Lollipop 5.0 και βοηθά να έχουμε μια πιο ομαλή εμπειρία από ό, τι πριν. Οι επεξεργαστές στα Lollipop καταναλώνουν μικρότερη ισχύ και παρέχει μεγαλύτερη διάρκεια ζωής της μπαταρίας. Επίσης, υποστηρίζει RAW images και Multitasking χαρακτηριστικό που κάνει πραγματικά ένα Lollipop smartphone με προηγμένη δυνατότητα. Τα τηλέφωνα έρχονται με μεγαλύτερη μνήμη RAM, οι συσκευές Lollipop είναι έτοιμες να εκτελέσουν πολύ καλύτερα από αυτά που KitKat.

2.4.3 Διαφορές ανάμεσα στις εκδόσεις Lollipop και Marshmallow

1. Battery Life: Το Lollipop εισήγαγε το Project Volta, μια προσπάθεια για να αξιοποιήσουμε στο έπακρο την τελευταία σταγόνα μπαταρίας, ενώ εξακολουθούν να παραμένει αποτελεσματική. Αυτό σήμαινε μια 90-λεπτών εκτεταμένη χρήση με τη λειτουργία εξοικονόμησης μπαταρίας. Το Marshmallow πηγαίνει ένα βήμα παραπέρα με Doze. Doze είναι βασικά μια ρύθμιση η οποία ενεργοποιείται όταν η συσκευή μας δεν χρησιμοποιείται για παρατεταμένο χρονικό διάστημα. Το σύστημα μπαίνει σε μια βαθιά κατάσταση ύπνου.
2. Performance: Η απόδοση είναι σχεδόν παρόμοια και στις δύο εκδόσεις του λειτουργικού συστήματος, αλλά Marshmallow παίρνει ένα βήμα παραπέρα με την καλύτερη διαχείριση της μνήμης. Επίσης δεν πρέπει να ξεχνάμε την επιλογή διαχειριστή RAM για Marshmallow που κάνει σπουδαία δουλειά στην ενημέρωση.
3. App Permissions: Αυτή είναι μία από τις μεγαλύτερες διαφορές μεταξύ των δύο παραλλαγών. Το Lollipop, όπως και οι άλλες εκδόσεις του Android, δείχνουν App Permissions μόνο στο Play Store listing και install time, που ως επί το πλείστων

παραμελείτε από τους χρήστες. Το Marshmallow το κάνει αυτό καλύτερα δίνοντας στους χρήστες πιο κατατοπιστικό έλεγχο πάνω στο τι χρειάζονται τα Permission Apps. Τώρα, αντί να συμφωνηθεί με τη ψεύτικη λίστα των αδειών, είναι σε θέση να χορηγήσει ή να αρνηθεί τις άδειες τους ανάλογα με το τι η εφαρμογή χρειάζεται πραγματικά.

4. Flex storage: Μέχρι τώρα το Android έπαιξε καλά με εξωτερικά μέσα αποθήκευσης, αλλά ήταν δύο ξεχωριστές οντότητες. Πράγμα που σημαίνει οι εφαρμογές θα εγκατασταθούν στον εσωτερικό χώρο αποθήκευσης και όταν εξαντληθεί ο χώρος να τις αποθηκεύσουν εκτός. Το Marshmallow εισήγαγε τη Flex αποθήκευση που σας επιτρέπει να χρησιμοποιείτε την εξωτερική κάρτα SD, ως μέρος του εσωτερικού, διευρύνοντας έτσι στην πραγματικότητα εσωτερική χωρητικότητα αποθήκευσης σας.

5. Authentication: Τόσο οι πλατφόρμες που κυκλοφορούν φέρουν μαζί τους μια σειρά από νέους τρόπους για την απλούστευση ταυτότητας. Το Lollipop εισήγαγε Smart Lock που χρησιμοποιεί την ανίχνευση προσώπου, αξιόπιστη θέση, ακόμη και την φωνή ως κλειδί. Είναι επίσης σε θέση να ανιχνεύσει αν η συσκευή είναι σε αδράνεια ή κινείται και η συσκευή ξεκλειδώνεται με ευκολία. Το Marshmallow καθίσταται ευκολότερο προσθέτοντας εγγενή υποστήριξη για δακτυλικά αποτυπώματα, ώστε το ξεκλείδωμα της συσκευής σας είναι τώρα πολύ ευκολότερο και ασφαλές.

2.4.4 Διαφορές ανάμεσα σε Marshmallow και Nougat

Δεν υπάρχουν πολλές διαφορές ανάμεσα σε αυτές τις δύο εκδόσεις, ωστόσο υπάρχουν πολλές διαφορετικές ενημερώσεις και αναβαθμίσεις στο Nougat. Υπάρχουν διάφορες ενημερώσεις σε πολλά χαρακτηριστικά, όπως το Marshmallow χρησιμοποιεί Standard notification ενώ το Android Nougat 7.0 επιτρέπει τις modify notifications και opens up App. Το Doze Project της Google ενισχύει την διάρκεια ζωής της μπαταρίας στο Nougat σε σχέση με το Marshmallow. Εμφανισιακά δεν διαφέρουν πάρα πολύ.

2.5 Πεδίο δράσης της επιτήρησης από δημόσια θεσμικά όργανα

Ως μέρος των ευρύτερων αποκαλύψεων που έγιναν το 2013 από την επιτήρηση της μαζικού πληθυσμού τον Σεπτέμβριο του 2013, είναι ότι οι Αμερικανικές και Βρετανικές υπηρεσίες, η

Εθνική Υπηρεσία Ασφάλειας (NSA- National Security Agency) και το Goverment Communications Headquarters (GCHQ), αντίστοιχα έχουν πρόσβαση στα δεδομένα του χρήστη των iPhone, BlackBerry και Android συσκευών. Σημειώνεται ότι είναι σε θέση να διαβάσουν όλες τις πληροφορίες των έξυπνων τηλεφώνων συμπεριλαμβανομένου μηνυμάτων, τοποθεσιών, σημειώσεων και ηλεκτρονικών μηνυμάτων. Περαιτέρω αποκαλύψεις αναφέρουν πως έχουν την ικανότητα να υποκλέπτουν πληροφορίες που μεταδίδονται μέσω του διαδικτύου από τα μέσα κοινωνικής δικτύωσης και άλλες δημοφιλείς εφαρμογές όπως ήταν το Angry Birds [84]. Τα έγγραφα αποκαλύπτουν μία περαιτέρω προσπάθεια των μυστικών υπηρεσιών να παρακολουθούν τις Google maps αναζητήσεις και τα ερωτήματα που υποβλήθηκαν από τις συσκευές Android και για τη συλλογή πληροφοριών τοποθεσίας σε μεγάλες ποσότητες [84].

2.6 Απειλές ασφάλειας

Έρευνα τη εταιρείας ασφάλειας TrendMicro καταγράφει την κατάχρηση των υπηρεσιών ως το πιο κοινό είδος κακόβουλου λογισμικού Android, όπου τα μηνύματα κειμένου αποστέλλονται από μολυσμένα τηλέφωνα σε τηλεφωνικούς αριθμούς υψηλής χρέωσης χωρίς τη συναίνεση ή ακόμα τη γνώση του χρήστη [85]. Άλλο κακόβουλο λογισμικό εμφανίζει ανεπιθύμητες ή ενοχλητικές διαφημίσεις στη συσκευή ή αποστέλλει προσωπικές πληροφορίες σε μη εξουσιοδοτημένους τρίτους [85]. Οι απειλές στις συσκευές Android αυξάνονται με γεωμετρική πρόοδο. Ωστόσο οι μηχανικοί της Google υποστηρίζουν ότι το κακόβουλο λογισμικό και η απειλή των ιών σε Android μεγαλοποιούνται από εταιρείες ασφάλειας για εμπορικούς λόγους [46,86] και κατηγορούν τον κλάδο της βιομηχανίας ασφάλειας ότι σκορπά το φόβο για να πωλήσει λογισμικά προστασίας ιών στους χρήστες [46]. Η Google πιστεύει ότι τα πραγματικά επικίνδυνα κακόβουλα λογισμικά είναι εξαιρετικά σπάνια [86] και σύμφωνα με έρευνα που έχει διεξαχθεί από την F-Secure δείχνει ότι μόνο το 0.5% των Android κακόβουλων λογισμικών προέρχονται από το Google Play Store [87].

Το Android fragmentation είναι πρόβλημα για την ασφάλεια διότι επιδιορθώνει (patches) τα σφάλματα (bugs) που βρέθηκαν στον πυρήνα του λειτουργικού συστήματος, συχνά όμως δεν φτάνει στους χρήστες παλαιότερων και χαμηλότερου κόστους συσκευών [88,27]. Ένα σύνολο από ερευνητές λέει ότι η αποτυχία των προμηθευτών να υποστηρίξουν παλαιότερες συσκευές με επιδιορθώσεις και ενημερώσεις καθιστά πέραν του 87% των ενεργών συσκευών ευάλωτες [37,63]. Τα Android smartphones έχουν τη δυνατότητα να αναφέρουν τη τοποθεσία των Wi-Fi Access points, που σημειώνουν οι χρήστες των τηλεφώνων όσο μετακινούνται για να

δημιουργήσουν βάσεις δεδομένων που περιέχουν τις φυσικές θέσεις εκατοντάδων εκατομμυρίων των εν λόγω Access points. Αυτές οι βάσεις δεδομένων αποτελούν ηλεκτρονικούς χάρτες για τον εντοπισμό των έξυπνων τηλεφώνων, που επιτρέπει την εκτέλεση εφαρμογών όπως το Foursquare, Google Latitude, Facebook Places [89]. Third party λογισμικά παρακολούθησης όπως το TraitDroid [42],σε μερικές περιπτώσεις να ανιχνεύσει προσωπικές πληροφορίες που αποστέλλονται από εφαρμογές σε απομακρυσμένους διακομιστές [90].

2.6.1 Τεχνικά χαρακτηριστικά ασφάλειας Android

Οι εφαρμογές Android τρέχουν σε ένα περιβάλλον Sandbox, μία απομονωμένη περιοχή του συστήματος που δεν έχει πρόσβαση στους υπόλοιπους πόρους του συστήματος εκτός και αν τα δικαιώματα πρόσβασης χορηγούνται ρητά από τη συσκευή όταν έχει εγκατασταθεί η εφαρμογή. Μετά την εξέταση των εν λόγω δικαιωμάτων ο χρήστης μπορεί να επιλέξει τα δεχθεί ή να τα απορρίψει εγκαθιστώντας τη εφαρμογή αποδέχοντας την [29]. Το sandboxing (σύστημα περιβάλλοντος δοκιμών) και τα δικαιώματα του συστήματος μειώνουν το αντίκτυπο των τρωτών σημείων και σφαλμάτων στις εφαρμογές αλλά οι προγραμματιστές συγχύζονται και η περιορισμένη τεκμηρίωση οδηγεί τις εφαρμογές στο να ζητούν περιττά δικαιώματα, μειώνοντας έτσι την αποτελεσματικότητα της [91]. Αρκετές επιχειρήσεις παροχής υπηρεσιών ασφάλειας όπως η Lookout Mobile Security [47], η AVG Technologies και η McAfee [92] έχουν επίσης κυκλοφορήσει antivirus λογισμικά για Android συσκευές. Αυτό το λογισμικό είναι αναποτελεσματικό ως sandboxing επίσης εφαρμόζεται τόσο σε εφαρμογές, περιορίζοντας την ικανότητα τους να σαρώσουν βαθύτερα το σύστημα για απειλές [65].

2.7 Κακόβουλες συμπεριφορές στα Android

Για την αποφυγή του κακόβουλου λογισμικού είναι σημαντική η βαθιά και ακριβής κατανόηση τους ώστε να ληφθούν ανάλογα μέτρα προστασίας για προστασία των δεδομένων των χρηστών. Υπάρχουν εκατοντάδες τεχνικές κακόβουλου λογισμικού που επιτίθενται σε πλατφόρμες Android με πολλούς τρόπους όπως η αποστολή μηνυμάτων χωρίς τη γνώση του θύματος και τη διαγραφή αυτών, στέλνοντας προσωπικές πληροφορίες του χρήστη σε άλλους διακομιστές. Επομένως υπάρχει μεγάλη ανάγκη για προστασία των δεδομένων του χρήστη από κακόβουλα λογισμικά.

2.7.1 Ανάλυση Κακόβουλου Λογισμικού

Το εύρος και ο αριθμός των κακόβουλων λογισμικών που ανιχνεύονται αυξάνονται κάθε χρόνο. Σύμφωνα με την TrendMicro, τα κακόβουλα λογισμικά είχαν αυξηθεί σε 7 εκατομμύρια το 1ο μισό του 2015 [93,94]. Η συμπεριφορά των οικογενειών κακόβουλου λογισμικού παρουσιάζεται πιο κάτω.

A. Trojans εμφανίζονται στο χρήστη σαν καλοήθης εφαρμογές [41] ενώ στην πραγματικότητα κλέβουν εμπιστευτικές πληροφορίες του χρήστη χωρίς τη γνώση του. Τέτοιες εφαρμογές έχουν πρόσβαση στο ιστορικό περιήγησης του χρήστη, στα μηνύματα, στις επαφές και στους IMEI αριθμούς της συσκευή και έτσι πραγματοποιείται κλοπή πληροφοριών χωρίς τη συγκατάθεση του χρήστη. Το FakeNetflix [48] είναι ένα παράδειγμα τέτοιου κακόβουλου λογισμικού που παρέχει στο χρήστη διεπαφή παρόμοια με αυτή της αυθεντικής Netflix εφαρμογής και συλλέγει τα διαπιστευτήρια σύνδεσης του χρήστη. Υπάρχουν και τα SMS Trojans που εκμεταλλεύονται κορυφαίες υπηρεσίες για να υποστεί οικονομική ζημία το θύμα. Το Fakeplayer είναι ένα γνωστό SMS Trojan που στέλνει μηνύματα σε αριθμούς υψηλής χρέωσης χωρίς την γνώση των χρηστών [78]. Το Zsone [60] και Android.foney καταγράφουν τραπεζικές πληροφορίες όπως αριθμούς λογαριασμού και κωδικούς.

B. Backdoors απασχολεί εκμεταλλεύσεις root (υπερχρήστη) για να χορηγήσει δικαιώματα υπερχρήστη σε κακόβουλα λογισμικά και τους διευκολύνει να κρυφτούν από antivirus. Τα Exploit, Ragaeagainstthecage (RATC) και Zimperlich είναι οι τρείς πιο δημοφιλείς εκμεταλλεύσεις υπερχρήστη που αποκτούν τον πλήρη έλεγχο μίας συσκευής [95] τα δύο πρώτα είναι κρυπτογραφημένες μορφές. Επιπλέον το DroidKungFu χρησιμοποιεί εκμεταλλεύσεις υπερχρήστη. Όταν το DroidKungFu εκτελεστεί, πρώτα εκτελεί και δρομολογεί τις εκμεταλλεύσεις υπερχρήστη. Αν η εκμετάλλευση αποκτήσει τον πλήρη έλεγχο της συσκευής και τα δικαιώματα του υπερχρήστη, το κακόβουλο λογισμικό είναι σε θέση να εκτελέσει οποιαδήποτε λειτουργία στη συσκευή και να εγκαταστήσει εφαρμογές χωρίς την συγκατάθεση του χρήστη [79].

C. Worms είναι κακόβουλα λογισμικά που δημιουργούν αντίγραφα του εαυτού τους και τα διανέμουν μέσω του δικτύου. Για παράδειγμα τα Bluetooth worms εξαπλώνουν μέσω των Bluetooth δικτύου κακόβουλο λογισμικό. Ένα τέτοιου είδους worm είναι το Android.Obad. OS [96].

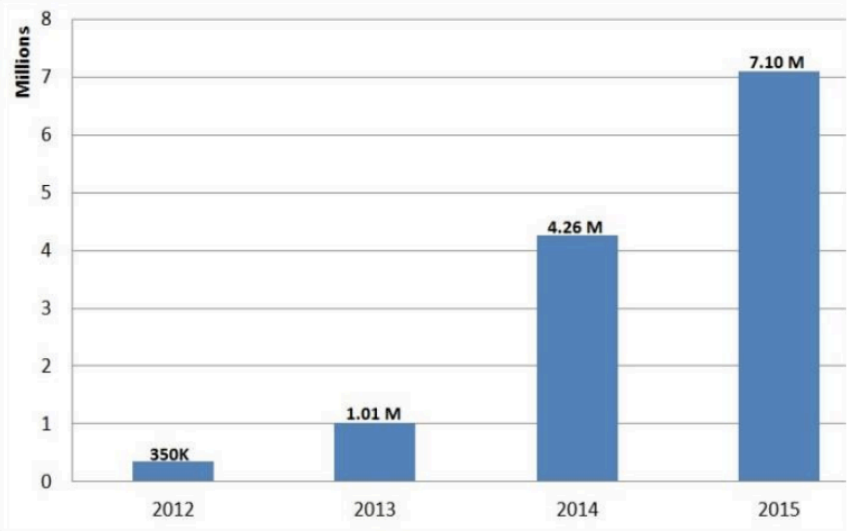
D. Spyware. Nickyspy [78] και το GPSSpy [11] είναι παραδείγματα Spyware εφαρμογών που εμφανίζονται σαν καλοήθεις εφαρμογές αλλά στην πραγματικότητα

παρακολουθούν εμπιστευτικές πληροφορίες του χρήστη όπως μηνύματα, τοποθεσίες. Personal spywares μπορούν να εγκαταστήσουν φορτίο κακόβουλου λογισμικού χωρίς τη γνώση του χρήστη, στέλνει πληροφορίες όπως μηνύματα και επαφές του χρήστη στον επιτιθέμενο.

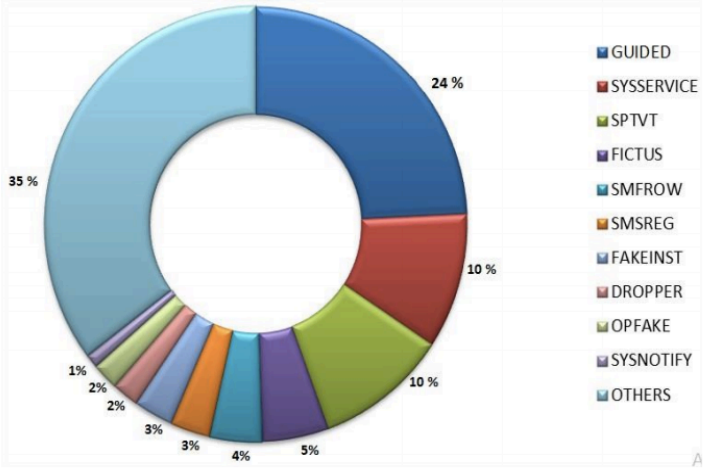
E. Botnets είναι ένα δίκτυο που βάζει σε κίνδυνο τις Android συσκευές. Το Botmaste είναι ένας απομακρυσμένος διακομιστής που ελέγχει το botnet μέσω C&C δικτύου. Το Geinimi [78] είναι ένα Android Botnet.

F. Ransomwares αποτρέπουν τον χρήστη από το να έχει πρόσβαση στα δεδομένα της συσκευής με κλείδωμα της μέχρι να πληρωθεί ένα ποσό λύτρων. Το FakeDefender είναι ένα κακόβουλο λογισμικό το οποίο μεταμφιέζεται στο antivirus Avast!, κλειδώνει τη συσκευή του θύματος και αναγκάζει το χρήστη να πληρώσει για να ξεκλειδωθεί.

G. Riskwares είναι νόμιμο λογισμικό εκμετάλλευσης από συγγραφείς κακόβουλου λογισμικού οι οποίοι στοχεύουν στη μείωση της απόδοσης της συσκευής ή να βλάψουν δεδομένα με διαγραφή, αντιγραφή ή τροποποίηση τους [97].



Γράφημα 2.2: Δείχνει την αύξηση των κακόβουλων λογισμικών με το πέρας των ετών μέχρι το 2015.



Γράφημα 2.3: Αναπαράσταση των πιο δημοφιλών οικογενειών κακόβουλου λογισμικού που καταγράφηκαν από την TrendMicro το 2ο τρίμηνο του 2015.

2.7.2 Κακόβουλες Τεχνικές Διείσδυσης.

a. *Repackaging (Ανασυσκευασία-Ανακατασκευή*). Οι συγγραφείς κακόβουλου λογισμικού ανασκευάζουν δημοφιλείς εφαρμογές του Android official market, το Google Play και τις κατανέμουν σε λιγότερό δημοφιλείς αγορές Android. Το repackaging περιλαμβάνει αποσυναρμολόγηση των γνωστών καλοήθη εφαρμογών, η οποία γίνεται με τη χρήση reverse engineering εργαλείων. Κατά τη διάρκεια αυτής της διαδικασίας οι συγγραφείς κακόβουλου λογισμικού αλλάζουν την υπογραφή της επανασυσκευασμένης εφαρμογής και έτσι η εφαρμογή φαίνεται νέα στο antimalware. Η TrendMicro αναφέρει πως το 77% των 50 πιο δημοφιλών εφαρμογών που διατίθενται στο Google Play είναι ανασκευασμένα [71].

b. *Drive by Download*, αναφέρεται στην ακούσια λήψη κακόβουλου λογισμικού στο παρασκήνιο. Έτσι προκύπτουν επιθέσεις όταν ο χρήστης επισκέπτεται μία ιστοσελίδα που περιέχει κακόβουλο λογισμικό και το εγχέει στη συσκευή του θύματος χωρίς τη γνώση του πάντα. Οι συγγραφείς κακόβουλου λογισμικού χρησιμοποιούν το Android/NotCompatible [98] που είναι μία Drive by Download εφαρμογή.

c. *Dynamic Payloads,* είναι κακόβουλα λογισμικά που διαπερνούν τις συσκευές Android μέσω τεχνικών Dynamic payload. Κρυπτογραφούν το κακόβουλο περιεχόμενο και ενσωματώνει APK πόρους. Μετά την εγκατάσταση η εφαρμογή αποκρυπτογραφεί το κρυπτογραφημένο κακόβουλο φορτίο και εκτελεί κακόβουλο κώδικα. Μερικά κακόβουλα λογισμικά εγκαθιστούν το κακόβουλο περιεχόμενο δυναμικά από

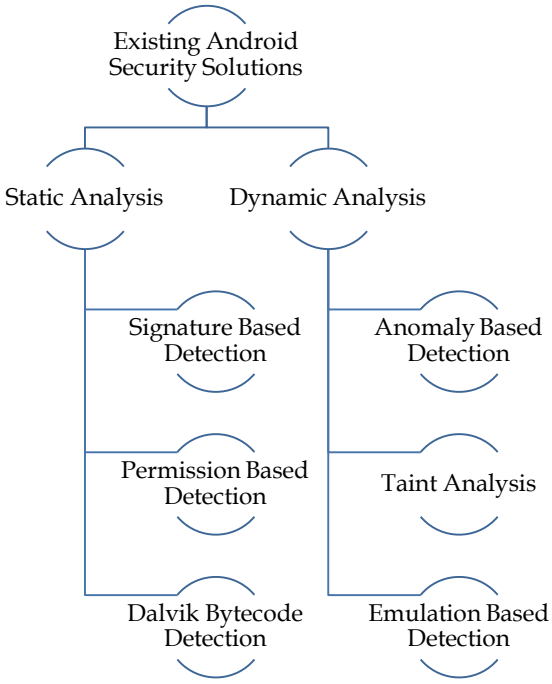
απομακρυσμένους διακομιστές και δεν ανιχνεύεται με προσέγγιση στατικής ανάλυσης [17].

d. *Stealth Malware Techniques*, οι σαρωτές συσκευών Android κακόβουλου λογισμικού δεν μπορούν να εκτελούν βαθιά ανάλυση επειδή υπάρχει περιορισμένη διαθεσιμότητα πόρων όπως μπαταρία. Οι προγραμματιστές κακόβουλου λογισμικού εκμεταλλεύονται τα τρωτά σημεία του υλικού και συσκοτίζει τον κακόβουλο κώδικα για να παρακάμψει εύκολα το antimalware. Διαφορετικές Stealth Techniques είναι: Key permutation, dynamic loading, native code execution, code encryption και java reflection.

2.8 Τεχνικές ανάλυσης κακόβουλου λογισμικού

Οι συνεχώς αυξανόμενες απειλές κακόβουλου λογισμικού έχουν αναγκάσει την antimalware βιομηχανία να αναπτύξει λύσεις για μετριασμό του κινδύνου κακόβουλης εφαρμογής σε Android smartphones και άλλες Android συσκευές. Δύο προσεγγίσεις χρησιμοποιούνται για αυτό το σκοπό: Στατική και Δυναμική προσέγγιση. Τα antivirus προγράμματα χρησιμοποιούν αυτές τις προσεγγίσεις για να προστατέψουν τα κινητά συστήματα από κακόβουλες επιθέσεις. Ανιχνεύουν κακόβουλες εφαρμογές και ειδοποιούν το χρήστη για τις εν λόγω εφαρμογές, λαμβάνουν μέτρα όπως διαγραφή των κακόβουλων λογισμικών. Σαν αποτέλεσμα της απειλής, του κακόβουλου λογισμικού και του μηχανισμού προστασίας που προσφέρεται από τα Android antimalware προγράμματα, η γενική κατάσταση του κινδύνου των Android χρηστών είναι δύσκολο να εκτιμηθεί.

Υπάρχουν δύο κύριες προσεγγίσεις για τη ανάλυση Android κακόβουλου λογισμικού η Στατική και η Δυναμική προσέγγιση. Περαιτέρω κατηγοριοποιείται το antimalware χρησιμοποιώντας τη Στατική και Δυναμική προσέγγιση.



Εικόνα 2.2: Δείχνει την ταξινόμηση των antimalware τεχνικών.

2.8.1 Στατική ανάλυση

Η Στατική προσέγγιση είναι ένας τρόπος να ελέγξουμε τις λειτουργίες και την κακεντρέχεια της εφαρμογής αποσυναρμολογώντας την και με ανάλυση του πηγαίου κώδικα χωρίς όμως την εκτέλεση της εφαρμογής. Χρήσιμη είναι η εύρεση κακόβουλων συμπεριφορών που μπορεί να μην λειτουργούν μέχρι να βρεθούν σε συγκεκριμένη κατάσταση.

A. *Signature Based Approach,* είναι η μέθοδος ανίχνευσης κακόβουλου λογισμικού βασισμένη στην υπογραφή που χρησιμοποιείται κοινώς από εμπορικά προϊόντα antimalware. Αυτή η μέθοδος εξάγει σημασιολογικά πρότυπα και δημιουργεί μία μοναδική υπογραφή [26]. Ένα πρόγραμμα ταξινομείται ως κακόβουλο εάν η υπογραφή του ταιριάζει με μία υπάρχουσα οικογένεια υπογραφών κακόβουλου λογισμικού. Το κύριο μειονέκτημα αυτής της προσέγγισης είναι ότι εύκολα μπορεί να παρακαμφθεί με κώδικα συσκότισης επειδή μπορεί να εντοπίσει τα υπάρχοντα κακόβουλα λογισμικά και αποτυγχάνει απέναντι στις άγνωστες παραλλαγές κακόβουλου λογισμικού. Το AndroSimilar [23] προτάθηκε ως μία ισχυρή στατική μέθοδος υπογραφής για την ανίχνευση άγνωστων παραλλαγών των υπάρχων κακόβουλων λογισμικών, που συνήθως δημιουργούνται με χρήση επανασυναρμολόγησης και τεχνικών κώδικα συσκότισης. Δημιουργεί μεταβλητό μήκος υπογραφής για την υπό δοκιμή υπογραφή και την συγκρίνει με τις υπογραφές στην AndroSimilar βάση δεδομένων κακόβουλου

λογισμικού και ταυτοποιείται η εφαρμογή ως κακοήθης ή ως καλοήθης βάσει ποσοστού ομοιότητας. Το DroidAnalytics [76] είναι ένα ακόμα σύστημα ανάλυσης βασισμένο στην υπογραφή το οποίο εξάγει και αναλύει εφαρμογές σε op-code επίπεδο. Όχι μόνο δημιουργεί την υπογραφή αλλά συσχετίζει το κακόβουλο λογισμικό μετά τα ήδη υπάρχοντα μετά την αναγνώριση του κακόβουλου περιεχομένου. Ο βαθμός ομοιότητας χρησιμεύει για την ανίχνευση των ανασκευασμένων κακόβουλων λογισμικών, δεν είναι 100% λύση, μπορεί να παρέχει false positive ακόμη να κατατάσσει νόμιμες εφαρμογές σαν κακοήθεις.

Περιορισμοί της Signature Based μεθόδου: Παρόλο που η ανίχνευση βάσει υπογραφής είναι πολύ αποτελεσματική για γνωστά κακόβουλα λογισμικά δεν μπορεί να ανιχνεύσει άγνωστους τύπους κακόβουλου λογισμικού. Επίσης υπάρχει περιορισμός στη βάση δεδομένων υπογραφής τα περισσότερα κακόβουλα λογισμικά παραμένουν μη ανιχνεύσιμα.

Β. *Permissions Based Analysis,* στα Android συστήματα τα δικαιώματα που απαιτούνται από την εφαρμογή παίζουν ρόλο ζωτικής σημασίας στα δικαιώματα πρόσβασης που την διέπουν. Από προεπιλογή οι εφαρμογές δεν έχουν άδεια να έχουν πρόσβαση στα δεδομένα του χρήστη και έτσι επηρεάζεται η ασφάλεια του συστήματος. Κατά την διάρκεια της εγκατάστασης ο χρήστης επιτρέπει στην εφαρμογή να έχει πρόσβαση σε όλους του πόρους που ζητά. Οι προγραμματιστές αναφέρουν τα αιτούμενα δικαιώματα για τους πόρους στο AndroidManifest.xml αρχείο. Στην πραγματικότητα το σύνολο των δηλωθέντων δικαιωμάτων δεν είναι απαραίτητα για τη συγκεκριμένη εφαρμογή. Ο Ref [24] δηλώνει ότι τον περισσότερο χρόνο οι προγραμματιστές δηλώνουν τα δικαιώματα που τελικά δεν απαιτούνται από την εφαρμογή, έτσι γίνεται δύσκολα ανιχνεύσιμη η συμπεριφορά του κακόβουλου λογισμικού. Το Antimalware αναλύει το AndroidManifest.xml αρχείο όπου όλα τα αιτούμενα δικαιώματα για τους πόρους της εφαρμογής αναφέρονται. Ο Stowaway [24] εκθέτει τα δικαιώματα πάνω από το προνομιακό πρόβλημα του Android, όπου μία εφαρμογή αιτείται περισσότερα δικαιώματα από ότι χρησιμοποιεί τελικά. Ο Stowaway εκτελεί στατική ανάλυση για τον καθορισμό των API calls (κλήσεων) που η εφαρμογή επικαλείται και στη συνέχεια χαρτογραφεί τα δικαιώματα που απαιτούνται από τις API κλήσεις. Με την ανίχνευση κακόβουλου λογισμικού μηχανισμού [53] το manifest αρχείο αναλύεται και εξάγονται πληροφορίες όπως: Permissions, intent files (action, category και priority), process name και redefined Permissions για την ανίχνευση κακόβουλης

συμπεριφοράς μίας εφαρμογής. Μετά την εξαγωγή αυτών των πληροφοριών μπορούμε να συγκρίνουμε τη λίστα κλειδιών στην προτεινόμενη μέθοδο και στη συνέχεια υπολογίζεται ο βαθμός κακοήθειας. Το Weka [48] είναι εργαλείο εξόρυξης δεδομένων για τον υπολογισμό της τιμής κατωφλίου. Στο τέλος συγκρίνεται ο βαθμός κακοήθειας με την τιμή κατωφλίου και ταξινομεί την εφαρμογή σαν κακοήθη αν ο βαθμός υπερβαίνει την τιμή κατωφλίου. Χρησιμοποιήθηκαν 365 δείγματα για να δοκιμάσουν την αποτελεσματικότητα της προτεινόμενης λύσης και η λύση παρέχει 90% ακρίβεια ανίχνευσης. Ο μηχανισμός εξοικονόμησης κόστους περιλαμβάνει ανάλυση του manifest αρχείου και μπορεί να εφαρμοστεί εύκολα σε όλες τις αρχιτεκτονικές ανίχνευσης για την αποτελεσματική ανίχνευση κακόβουλου λογισμικού. Επιπλέον μπορεί να ανιχνεύσει κακόβουλα λογισμικά που δεν είναι ανιχνεύσιμα με Signature Based μέθοδο ανίχνευσης. Αυτή η λύση περιορίζεται στις πληροφορίες που περιέχει το manifest αρχείο. Ακόμη μπορεί να ανιχνεύσει και δείγματα adware. Οι C.Y Haunget και συνεργάτες [34] πρότειναν μία μέθοδο για καλύτερη ανίχνευση αυτή της ανίχνευσης κακόβουλου λογισμικού βασισμένη στα δικαιώματα, η οποία περιλαμβάνει ανάλυση τόσο των αιτημάτων όσο και των απαιτούμενων δικαιωμάτων. Αναλύει την ευκολία ανάκτησης χαρακτηριστικών και στη συνέχεια ονομάζει την εφαρμογή σαν κακοήθη ή καλοήθη. Το PUMA παρουσιάστηκε από τον Sanz Borja και τους συνεργάτες του [52] για την ανίχνευση κακόβολων εφαρμογών με ανάλυση των αιτούμενων δικαιωμάτων για μία εφαρμογή. Χρησιμοποίησαν Permissions tags όπως: <uses-permission> και <uses-features> που παρουσιάζονται στο AndroidManifest.xml αρχείο για την ανάλυση της κακόβουλης συμπεριφοράς των εφαρμογών. Εφαρμόζονται σε διάφορους αλγόριθμους ταξινόμησης σε μία βάση δεδομένων με 357 καλοήθεις και 249 κακοήθεις εφαρμογές. Αυτή η λύση παρέχει υψηλό ποσοστό ανίχνευσης αλλά τα αποτελέσματα έχουν υψηλά ποσοστά false positives, επίσης δεν είναι επαρκές για την αποτελεσματική ανίχνευση των κακόβουλων λογισμικών και εξακολουθεί να απαιτεί πληροφορίες που σχετίζονται με άλλα χαρακτηριστικά και δυναμική ανάλυση. Ο Shin [58] χρησιμοποιεί μία κατάσταση machine based approach και επισήμως αναλύει το permission based Android security model. Το Security Distance Model προτάθηκε από τον Tang και τους συνεργάτες του [61] για τον μετριασμό του Android κακόβουλου λογισμικού. Είναι βασισμένο στην ιδέα ότι δεν είναι αρκετό μόνο ένα δικαίωμα σε μία εφαρμογή για να αποτελέσει απειλή για την ασφάλεια των Android συσκευών. Δηλαδή ένα αίτημα δικαιώματος εφαρμογής READ_PHONE_STATE μπορεί να έχει πρόσβαση στους αριθμούς τηλεφώνων και στο IMEI αλλά δεν μπορεί να μεταφέρει δεδομένα έξω από τη

συσκευή. Πρέπει να υπάρχει ένας συνδυασμός δικαιωμάτων για να επηρεαστεί το μοντέλο ασφάλειας της συσκευής όπως το δικαίωμα INTERNET που επιτρέπει την ‵ένωση‵ της συσκευής με το δίκτυο και θα χρειαστεί τη μετακίνηση δεδομένων σε κάποιο απομακρυσμένο διακομιστή. Ο Enck και συνεργάτες [22] ανέπτυξαν το KIRIN, είναι ένα εργαλείο που παρέχει μικρό βάρος χρόνου εγκατάστασης. Ορίζει κανόνες ασφάλειας και απλά συγκρίνει τα αιτούμενα δικαιώματα της εφαρμογής, με τους κανόνες ασφάλειας και πιστοποιεί την εφαρμογή σαν κακόβουλη εάν αποτυγχάνει να περάσει όλους τους κανόνες ασφάλειας. Η εγκατάσταση της εφαρμογής ματαιώνεται εάν η εφαρμογή αποδίδεται ως κακόβουλο λογισμικό. Δοκίμασαν 311 εφαρμογές που εγκαταστάθηκαν από το επίσημο Android Market και απέτυχαν να περάσουν τους κανόνες μόνο 5 εφαρμογές. Η προτεινόμενη λύση αναλύει μόνο το αρχείο Manifest.xml. Ο περιορισμός του KIRIN περιλαμβάνει ότι μπορεί να δηλώσει κάποιες νόμιμες εφαρμογές σαν κακόβουλο λογισμικό, επειδή οι πληροφορίες που παρέχονται για την πιστοποίηση της εφαρμογής δεν είναι επαρκής για την ανίχνευση του κακόβουλου λογισμικού. DroidMat [67]είναι ένα εργαλείο που εξάγει πληροφορίες από το manifest αρχείο όπως Permissions, messaging passing through intents and API calls tracing για την ανάλυση της συμπεριφοράς της εφαρμογής. Εφαρμόζει K-Means clustering που αυξάνει την ικανότητα ανίχνευσης κακόβουλου λογισμικού χρησιμοποιώντας το KNN αλγόριθμο [95]. Είναι πιο αποτελεσματικό από το Androguard [18] καθώς χρειάζεται λιγότερο χρόνο για τον εντοπισμός 1738 εφαρμογών σαν κακοήθεις ή καλοήθεις.

Περιορισμοί Permission Based μεθόδου: Είναι ένα γρήγορο φίλτρο για την εφαρμογή σάρωσης και τον εντοπισμό ότι η εφαρμογή είναι καλοήθης ή κακόβουλο λογισμικό, αλλά αναλύει μόνο το manifest αρχείο, δεν αναλύει άλλα αρχεία που μπορεί να περιέχουν κακόβουλο κώδικα. Επίσης υπάρχει πολύ μικρή διαφορά στα δικαιώματα που χρησιμοποιούνται από τις καλοήθεις ή κακοήθεις εφαρμογές. Οι Permission Based μέθοδοι απαιτούν δεύτερο πέρασμα για την παροχή αποτελεσματικής ανίχνευσης κακόβουλου λογισμικού.

C. *Dalvik Bytecode Analysis,* στο Android, Dalvik είναι μία βάση μητρώου (register-based) VM. Οι Android εφαρμογές αναπτύχθηκαν σε γλώσσα Java Bytecode και στη συνέχεια μετατρέπεται σε Dalvik Bytecode. Η Bytecode Analysis μας βοηθά στην ανάλυση της συμπεριφοράς της εφαρμογής. Το SCANDAL [36] είναι ένας στατικός αναλυτής που αναλύει το Dalvik Bytecode των εφαρμογών και ανιχνεύει τη διαρροή προστασίας της ιδιωτικής ζωής σε εφαρμογές. Καθορίζει τη ροή δεδομένων από τη

πηγή πληροφοριών σε οποιοδήποτε απομακρυσμένο διακομιστή. Οι Karlsen και συνεργάτες [66] παρουσιάζουν την πρώτη επισημοποίηση του Dalvik Bytecode με Java χαρακτηριστικά. Εξέτασαν 1700 δημοφιλείς εφαρμογές Android για να καθορίσουν τί Dalvik Bytecode χαρακτηριστικά και εντολές χρησιμοποιούνται επί το πλείστων στις εφαρμογές Android. Βοηθά στην ανάλυση ελέγχου και ανάλυση ροής δεδομένων για την ανίχνευση κακόβουλων εφαρμογών ή για τον εντοπισμό ευαίσθητων API calls που επικαλούνται κατά τη διάρκεια της εκτέλεσης. Υποστηρίζει δυναμική επιδιόρθωση και ανακλαστικά χαρακτηριστικά. Το DroidMoss [77]εξάγει την Dalvik Bytecode ακολουθία και ο προγραμματιστής πληροφοριών με χρήση του εργαλείου baksmali [99] και δημιουργεί fingerprint (αποτύπωμα) για κάθε εφαρμογή με τη χρήση fuzzy hashing τεχνικών για να δημιουργηθεί σταθερού μεγέθους υπογραφή για τον εντοπισμό ανασκευασμένων εφαρμογών. Με βάση τον βαθμό ομοιότητας προσδιορίζονται οι ανασκευασμένες εφαρμογές. Η προτεινόμενη λύση δεν μπορεί να ανιχνεύσει τις ανασκευασμένες εφαρμογές των οποίων η αυθεντική δεν παρουσιάζεται στη βάση δεδομένων, έτσι πολλά κακόβουλα λογισμικά παραμένουν μη ανιχνεύσιμα. Το DroidAPIMiner [01] αναπτύχθηκε βάσει του Androguard [18], όπου αναγνωρίζει το κακόβουλο λογισμικό με την παρακολούθηση των API calls, επικίνδυνων παραμέτρων επίκλησης και package level πληροφοριών με Bytecode. Για την ταξινόμηση των εφαρμογών σαν καλοήθεις ή κακοήθεις εφαρμόζει τον KNN αλγόριθμο [18] και ανιχνεύεται με 99% ακρίβεια και 2.2% false positive. Ο Fuchs και συνεργάτες [28] παρουσίασαν το SCandroid που αναλύει την Android εφαρμογή στατικά όπως έχει εγκατασταθεί και εκτελεί ανάλυση ροή δεδομένων για να ελέγξει αν είναι συνεπής ή όχι. Βάσει της ροής δεδομένων δηλώνει αν η εφαρμογή είναι ασφαλής για να τρέξει με τα αιτούμενα δικαιώματα. Dexpler tool [04] μετατρέπει τον Dalvik Bytecode σε Jimple code που χρησιμοποιείται σε static analysis framework name Soot [50]. Κάνει το Soot να διαβάζει απευθείας τον Dalvik Bytecode σε Java Bytecode. Το ComDroid [13] είναι ένα εργαλείο που ανιχνεύει τα τρωτά σημεία επικοινωνίας ανάμεσα σε εφαρμογές Android. Ανέλυσαν 20 δείγματα και 34 εκμεταλλεύσιμα τρωτά σημεία μεταξύ 12 εφαρμογών. Χρησιμοποιεί το Dedxer εργαλείο [45] για να αποσυναρμολογήσει τα αρχεία dex στην εφαρμογή. Εκτελεί στατική ανάλυση σε Dalvik αρχεία, αναλύει τη λίστα δικαιωμάτων στο αρχείο manifest.xml της εφαρμογής.

Περιορισμοί Dalvik Bytecode μεθόδου: Σε αυτή τη μέθοδο ανάλυσης εκτελείται σε επίπεδο εντολών και καταναλώνει περισσότερο χώρο ενέργειας και αποθήκευσης. Οι συσκευές Android είναι φτωχοί πόροι και έτσι περιορίζεται η προσέγγιση ανίχνευσης

2.8.2 Δυναμική ανάλυση

Η Δυναμική ανάλυση εξετάζει την εφαρμογή κατά την διάρκεια της εκτέλεσης της. Μπορεί να χάσουν κάποια κομμάτια κώδικα που δεν εκτελούνται αλλά μπορούν εύκολα να εντοπίσουν κακόβουλες συμπεριφορές που δεν ανιχνεύονται με μεθόδους στατικής ανάλυσης. Αν και οι στατικές μέθοδοι ανάλυσης είναι πιο γρήγορες στην ανίχνευση κακόβουλου λογισμικού και στα κρυπτογραφημένα κακόβουλα λογισμικά. Ο Egele [20] παρέχει λεπτομερή επισκόπηση των διάφορων δυναμικών μεθόδων ανάλυσης που χρησιμοποιούνται για διάκριση ανάμεσα σε καλοήθεις και κακοήθεις εφαρμογές. Η μέθοδος δυναμικής ανάλυσης είναι αποτελεσματική απέναντι στις πολυμορφικές και μεταμορφικές τεχνικές κώδικα συσκότισης από κακόβουλα λογισμικά [73] αλλά απαιτεί περισσότερους πόρους. Εξαιτίας της μεγαλύτερης αποτελεσματικότητας της στη μελέτη αυτή θα χρησιμοποιηθεί η Δυναμική ανάλυση.

1) *Anomaly Based Detection.* Το Crow Droid [08] προτάθηκε για να ανιχνεύσει τη συμπεριφορά των εφαρμογών δυναμικά. Οι λεπτομέρειες του συστήματος κλήσεων που επικαλείται από την εφαρμογή συλλέγονται με τη βοήθεια του εργαλείου Strace [100] και στη συνέχει η εφαρμογή crowdsourcing, η οποία εγκαθίσταται στη συσκευή και δημιουργεί ένα αρχείο καταγραφής και το στέλνει σε απομακρυσμένο διακομιστή. Το αρχείο καταγραφής μπορεί να περιγράφει τις ακόλουθες πληροφορίες: Device information, apps installed on device και system calls. Από την πλευρά του διακομιστή για να ταξινομηθεί η εφαρμογή σαν κακόβουλη ή καλόβουλη εφαρμόζεται 2-mean clustering αλγόριθμος. Τα αποτελέσματα αποθηκεύονται στη βάση δεδομένων του διακομιστή. Αυτή η λύση παρέχει βαθιά ανάλυση και έτσι απαιτείται μεγάλος αριθμός πόρων. Επιπλέον απαιτεί εφαρμογή πελάτη (client app) για να εγκατασταθεί στη συσκευή του χρήστη και ίσως ταξινομεί τη νόμιμη εφαρμογή σαν κακόβουλη στην περίπτωση που επικαλείται περισσότερες κλήσεις συστήματος. Το Andromly [57] είναι μία συμπεριφορά βασισμένη σε Android malware detection system. Προκειμένου να ταξινομηθεί η εφαρμογή σαν καλοήθης ή κακοήθης παρακολουθούνται συνεχώς διάφορα χαρακτηριστικά και μοτίβα που δείχνουν την κατάσταση της συσκευής όπως το επίπεδο της μπαταρίας και η κατανάλωση της CPU. Ενώ βρίσκεται σε λειτουργία, στη συνέχεια εφαρμόζει αλγόριθμο μηχανικής μάθησης για τη διάκριση των εφαρμογών σε καλοήθεις ή κακοήθεις. Αυτή η λύση ανιχνεύει συνεχείς επιθέσεις και μπορεί να ειδοποιήσει το χρήστη σχετικά με αυτές τις επιθέσεις. Το AntiMalDroid [75] είναι ένα πλαίσιο ανίχνευσης κακόβουλου λογισμικού χρησιμοποιώντας SVM αλγόριθμο που προτάθηκε για να προσδιορίσει τις κακόβουλες εφαρμογές και τις παραλλαγές

τους κατά τη διάρκεια της εκτέλεσης τους. Αρχικά παρακολουθεί τη συμπεριφορά των εφαρμογών και τα χαρακτηριστικά τους ώστε να τις κατηγοριοποιήσει. Στη συνέχεια θέτει δύο τύπους χαρακτηριστικών σε learning module. Έπειτα αποθηκεύει την υπογραφή σε βάση δεδομένων και τη συγκρίνει με τις υπογραφές των υπάρχων κακόβουλων ή κακόβουλων εφαρμογών. Έτσι παρέχεται υψηλό ποσοστό ανίχνευσης αλλά καταναλώνει αρκετό χρόνο.

2) *Taint Analysis.* Ο Enck και συνεργάτες [21] πρότειναν το Traint Droid που παρέχει ένα ευρύ σύστημα παρακολούθησης ροής πληροφοριών για Android. Μπορεί ταυτόχρονα να παρακολουθεί πολλαπλές πηγές ευαίσθητων δεδομένων όπως η κάμερα, το μικρόφωνο και το GPS και εντοπίζει διαρροή δεδομένων σε third party προγραμματιστές εφαρμογών. Κατονομάζει τα ευαίσθητα δεδομένα , παρακολουθεί αυτά τα δεδομένα και τις εφαρμογές όταν αλλοιωμένα δεδομένα αφήνονται να μετακινούνται από τη συσκευή. Παρέχει αποτελεσματική παρακολούθηση ευαίσθητων πληροφοριών αλλά δεν εκτελούν παρακολούθηση ροής ελέγχου. Επίσης δεν μπορεί να παρακολουθεί πληροφορίες, τις αφήνει στον πάγο και επιστρέφει σε απάντηση του δικτύου.

3) *Emulation Based Detection. To* DroidScope [70] είναι μία πλατφόρμα δυναμικής ανάλυσης Android βασισμένη σε Virtual Machine. Σαν antimalware ανιχνεύει την παρουσία κακόβουλων λογισμικών επειδή και τα δύο παραμένουν στο ίδιο περιβάλλον εκτέλεσης, έτσι τα κακόβουλα λογισμικά μπορούν να ανιχνεύσουν την παρουσία antimalware. Το DroidScope παρακολουθεί ολόκληρο το λειτουργικό σύστημα από την παραμονή τους έξω από το περιβάλλον και έτσι έχουν περισσότερα προνόμια από τα κακόβουλα προγράμματα. Παρακολουθεί επίσης το Dalvik, έτσι υπάρχει κλιμάκωση των επιθέσεων προνομίων Kernel και έτσι μπορεί να ανιχνευτεί. Το DroidDream και το DroidKungFu [68] ανιχνεύθηκαν με αυτή την τεχνική.

Ο Blaising και συνεργάτες [05] πρότειναν το Android Application Sandbox (AASandbox) , το οποίο ανιχνεύει ύποπτες εφαρμογές με εκτέλεση τόσο στατικής και δυναμικής ανάλυσης. Εξάγει το .dex αρχείο σε αναγνωρίσιμη μορφή και στη συνέχεια εκτελεί στατική ανάλυση της εφαρμογής. Στη συνέχεια αναλύει χαμηλού επιπέδου αλληλεπιδράσεις με το σύστημα από την εκτέλεση της εφαρμογής σε απομακρυσμένο περιβάλλον Sandbox. Οι δράσεις της εφαρμογής περιορίζονται στο Sandbox λόγω της πολιτικής ασφάλειας και δεν επηρεάζει τα δεδομένα της συσκευής. Χρησιμοποιεί το Money tool για να αναλύσει τη συμπεριφορά της εφαρμογής

δυναμικά που δημιουργεί τυχαία γεγονότα χρήστη όπως αγγίγματα, κλίκς και χειρονομίες, δεν μπορεί να ανιχνεύσει νέους τύπους κακόβουλου λογισμικού

Όταν ανιχνεύεται Android malware, είτε με static detection method ή με Dynamic method, το πρώτο βήμα είναι η απόκτηση του application 's way of behaviours συμπεριλαμβανομένων των κανονικών εφαρμογών και του κακόβουλου λογισμικού, τότε χρησιμοποιεί machine learning για να αποκτήσει το χαρακτηριστικό του κακόβουλου λογισμικού για να διακρίνει τις κακόβουλες εφαρμογές από τις κανονικές. Εστιάζει στην ανάλυση της λειτουργίας κλήσεων των Android applications, η έκθεση αναλύει την λειτουργία κλήσεων του κακόβουλου λογισμικού για να αποκτήσει τυπικό file χαρακτηριστικό του κακόβουλου λογισμικού, το οποίο χρησιμοποιείται σαν βάση της ανίχνευσης [74].

Προκειμένου να καθοριστεί αν μια εφαρμογή είναι κακόβουλη ή όχι, πρέπει να αναλυθεί με μεγάλη προσπάθεια. Τα χαρακτηριστικά του, καθώς και το εύρος λειτουργίας της πρέπει να τεκμηριώνεται. Τα αποτελέσματα της στατικής ανάλυσης χρησιμοποιούνται για να καθοδηγήσουν την ακόλουθη δυναμική ανάλυση που περιγράφεται. Η δυναμική ανάλυση εκτελεί αυτόματα τις εφαρμογές σε ένα τροποποιημένο σύστημα Android, με τη βοήθεια του εξομοιωτή Android.

2.8.3 Ανάλυση σε εικονικό περιβάλλον

Στην ασφάλεια υπολογιστών, το Sandbox είναι ένας μηχανισμός ασφάλειας για το διαχωρισμό των τρέχων προγραμμάτων. Χρησιμοποιούνται συχνά για την εκτέλεση αδοκίμαστων και αναξιόπιστων προγραμμάτων ή κώδικα, που προέρχονται από πιθανά ανεπιβεβαίωτα ή μη αξιόπιστα third parties, suppliers, users ή websites, χωρίς να υπάρχει κίνδυνος βλάβης στο μηχάνημα υποδοχής ή το λειτουργικό σύστημα [31].

Το Sandbox παρέχει τυπικά ένα αυστηρό ελεγχόμενο σύνολο πόρων για τα προγράμματα επισκεπτών για να τρέξουν μέσα, όπως ένας άδειος δίσκος και μνήμη. Δίνει πρόσβαση στο δίκτυο, η ικανότητα να επιθεωρήσει το σύστημα του κεντρικού υπολογιστή ή να διαβάζονται συσκευές εισόδου που συνήθως έχουν αποκλειστεί ή είναι αυστηρά περιορισμένες. Με την έννοια της παροχής ενός υψηλού ελεγχόμενου περιβάλλοντος, sandboxes μπορεί να θεωρηθεί ως ένα παράδειγμα εικονοποίησης (virtualization). Το Sandboxing χρησιμοποιείται για να δοκιμάσει ανεπιβεβαίωτα προγράμματα που μπορεί να περιέχουν ένα ιό ή άλλο κακόβουλο κώδικα χωρίς να επιτρέπει το λογισμικό να βλάψει τη συσκευή υποδοχής [30].

Παράδειγμα εφαρμογής του Sandbox είναι τα Virtual Machines, τα μιμούνται πλήρως ένα ενός host Computer στο οποίο ένα συμβατικό λειτουργικό σύστημα μπορεί να εκκινήσει και να τρέξει σαν ένα πραγματικό hardware. Το φιλοξενούμενο λειτουργικό σύστημα τρέχει σε sandboxed υπό την έννοια ότι δεν λειτουργεί εγγενώς στο host και μπορεί να έχει πρόσβαση σε host πόρους μέσω εξομοιωτή (emulator). Επιπλέον Sandboxing on Native host, όπου οι ερευνητές ασφάλειας βασίζονται σε μεγάλο βαθμό στις τεχνολογίες Sandboxing για την ανάλυση κακόβουλων συμπεριφορών. Με τη δημιουργία ενός περιβάλλοντος που μιμείται ή αναπαράγει στοχευμένους επιτραπέζιους υπολογιστές, οι ερευνητές μπορεί να αξιολογήσουν πόσο το κακόβουλο λογισμικό μολύνει και θέτει σε κίνδυνο τους hosts που αποτελούν στόχο. Πολυάριθμες υπηρεσίες ανάλυσης κακόβουλου λογισμικού βασίζονται σε τεχνολογίες Sandboxing [72].

Συνολικά, υπάρχουν πολύ λίγα συστήματα ανάλυσης που συνδυάζουν στατική και δυναμική ανάλυση και κανένας που να παρακολουθεί δυναμικά τις δύο δράσεις στο πλαίσιο του Dalvik VM και έξω από αυτό, στις μητρικές βιβλιοθήκες [101]. Εντός του στατικού μέρους ανάλυσης μπορούμε να αναλύσουμε την εφαρμογή με διάφορες ενότητες για να πάρουμε μια γενική εικόνα της εφαρμογής. Πρώτον, μπορούμε να εκτελέσουμε πολλές σαρώσεις antivirus που χρησιμοποιούν την υπηρεσία VirusTotal [32], δεύτερον, ανάλυση του manifest file, και, τέλος, αποκωδικοποίηση της εφαρμογής για τον καλύτερο εντοπισμό ύποπτου κώδικα. Με τη δυναμική ανάλυση, εκτελούμε την εφαρμογή σε ένα εξομοιωτή και καταγράφουνε κάθε λειτουργία της εφαρμογής, δηλαδή, καταγράφουμε τόσο τις ενέργειες που εκτελούνται στο Java Virtual Machines π.χ. το Dalvik [32]και δράσεις που εκτελούνται στις μητρικές βιβλιοθήκες μπορεί να συνδυαστούν με την εφαρμογή. Για την καλύτερη γνώσης μας, το Mobile Sandbox είναι το πρώτο πλαίσιο ανάλυσης για την πλατφόρμα Android που έχει αυτήν τη δυνατότητα.

Για την αξιολόγηση του συστήματος [32],συλλέχθηκαν πάνω από 136.000 ελεύθερες διαθέσιμες εφαρμογές από τις πιο σημαντικές Ασιατικές αγορές και το Google Play-Market. Συλλέχθηκαν επίσης περίπου 7.500 δείγματα κακόβουλου λογισμικού από διάφορες οικογένειες malware. Στη συνέχεια χρησιμοποιήθηκε το Mobile-Sandbox για να αναλύσει αυτόματα 40.000 τυχαία επιλεγμένες εφαρμογές και από τα δύο σύνολα δειγμάτων. Μέσα σε αυτά τα 40.000 δείγματα το σύστημά μας ανίχνευσε 4.641 κακόβουλες εφαρμογές και επιπλέον 5 ύποπτα δείγματα τα οποία προσπαθούν να κρύψουν κακόβουλες ενέργειες τους μέσα σε εγγενή κώδικα (Native code). Αυτή η εικόνα δείχνει σαφώς ότι τα σημερινά συστήματα ανάλυσης αποτελούν σημαντικές πιθανές απειλές.

Το Mobile-Sandbox, είναι ένας στατικός και δυναμικός αναλυτής για Android εφαρμογές με σκοπό να υποστηρίξει malware αναλυτές για να ανιχνεύσουν κακόβουλη συμπεριφορά. Στην στατική ανάλυση θα αναλύσει το Manifest file της εφαρμογής και θα κάνει (decompile) ανακατασκευή της αίτησης. Σε ένα περαιτέρω βήμα θα καθορίσει αν η εφαρμογή χρησιμοποιεί αναζήτηση ύποπτων δικαιωμάτων ή προθέσεων. Το δεύτερο μέρος του Sandbox μας εκτελεί την δυναμική ανάλυση όπου θα εκτελέσει την εφαρμογή, προκειμένου να καταγράφει όλες τις ενέργειες που εκτελούνται συμπεριλαμβανομένων και εκείνων που απορρέουν από τις Native API calls.

Υπάρχουν ακόμη πολλά σημεία προς βελτίωση του Mobile-Sandbox, ιδίως όσον αφορά την απόδοση. Τα Mobile Sandboxes μπορούν να χρησιμοποιηθούν για την ανίχνευση κακόβουλου λογισμικού δυναμικά. Τα sandboxes μπορούν να τρέξουν κώδικες και αυτοματοποιημένα εργαλεία και στη συνέχεια να χρησιμοποιηθούν για να εντοπίσουν τι κάνει η εφαρμογή, ωστόσο έχουν αποδειχθεί ότι είναι ευάλωτα σε ορισμένους τύπους κακόβουλου λογισμικού. Οι Maier και συνεργάτες [43] βρήκε ότι οι επιθέσεις Divide and Conquer χρησιμοποιούν ένα συνδυασμό από Dynamic code loading και fingerprinting για την αποφυγή σαρωτών κακόβουλου λογισμικού.

Προκειμένου να βελτιωθούν τα sandboxes, προτείνεται η χρήση αλγορίθμων μηχανικής μάθησης και έτσι μπορεί να αναπτυχθούν από ένα ζεύγος δείγματος που αναλύθηκαν με το χέρι [79]

Σημαντικό πρόβλημα αποτελεί η ανίχνευση κακόβουλου λογισμικού σε κινητές συσκευές. Υπάρχουν αρκετές προσεγγίσεις [19,35] όπως η παρακολούθηση της ενεργειακής χρήσης των εφαρμογών και αναφορά ανώμαλης κατανάλωσης. Άλλοι παρακολουθούν [08,69] τις κλήσεις του συστήματος και προσπαθούν να ανιχνεύσουν ασυνήθιστα μοτίβα κλήσεων συστήματος. Άλλες προσεγγίσεις χρησιμοποιούν περισσότερο παραδοσιακή σύγκριση με γνωστά κακόβουλα λογισμικά [06] ή άλλες ερευνητικές μεθόδους. Παραδοσιακά η προσέγγιση στατικής ανάλυσης [14,40], όπου μπορεί να επικεντρώνονται στη σύγκριση προγραμμάτων για να γνωρίσει το κακόβουλο λογισμικό βασισμένο σε κώδικα προγραμματισμού, που ψάχνει για υπογραφές ή άλλες μεθόδους. Άλλες προσεγγίσεις [38,54,62] εστιάζουν στη χρήση προσεγγίσεων αλγορίθμου μηχανικής μάθησης (machine learning algorithm) και εξόρυξης δεδομένων (Data Mining) για την ανίχνευση κακόβουλου λογισμικού. Οι Schultz και συνεργάτες [54] σύγκριναν τρείς αλγόριθμους μηχανικής μάθησης που εκπαιδεύτηκαν σε τρία χαρακτηριστικά: DLL και system calls που γίνονται από πρόγραμμα, string που βρέθηκαν σε δυαδικό πρόγραμμα (program binary) και raw hexadecimal δυαδική αναπαράσταση. Οι Kotler και Maloof [38] εκπαίδευσαν πολλούς αλγόριθμους μηχανικής μάθησης σε byte string n-grams.

Κεφάλαιο 3
Σχεδιασμός

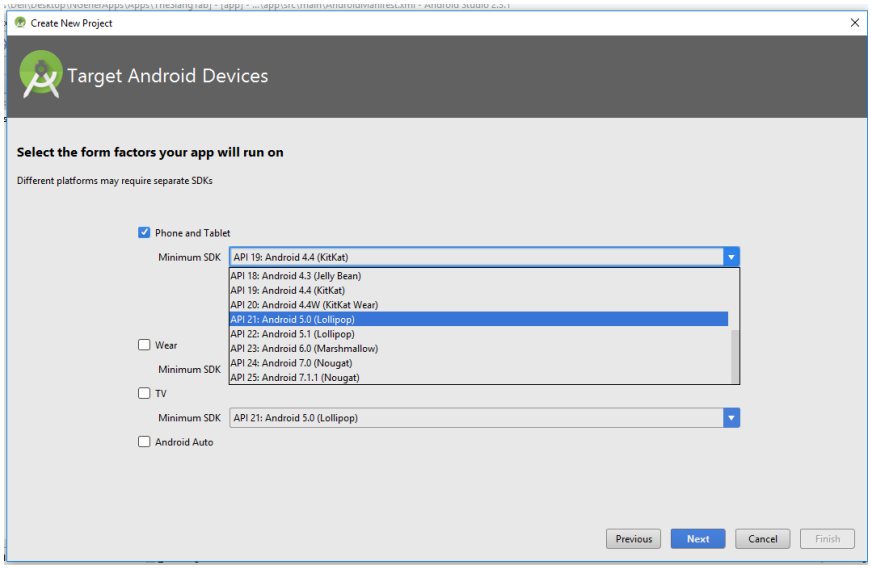
3.1 Απαιτήσεις σε λογισμικό και υλικό

Η επιτυχής εκτέλεση της πειραματική διαδικασίας απαιτεί συγκεκριμένο υλικό και λογισμικό όπως παρουσιάζονται στην συνέχεια. Για την διαδικασία σύγκρισης των τριών εκδόσεων και των 50 εφαρμογών χρησιμοποιήθηκαν συγκεκριμένο λογισμικό και Android συσκευές.

3.1.1 Λογισμικό

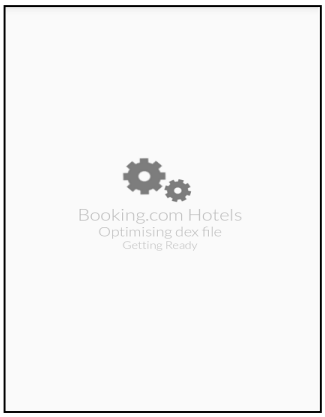
Αρχικά χρησιμοποιήθηκε η γλώσσα προγραμματισμού Android Studio, το κριτήριο επιλογής ήταν πως μπορεί ένας προγραμματιστής να τρέξει τον κώδικα μιας εφαρμογής σε οποιαδήποτε έκδοση Android. Για την εκκίνηση της διαδικασίας εγκατέστησα στον υπολογιστή το Android Studio. Στην συγκεκριμένη γλώσσα προγραμματισμού χρησιμοποιούνται για τη δημιουργία της εφαρμογής οι γλώσσες προγραμματισμού Java, Manifest.xml και οι activities (xml). Η χρήση Java βοηθάει τον προγραμματιστή να γράψει τον κώδικα που θα εκτελείται όταν τρέχει η εφαρμογή.

Η χρήση των activities (xml) επιτρέπει στον προγραμματιστή να πραγματοποιήσει τον σχεδιασμό της εφαρμογής. Τα activities (xml) ενώνονται προγραμματιστικά με τον κώδικα Java. Το Manifest.xml είναι υπεύθυνο για να ζητάει άδεια από τις συσκευές ώστε να χρησιμοποιήσει για παράδειγμα το διαδίκτυο, τις επαφές κλπ. Επίσης ελέγχει την ιεραρχία των activities (xml).



Εικόνα 3.1: Android Studio.

Ακολούθως έγινε χρήση του Show Java, είναι μια εφαρμογή που εμφανίζει τον κώδικα μιας εφαρμογής όπως το Manifest.xml, activities(xml) και της Java. Η συγκεκριμένη εφαρμογή τρέχει σε συσκευές Android και μπορεί να την κατεβάσουν όλοι οι χρήστες από το Play Store. Με την εν λόγω εφαρμογή μπορούμε να βρούμε τον κώδικα των εφαρμογών και να τον τρέξουμε στο Android Studio στις εκδόσεις KITKAT, Lollipop και Marshmallow. Συνήθως παίρνει περίπου 5-10 λεπτά μέχρι να εμφανισθεί ο κώδικας. Πιο κάτω η εικόνα δείχνει πως λειτουργεί η εφαρμογή.



Εικόνα 3.2: Εφαρμογή Show Java.

Για τη ανάλυση του Network Traffic που γίνεται κατά το χρόνο εκτέλεσης των εφαρμογών έγινε με τη χρήση του Wireshark. Το Wireshark είναι ένα λογισμικό ανοικτού και ελεύθερου κώδικα που μας βοηθά στην ανάλυση των πρωτοκόλλων δικτύου. Αποτελεί ένα ιδιαιτέρως χρήσιμο εργαλείο αφού βοηθά στον εντοπισμό και την αντιμετώπιση προβλημάτων που μπορεί μα υπάρξουν σε ένα δίκτυο.

Οι εφαρμογές που αναλύθηκαν είναι 360 Security – Antivirus Free, Run Cow Run, Solar System Scope, Mean Spheres Attack, Fillshape, Piques, Lunchbox, Calorie Counter, 3D Charts, Root Browser, Enemy Strike, Audio Manager, Vector, Spy Mouse, Link2SD, 4Shared, John NES - NES Emulator, Clean Master – Free Antivirus ,3C Toolbox, 9GAG, Root Tool Case, Mobile Security & Antivirus, Unit Converter, Dual Sim Selector και Smart IPTV προέρχονταν από το Play store. Για τη μελέτη και ανάλυση των κρακαρισμένων εφαρμογών από τη σελίδα http://www.crackapk.com εγκαταστάθηκαν οι εφαρμογές: 360 Security – Antivirus Free, Run Cow Run, Solar System Scope, Mean Spheres Attack, Fillshape, Piques, Lunchbox, Calorie Counter, 3D Charts, Root Browser, Enemy Strike, Audio Manager, Vector, Spy Mouse, Link2SD, 4Shared, John NES - NES Emulator, Clean Master – Free Antivirus και από τη σελίδα http://www.appcake.net/ οι εφαρμογές: 3C Toolbox, 9GAG, Root Tool Case, Mobile Security & Antivirus, Unit Converter, Dual Sim Selector και Smart IPTV.

3.1.2 Υλικό

Για την πραγματοποίησης της συγκεκριμένης μεταπτυχιακής διατριβής απαιτούνται και υλικά. Χρησιμοποιήθηκε Ηλεκτρονικός υπολογιστής, για την εγκατάσταση του Android Studio., ο οποίος είχε τα εξής χαρακτηριστικά: Processor: Intel(R) Core(TM) i3-2370M CPU @ 2.40 GHz, Installed Memory (RAM): 4,00 GB, System type: 64-bit Operating System, x64-based processor.

Για τη συγκριτική μελέτη της χρήσης CPU, RAM αλλά και του Network Traffic στις τρείς εκδόσεις Android από τις εφαρμογές χρησιμοποιήθηκαν τρία κινητά τηλέφωνα Android. Το Samsung Galaxy S3 neo χρησιμοποιήθηκε για την μέτρηση της RAM και της CPU των 25 εφαρμογών που προέρχονται από το Play Store, των 25 εφαρμογών που είναι κρακαρισμένες (από Third Party Markets) αλλά και για την εγκατάσταση της εφαρμογής Show Java, αντιπροσωπεύει το λογισμικό σύστημα 4.4.2 KIT KAT. Το Samsung Galaxy S5 χρησιμοποιήθηκε για την μέτρηση της RAM και της CPU των 25 εφαρμογών που προέρχονται από το Play Store, των 25 εφαρμογών που είναι κρακαρισμένες (από Third Party Markets) αλλά και για την εγκατάσταση της εφαρμογής Show Java, αντιπροσωπεύει το λογισμικό σύστημα 5.0 Lollipop. Και το Samsung

Galaxy S7 χρησιμοποιήθηκε για την μέτρηση της RAM και της CPU των 25 εφαρμογών που προέρχονται από το Play Store, των 25 εφαρμογών που είναι κρακαρισμένες (από Third Party Markets) αλλά και για την εγκατάσταση της εφαρμογής Show Java, αντιπροσωπεύει το λογισμικό σύστημα 6.0 Marshmallow.

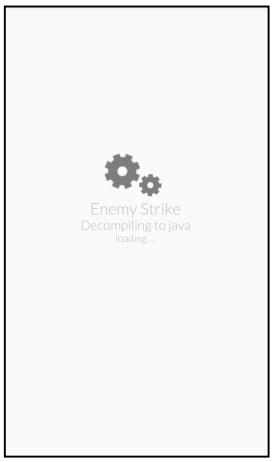
3.2 Πειραματική Διαδικασία

Οι χρήστες καθημερινά κάνουν εγκατάσταση καινούργιες εφαρμογές. Πολλές φορές οι εφαρμογές αυτές είναι καλοήθεις. Δηλαδή έχουν σκοπό να κάνουν τη ζωή του χρήστη πιο εύκολη, να πουλήσουν προϊόντα και να τον ψυχαγωγήσουν. Μερικές φορές αυτές οι εφαρμογές είναι κακόβουλες. Δηλαδή έχουν σκοπό να υποκλέψουν στοιχεία από τον χρήστη, να στείλουν προσβλητικά μηνύματα ή ακόμα να καταστρέψουν τη συσκευή. Αυτό συμβαίνει είναι γιατί η Google επιτρέπει το ανέβασμα της οποιασδήποτε εφαρμογής στο Play Store. Αυτό διορθώνεται από τους προγραμματιστές της Google γιατί μελετούν και παρακολουθούν τις εφαρμογές και αν κάποια εφαρμογή είναι κακόβουλη αφαιρείται από το Play Store.

Με μια διαδικασία που εξηγείται αναλυτικά πιο κάτω θα αναλύσω 25 εφαρμογές από το Play Store και 25 εφαρμογές από Third Party Android Markets που είναι το Crackapk [103] και το Appcake [103], μελετώντας τον κώδικα τους, τη χρήση που κάνουν σε RAM και CPU αλλά και το Network Traffic που πραγματοποιείται.

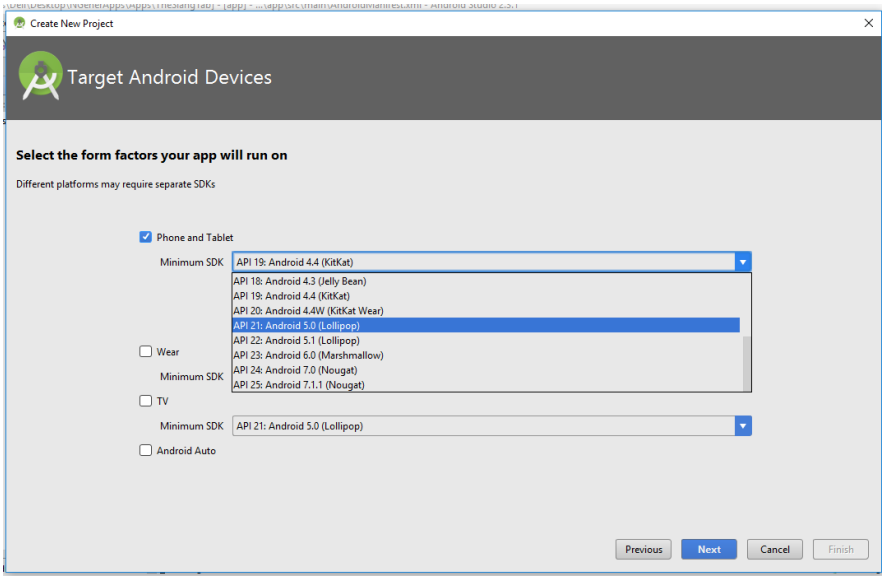
1. Για την μελέτη των δικαιωμάτων που αιτούνται οι εφαρμογές από τις συσκευές Android είναι η ακόλουθη:

I. Αρχικά έγινε χρήση της συσκευής Samsung Galaxy S3 neo, και του προγράμματος Show Java για να εμφανισθούν οι κώδικες της κάθε εφαρμογής που προέρχεται από το Play Store.



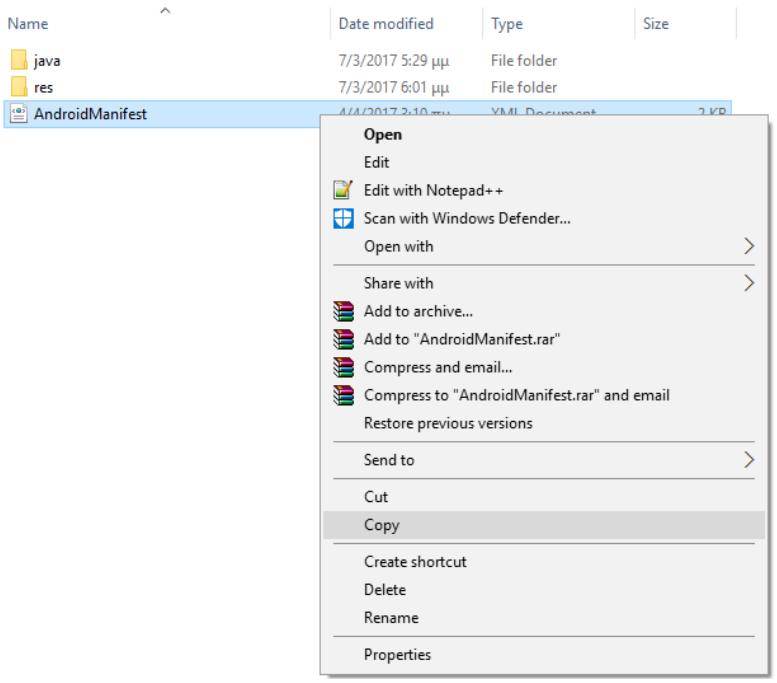
Εικόνα 3.3: Χρήση του Show Java για την εμφάνιση του κώδικα της εφαρμογής Enemy Strike.

II. Δημιουργήσαμε ένα καινούργιο Android Project επιλέγοντας κάθε φορά την έκδοση λογισμικού. Για κάθε εφαρμογή χρησιμοποιήθηκαν οι εκδόσεις λογισμικού KITKAT, Lollipop και Marshmallow.



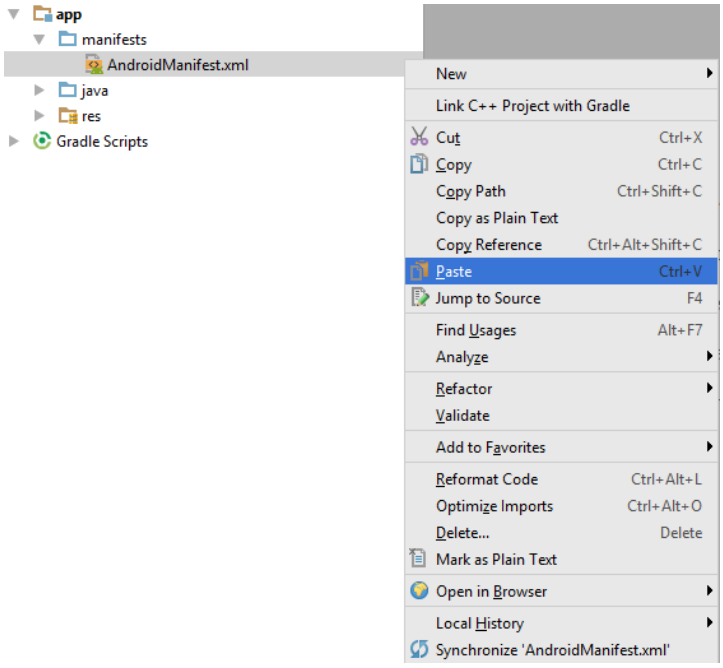
Εικόνα 3.4: Δημιουργία καινούργιου Android Project.

III. Αντιγράφοντας το αρχείο με την ονομασία Manifest από τον φάκελο της εφαρμογής στον καινούριο Manifest.



Εικόνα 3.5: Αντιγραφή του αρχείου Manifest από τον φάκελο της εφαρμογής στον καινούργιο Manifest.

IV. Στη συνέχεια έγινε επικόλληση στον καινούριο φάκελο Manifest.



Εικόνα 3.6:Επικόλληση στον καινούργιο φάκελο Manifest.

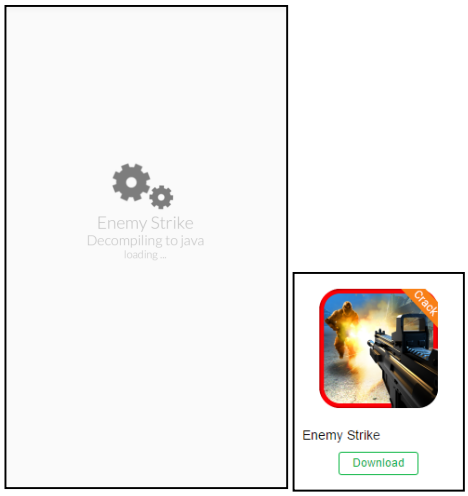
V. Τελικά, βλέπουμε όλα τα δικαιώματα (Permissions) στο Manifest.xml και τα αποτελέσματα από τις δοκιμές.

```
<uses-permission android: name="android.permission.INTERNET" />
<uses-permission
android:name="android.permission.MEDIA_CONTENT_CONTROL"/>
<uses-permission
android:name="android.permission.READ_CONTACTS"/>
```

Εικόνα 3.7:Η εφαρμογή Enemy Strike ζητά άδεια από τη συσκευή για να χρησιμοποιήσει το διαδίκτυο, να γνωρίζει ποιο περιεχόμενο αναπαράγεται και να ελέγχει την αναπαραγωγή του και να διαβάζει τις επαφές.

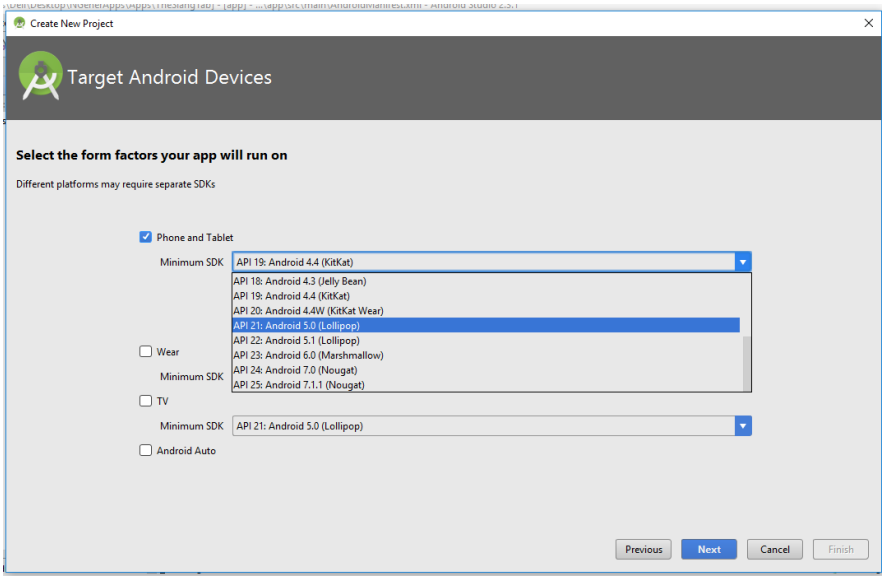
2. Πιο κάτω ακολουθεί η περιγραφή της διαδικασίας ανάλυσης εφαρμογών που προέρχονται από Third Party Android Markets και είναι κρακαρισμένες.

I. Αρχικά κατέβασα τις εφαρμογές από τις ιστοσελίδες http://www.crackapk.com/, http://www.appcake.net/. Χρησιμοποίησα στη συσκευή Samsung Galaxy S3 neo και το πρόγραμμα Show Java για την εμφάνιση των κωδίκων της κάθε εφαρμογής.



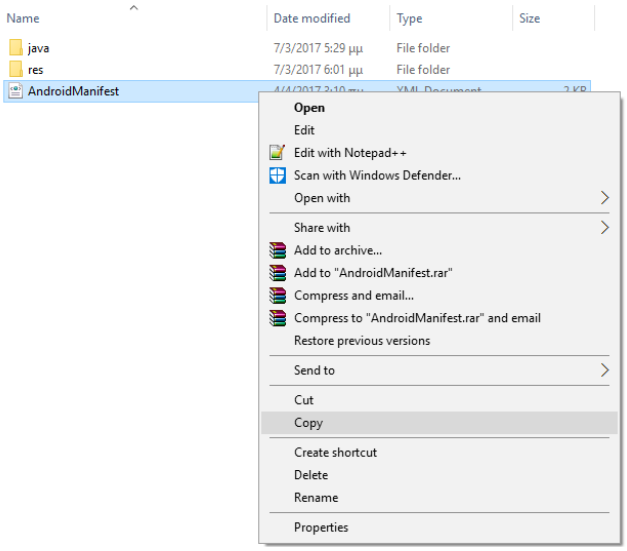
Εικόνα 3.8: Η εφαρμογή Enemy Strike και το Show Java.

II. Στη συνέχεια δημιούργησα καινούργιο Android Project επιλέγοντας κάθε φορά την έκδοση λογισμικού. Για κάθε εφαρμογή χρησιμοποιήθηκαν οι εκδόσεις λογισμικού KITKAT, Lollipop και Marshmallow.



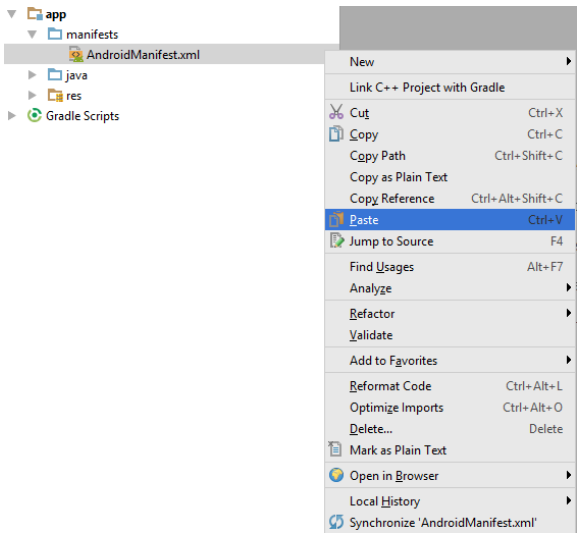
Εικόνα 3.9: Δημιουργία καινούργιου Android Project.

III. Αντέγραψα το αρχείο με την ονομασία Manifest από τον φάκελο της εφαρμογής στον καινούριο Manifest.



Εικόνα 3.10: Αντιγραφή Manifest.

IV. Ακολούθως γίνεται επικόλληση στον καινούργιο φάκελο Manifest.



Εικόνα 3.11: Επικόλληση στο Manifest.

V. Έτσι βλέπουμε όλα τα δικαιώματα (permissions) στο Manifest.xml και τα αποτελέσματα από τις δοκιμές.

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission
android:name="android.permission.MEDIA_CONTENT_CONTROL"/>
<uses-permission
android:name="android.permission.READ_CONTACTS"/>
<uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
```

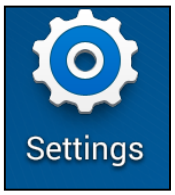
Εικόνα 3.12: Η εφαρμογή Enemy Strike (Cracked) ζητά άδεια από τη συσκευή για να χρησιμοποιήσει το διαδίκτυο, να γνωρίζει ποιο περιεχόμενο αναπαράγεται και να ελέγχει την αναπαραγωγή του, να διαβάζει τις επαφές, να διαβάζει και να γράφει δημόσια δεδομένα στο κοινόχρηστο εξωτερικό αποθηκευτικό χώρο.

3. Για τη διαδικασία της μέτρησης της CPU και της RAM που χρησιμοποιούν οι εφαρμογές που προέρχονται από το Google Play Store και από τα Third Party Markets, στις τρείς υπό μελέτη εκδόσεις Android KITKAT, Lollipop και Marshmallow είναι η ακόλουθη:

I. Αρχικά τρέχει η εφαρμογή στη συσκευή.

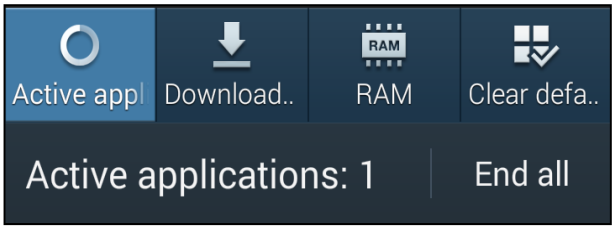
Εικόνα 3.13: Στο συγκεκριμένο παράδειγμα τρέχει η εφαρμογή Enemy Strike.

II. Επιλέγουμε τις ρυθμίσεις της συσκευής.



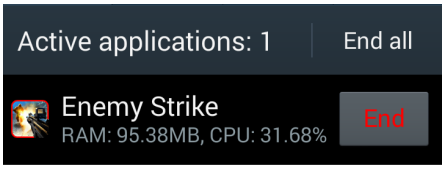
Εικόνα 3.14:Settings συσκευής Android.

III. Ακολούθως προχωράμε στις ενεργές εφαρμογές.

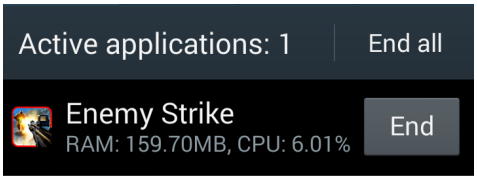


Εικόνα 3.15: Επιλογή ενεργών εφαρμογών.

IV. Έτσι βλέπουμε τη χρήση των RAM και CPU που κάνει η εφαρμογή Enemy Strike.

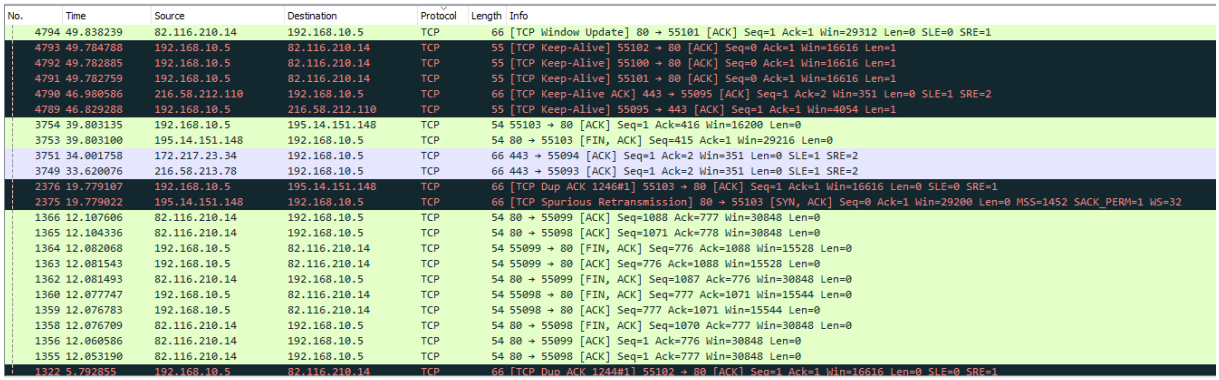


Εικόνα 3.16: Enemy Strike Play Store



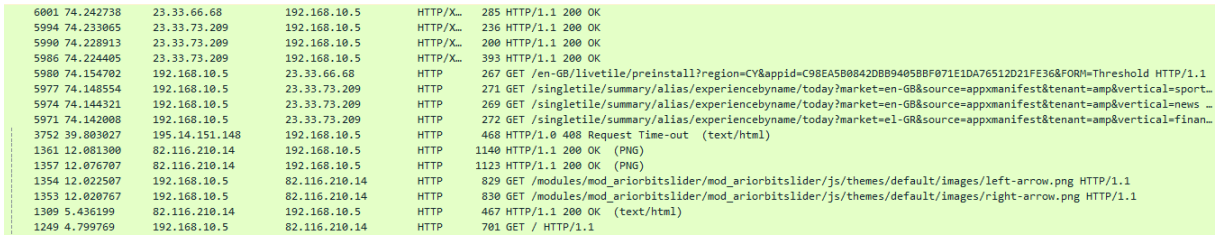
Εικόνα 3.17: Enemy Strike Cracked

4. Με τη βοήθεια του προγράμματος Wireshark παρακολουθούμε το Network Traffic. Επιλέγουμε το πρωτόκολλο που θέλουμε να παρακολουθήσουμε και εμφανίζονται τα αποτελέσματα. Η ίδια διαδικασία έγινε για τις εφαρμογές από το Play Store αλλά και για τις εφαρμογές από τις ιστοσελίδες.



No.	Time	Source	Destination	Protocol	Length	Info
4794	49.838239	82.116.210.14	192.168.10.5	TCP	66	[TCP Window Update] 80 → 55101 [ACK] Seq=1 Ack=1 Win=29312 Len=0 SLE=0 SRE=1
4793	49.784788	192.168.10.5	82.116.210.14	TCP	55	[TCP Keep-Alive] 55102 → 80 [ACK] Seq=0 Ack=1 Win=16616 Len=1
4792	49.782885	192.168.10.5	82.116.210.14	TCP	55	[TCP Keep-Alive] 55100 → 80 [ACK] Seq=0 Ack=1 Win=16616 Len=1
4791	49.782759	192.168.10.5	82.116.210.14	TCP	55	[TCP Keep-Alive] 55101 → 80 [ACK] Seq=0 Ack=1 Win=16616 Len=1
4790	46.980586	216.58.212.110	192.168.10.5	TCP	66	[TCP Keep-Alive ACK] 443 → 55095 [ACK] Seq=1 Ack=2 Win=351 Len=0 SLE=1 SRE=2
4789	46.829288	192.168.10.5	216.58.212.110	TCP	55	[TCP Keep-Alive] 55095 → 443 [ACK] Seq=1 Ack=1 Win=4054 Len=1
3754	39.803135	192.168.10.5	195.14.151.148	TCP	54	55103 → 80 [ACK] Seq=1 Ack=416 Win=16200 Len=0
3753	39.803100	195.14.151.148	192.168.10.5	TCP	54	80 → 55103 [FIN, ACK] Seq=415 Ack=1 Win=29216 Len=0
3751	34.001758	172.217.23.34	192.168.10.5	TCP	66	443 → 55094 [ACK] Seq=1 Ack=2 Win=351 Len=0 SLE=1 SRE=2
3749	33.620076	216.58.213.78	192.168.10.5	TCP	66	443 → 55093 [ACK] Seq=1 Ack=2 Win=351 Len=0 SLE=1 SRE=2
2376	19.779107	192.168.10.5	195.14.151.148	TCP	66	[TCP Dup ACK 1246#1] 55103 → 80 [ACK] Seq=1 Ack=1 Win=16616 Len=0 SLE=0 SRE=1
2375	19.779022	195.14.151.148	192.168.10.5	TCP	66	[TCP Spurious Retransmission] 80 → 55103 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1452 SACK_PERM=1 WS=32
1366	12.107606	82.116.210.14	192.168.10.5	TCP	54	80 → 55099 [ACK] Seq=1088 Ack=777 Win=30848 Len=0
1365	12.104336	82.116.210.14	192.168.10.5	TCP	54	80 → 55098 [ACK] Seq=1071 Ack=778 Win=30848 Len=0
1364	12.082068	192.168.10.5	82.116.210.14	TCP	54	55099 → 80 [FIN, ACK] Seq=776 Ack=1088 Win=15528 Len=0
1363	12.081543	192.168.10.5	82.116.210.14	TCP	54	55099 → 80 [ACK] Seq=776 Ack=1088 Win=15528 Len=0
1362	12.081493	82.116.210.14	192.168.10.5	TCP	54	80 → 55099 [FIN, ACK] Seq=1087 Ack=776 Win=30848 Len=0
1360	12.077747	192.168.10.5	82.116.210.14	TCP	54	55098 → 80 [FIN, ACK] Seq=777 Ack=1071 Win=15544 Len=0
1359	12.076783	192.168.10.5	82.116.210.14	TCP	54	55098 → 80 [ACK] Seq=777 Ack=1071 Win=15544 Len=0
1358	12.076709	82.116.210.14	192.168.10.5	TCP	54	80 → 55098 [FIN, ACK] Seq=1070 Ack=777 Win=30848 Len=0
1356	12.060586	82.116.210.14	192.168.10.5	TCP	54	80 → 55099 [ACK] Seq=1 Ack=776 Win=30848 Len=0
1355	12.053190	82.116.210.14	192.168.10.5	TCP	54	80 → 55098 [ACK] Seq=1 Ack=777 Win=30848 Len=0
1322	5.792855	192.168.10.5	82.116.210.14	TCP	66	[TCP Dup ACK 1244#1] 55102 → 80 [ACK] Seq=1 Ack=1 Win=16616 Len=0 SLE=0 SRE=1

Εικόνα 3.18: TCP PROTOCOL



6001	74.242738	23.33.66.68	192.168.10.5	HTTP/X…	285	HTTP/1.1 200 OK
5994	74.233065	23.33.73.209	192.168.10.5	HTTP/X…	236	HTTP/1.1 200 OK
5990	74.228913	23.33.73.209	192.168.10.5	HTTP/X…	200	HTTP/1.1 200 OK
5986	74.224405	23.33.73.209	192.168.10.5	HTTP/X…	393	HTTP/1.1 200 OK
5980	74.154702	192.168.10.5	23.33.66.68	HTTP	267	GET /en-GB/livetile/preinstall?region=CY&appid=C98EA5B0842DBB9405BBF071E1DA76512D21FE36&FORM=Threshold HTTP/1.1
5977	74.148554	192.168.10.5	23.33.73.209	HTTP	271	GET /singletile/summary/alias/experiencebyname/today?market=en-GB&source=appxmanifest&tenant=amp&vertical=sport…
5974	74.144321	192.168.10.5	23.33.73.209	HTTP	269	GET /singletile/summary/alias/experiencebyname/today?market=en-GB&source=appxmanifest&tenant=amp&vertical=news …
5971	74.142008	192.168.10.5	23.33.73.209	HTTP	272	GET /singletile/summary/alias/experiencebyname/today?market=el-GR&source=appxmanifest&tenant=amp&vertical=finan…
3752	39.803027	195.14.151.148	192.168.10.5	HTTP	468	HTTP/1.0 408 Request Time-out (text/html)
1361	12.081300	82.116.210.14	192.168.10.5	HTTP	1140	HTTP/1.1 200 OK (PNG)
1357	12.076707	82.116.210.14	192.168.10.5	HTTP	1123	HTTP/1.1 200 OK (PNG)
1354	12.022507	192.168.10.5	82.116.210.14	HTTP	829	GET /modules/mod_ariorbitslider/mod_ariorbitslider/js/themes/default/images/left-arrow.png HTTP/1.1
1353	12.020767	192.168.10.5	82.116.210.14	HTTP	830	GET /modules/mod_ariorbitslider/mod_ariorbitslider/js/themes/default/images/right-arrow.png HTTP/1.1
1309	5.436199	82.116.210.14	192.168.10.5	HTTP	467	HTTP/1.1 200 OK (text/html)
1249	4.799769	192.168.10.5	82.116.210.14	HTTP	701	GET / HTTP/1.1

Εικόνα 3.19: HTTP PROTOCOL

Κεφάλαιο 4
Ανάλυση Πειραματικών Αποτελεσμάτων

Στο παρόν κεφάλαιο αναλύονται τα δικαιώματα, η χρήση της RAM και CPU, το Network Traffic που σημειώνουν οι εφαρμογές που προέρχονται από το Play Store αλλά και αυτές που είναι Cracked.

4.1 Android Permissions

Για να διατηρηθεί η ασφάλεια του συστήματος και των χρηστών, το Android απαιτεί από τις εφαρμογές να ζητούν άδεια πριν οι εφαρμογές μπορούν να χρησιμοποιήσουν συγκεκριμένα δεδομένα και λειτουργίες του συστήματος. Ανάλογα με το πόσο ευαίσθητη είναι η περιοχή, το σύστημα μπορεί να χορηγήσει αυτόματα την άδεια ή μπορεί να ζητήσει από το χρήστη να

εγκρίνει το αίτημα. Τα σημεία που διαχωρίζουν τις εφαρμογές σε καλοήθεις και κακόβουλες εφαρμογές είναι τα δικαιώματα που ζητάνε από τον χρήστη ή τις συσκευές. Τα δικαιώματα εμφανίζονται σε Normal Android Permissions και σε Dangerous Android Permissions από την Android [104].

4.1.1 Normal – Dangerous Android Permissions

Τα Normal Android Permissions καλύπτουν περιοχές όπου η εφαρμογή μας χρειάζεται πρόσβαση σε δεδομένα ή πόρους έξω από το Sandbox της εφαρμογής, αλλά όπου υπάρχει πολύ μικρός κίνδυνος για το απόρρητο του χρήστη ή για τη λειτουργία άλλων εφαρμογών. Για παράδειγμα, η άδεια για τη ρύθμιση της ζώνης ώρας είναι μια κανονική άδεια. Εάν μια εφαρμογή δηλώνει ότι χρειάζεται κανονική άδεια, το σύστημα δίνει αυτόματα την άδεια στην εφαρμογή.

Τα Dangerous Android Permissions καλύπτουν περιοχές όπου η εφαρμογή θέλει δεδομένα ή πηγές που αφορούν τις προσωπικές πληροφορίες του χρήστη ή ενδέχεται να επηρεάσουν τα αποθηκευμένα δεδομένα του χρήστη ή τη λειτουργία άλλων εφαρμογών. Για παράδειγμα, η δυνατότητα ανάγνωσης των επαφών του χρήστη είναι μια επικίνδυνη άδεια. Εάν μια εφαρμογή δηλώνει ότι χρειάζεται επικίνδυνη άδεια, ο χρήστης πρέπει να δώσει ρητά την άδεια στην εφαρμογή. Μία λίστα επικίνδυνων δικαιωμάτων είναι η πιο κάτω.

4.1.2 Ανάλυση δικαιωμάτων των Play Store και Cracked εφαρμογών

Λαμβάνοντας υπόψη τα δικαιώματα (Permissions) που αναφέρονται πιο πάνω κατέληξα στην ανάλυση εφαρμογών που διατίθενται στο Google Play Store αλλά και σε Third Party Markets όπως http://www.crackapk.com και το http://www.appcake.net/ στα οποία οι εφαρμογές βρίσκονται κρακαρισμένες. Οι εφαρμογές αναγνωρίζονται ως κακόβουλες από τα δικαιώματα που χρησιμοποιούν και τον κώδικα τους.

Παρακάτω εξηγείται ο σκοπός των δεκαέξι αιτούμενων δικαιωμάτων που βρέθηκαν κατά την ανάλυση των εφαρμογών [105], τα δικαιώματα αυτά φαίνονται στον **Πίνακα 4.1** (Παράρτημα Γ):

1. android.permission.INTERNET: Επιτρέπει σε μια εφαρμογή να εγκαταστήσει μια συντόμευση στο Launcher. Με επίπεδο προστασίας κανονικό.

2. android.permission.ACCESS_COARSE_LOCATION: Επιτρέπει σε μια εφαρμογή να έχει πρόσβαση κατά προσέγγιση σε κάποια τοποθεσία. Το επίπεδο προστασίας θεωρείται επικίνδυνο.

3. android.permission.ACCESS_FINE_LOCATION: Επιτρέπει σε μια εφαρμογή τη πρόσβαση σε ακριβή τοποθεσία. Με επίπεδο προστασίας επικίνδυνο.

4. android.permission.CAMERA: Επιβάλλει αυτομάτως το στοιχείο manifest για όλες τις λειτουργίες της κάμερας. Εάν δεν απαιτούνται όλες οι λειτουργίες της κάμερας ή μπορεί να λειτουργήσει σωστά εάν δεν είναι διαθέσιμη μια φωτογραφική μηχανή, τότε πρέπει να τροποποιηθεί το manifest κατάλληλα για να εγκατασταθεί σε συσκευές που δεν υποστηρίζουν όλες τις λειτουργίες της κάμερας. Το επίπεδο προστασίας είναι επικίνδυνο.

5. android.permission.BLUETOOTH: Επιτρέπει στις εφαρμογές τη σύνδεση σε συνδυασμένες συσκευές Bluetooth. Το επίπεδο προστασίας κανονικό.

6. android.permission.READ_EXTERNAL_STORAGE: Αυτή η άδεια δεν επιβάλλεται και όλες οι εφαρμογές εξακολουθούν να έχουν πρόσβαση για ανάγνωση από εξωτερικό αποθηκευτικό χώρο. Με το επίπεδο προστασίας να είναι επικίνδυνο.

7. android.permission.READ_CONTACTS: Επιτρέπει σε μια εφαρμογή να διαβάσει τα δεδομένα των επαφών του χρήστη και το επίπεδο προστασίας είναι επικίνδυνο.

8. android.permission.WRITE_EXTERNAL_STORAGE: Επιτρέπει σε μια εφαρμογή να γράφει σε εξωτερικό αποθηκευτικό χώρο. Το επίπεδο προστασίας είναι επικίνδυνο.

9. android.permission.MEDIA_CONTENT_CONTROL: Επιτρέπει σε μια εφαρμογή να γνωρίζει ποιο περιεχόμενο αναπαράγεται και ελέγχει την αναπαραγωγή του. Δεν γίνεται χρήση από third-party εφαρμογές λόγω της προστασίας της ιδιωτικής ζωής της που καταναλώνουν τα μέσα.

10. android.permission.READ_SMS: Επιτρέπει σε μια εφαρμογή να διαβάζει μηνύματα SMS. Το επίπεδο προστασίας το κατατάσσει ως επικίνδυνο.

11. android.permission.RECEIVE_SMS: Επιτρέπει σε μια εφαρμογή να λαμβάνει μηνύματα SMS, με επίπεδο προστασίας επικίνδυνο.

12. android.permission.SEND_SMS: Επιτρέπει σε μια εφαρμογή να στέλνει μηνύματα SMS. Το επίπεδο προστασίας θεωρείται επικίνδυνο.

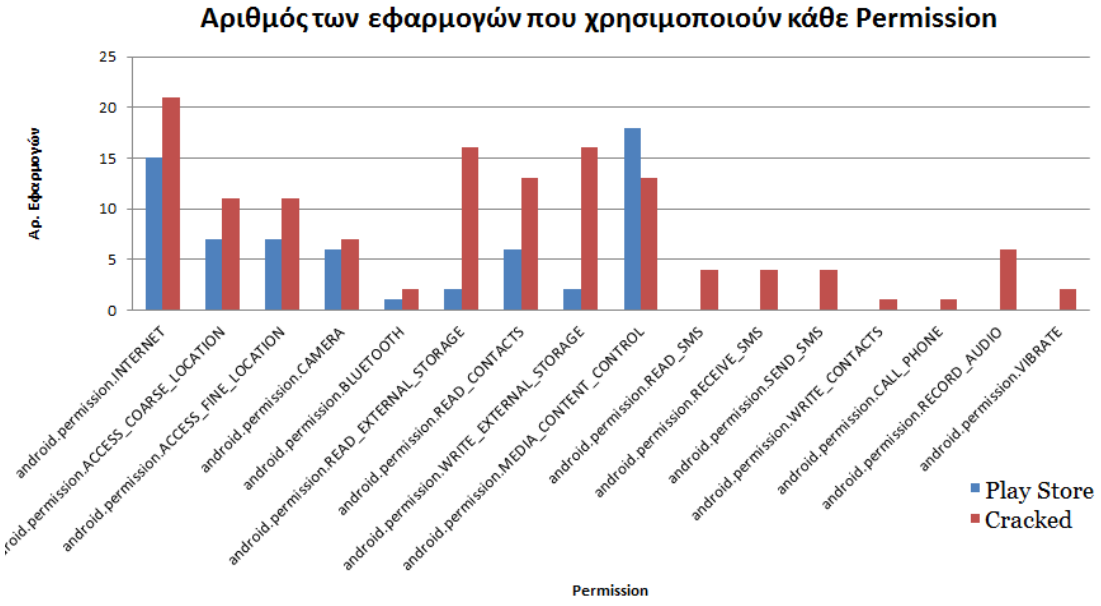
13. android.permission.WRITE_CONTACTS: Επιτρέπει σε μια εφαρμογή να γράψει τα δεδομένα επαφών του χρήστη. Το επίπεδο προστασίας θεωρείται επικίνδυνο.

14. android.permission.CALL_PHONE: Επιτρέπει σε μια εφαρμογή να ξεκινήσει μια τηλεφωνική κλήση χωρίς να περάσει από τη διεπαφή χρήστη του Dialer για να επιβεβαιώσει την κλήση ο χρήστης. Το επίπεδο προστασίας κατατάσσεται ως επικίνδυνο.

15. android.permission.RECORD_AUDIO: Επιτρέπει σε μια εφαρμογή την εγγραφή ήχου, με το επίπεδο προστασίας να είναι επικίνδυνο.

16. android.permission.VIBRATE: Επιτρέπει την πρόσβαση στον δονητή και το επίπεδο προστασίας θεωρείται κανονικό.

Πιο κάτω ακολουθεί δύο γραφήματα που περιγράφουν τον αριθμό των εφαρμογών που ζητούν το κάθε permission αλλά και τον αριθμό των permissions που ζητά η κάθε εφαρμογή σύμφωνα με τον **Πίνακα 4.1** (Παράρτημα Γ) ο οποίος περιγράφει τα δικαιώματα που αιτούνται οι εφαρμογές από το Play Store (μπλε χρώμα) και οι εφαρμογές που είναι κρακαρισμένες και είναι από Third Party Markets (πράσινο χρώμα).



Γράφημα 4.1: Αριθμός εφαρμογών που ζητούν το κάθε permission.

Στο Γράφημα 4.1, παρατηρούμε ότι οι εφαρμογές που είναι κρακαρισμένες αιτούνται περισσότερα δικαιώματα από αυτές που προέρχονται από το Play Store. Ακόμη τα δικαιώματα που αφορούν READ_SMS, RECEIVE_SMS, SEND_SMS, WRITE_CONTACTS, CALL_PHONE, RECORD_AUDIO και VIBRATE χρησιμοποιούνται μόνο από τις εφαρμογές που είναι κρακαρισμένες.

Γράφημα 4.2: Αριθμός permissions που ζητά η κάθε εφαρμογή.

Είναι εμφανές στο Γράφημα 4.2, πώς οι κρακαρισμένες εφαρμογές σε σχέση με αυτές που προέρχονται από το Play Store αιτούνται αρκετά περισσότερα δικαιώματα.

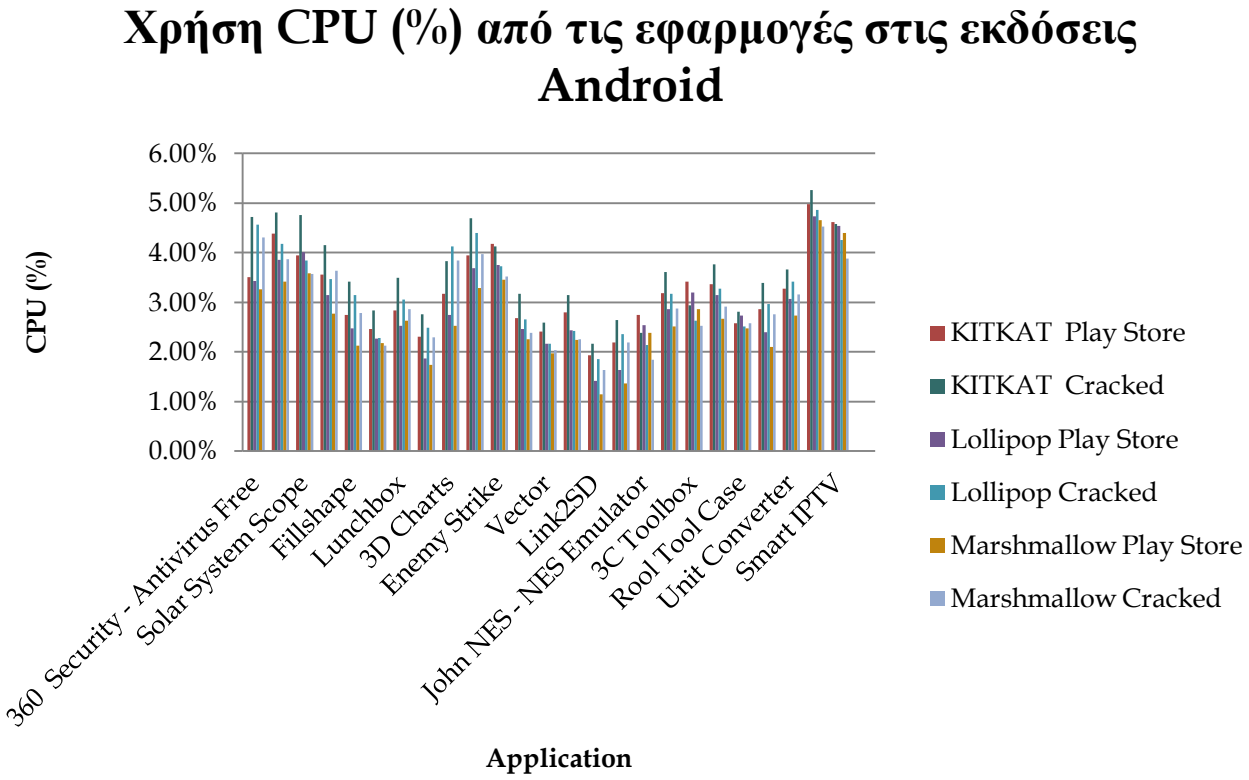
4.3 Ανάλυση χρήσης Ram και CPU των Android εκδόσεων KitKat, Lollipop και Marshmallow τις Play Store και Cracked εφαρμογές

Σε αυτό το μέρος της μελέτης, αναλύονται η χρήση των CPU και RAM που γίνεται από τις εφαρμογές που προέρχονται από το Play Store και από Third Party Markets (αναφέρονται ως Cracked). Η ανάλυση έγινε μετά από πειραματικές δοκιμές στις τρεις διαφορετικές συσκευές

που χρησιμοποιούν διαφορετικές εκδόσεις λογισμικού KITKAT, Lollipop και Marshmallow, οι συσκευές είναι οι: Samsung Galaxy S3 neo, Samsung Galaxy S5 και Samsung Galaxy S7 αντίστοιχα.

4.3.1 Μελέτη της χρήσης CPU από τις εφαρμογές

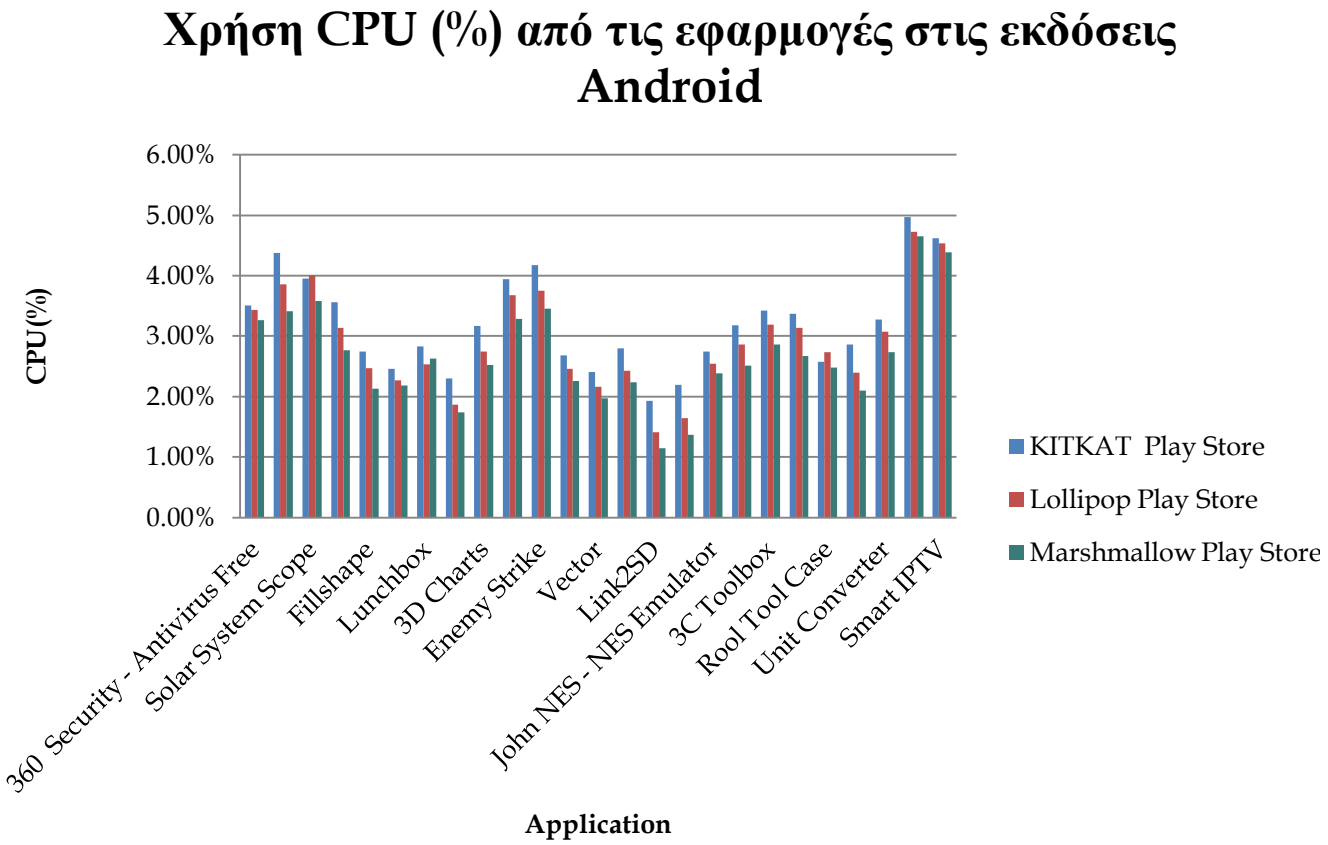
Ακολουθεί η γραφική απεικόνιση της χρήσης CPU (%) από τις εφαρμογές που προέρχονται από το Play Store και αυτών που είναι Cracked στις τρεις υπό μελέτη εκδόσεις Android, με βάση τον **Πίνακα 4.2** (Παράρτημα Γ))στον οποίο φαίνεται η χρήση της CPU (%) από τις εφαρμογές που προέρχονται από το Play Store και από τις εφαρμογές που είναι Cracked (από Third Party Stores) στην κάθε Android έκδοση.



Γράφημα 4.3: Χρήση CPU (%) από τις εφαρμογές στις εκδόσεις Android.

Στην πιο πάνω γραφική παράσταση (Γράφημα 4.3) παρουσιάζεται η χρήση της CPU (%) όλων των εφαρμογών ανάλογα με την προέλευσης τους σε όλες τις εκδόσεις που μελετώνται. Παρατηρούμε ότι οι κρακαρισμένες εφαρμογές κάνουν μεγαλύτερη χρήση της CPU σε σχέση με αυτές που προέρχονται από το Play Store. Ακόμη αξίζει να σημειωθεί ότι οι εφαρμογές και των δύο προελεύσεων με την ανάπτυξη νέων εκδόσεων Android χρησιμοποιούν λιγότερη CPU (%).

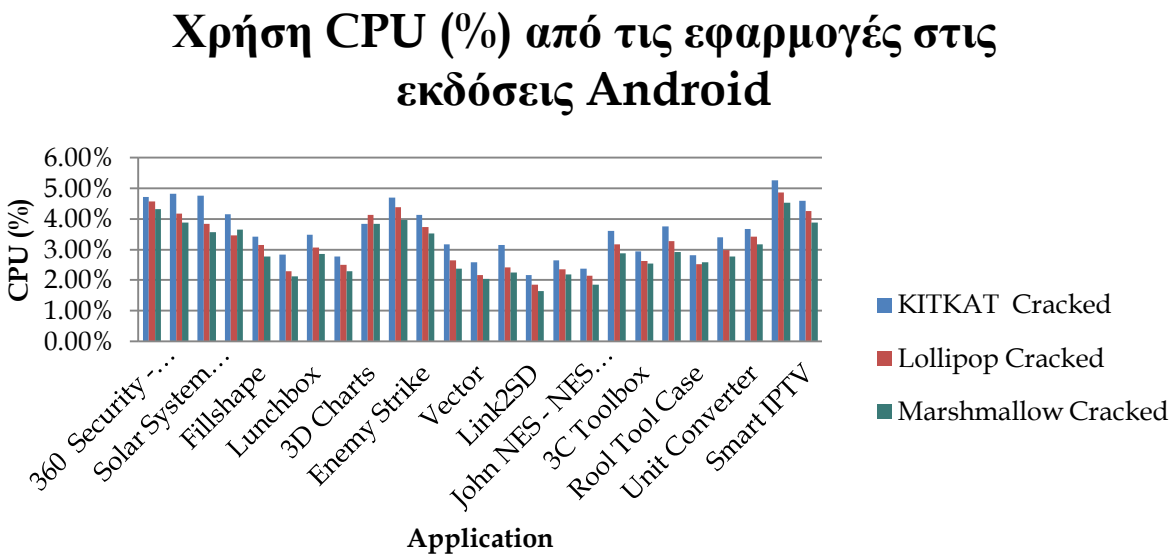
Ακολουθεί μία πιο αναλυτική μελέτη της χρήσης CPU (%) από τις εφαρμογές που προέρχονται από το Play Store στις εκδόσεις KitKat, Lollipop και Marshmallow.



Γράφημα 4.4: Χρήση CPU (%) από τις εφαρμογές που προέρχονται από το Play Store στις εκδόσεις Android.

Παρατηρούμε (Γράφημα 4.4) ότι όλες οι εφαρμογές που προέρχονται από το Play Store κάνουν μικρότερη χρήση της CPU με την ανάπτυξη νέων εκδόσεων Android. Σημειώνεται ότι στην έκδοση Marshmallow γίνεται μικρότερη χρήση της CPU , μετά ακολουθεί η έκδοση Lollipop και KitKat, όπως και η σειρά έκδοσης τους.

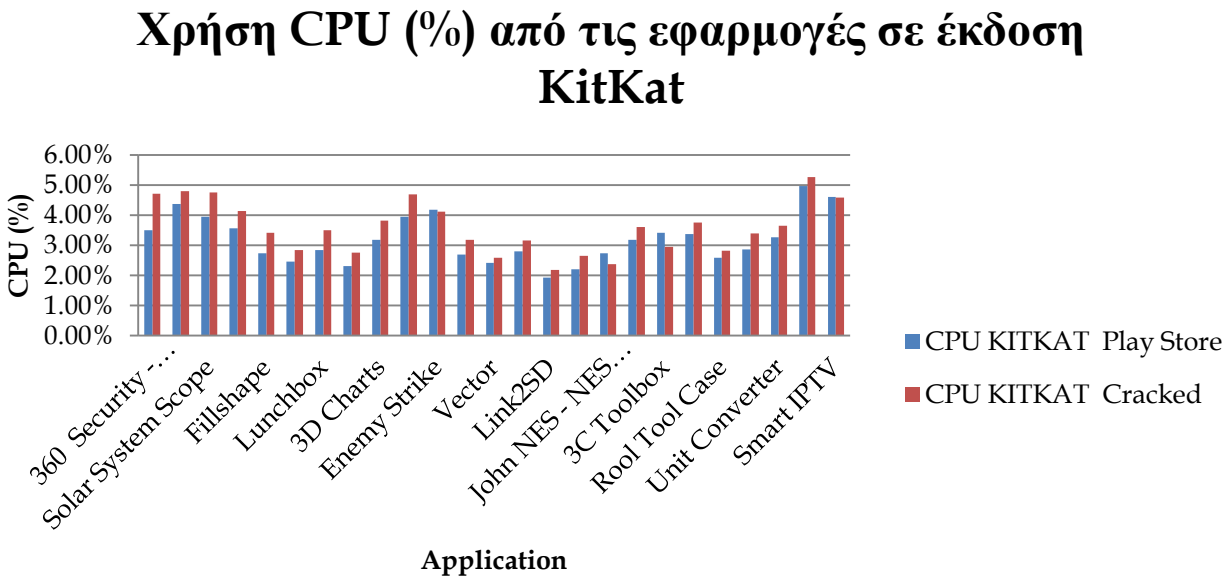
Στην συνέχεια υπάρχει μία πιο αναλυτική μελέτη της χρήσης CPU (%) από τις εφαρμογές που είναι Cracked στις εκδόσεις KitKat, Lollipop και Marshmallow.



Γράφημα 4.5: Χρήση CPU (%) από τις εφαρμογές που είναι Cracked στις εκδόσεις Android.

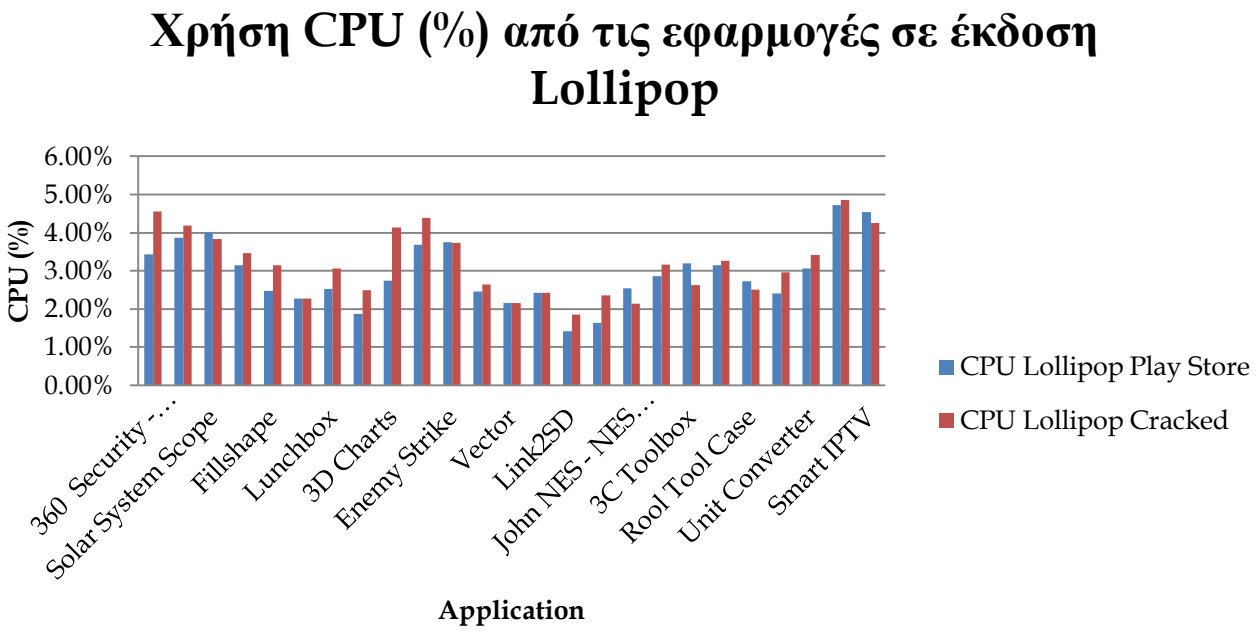
Παρατηρούμε (Γράφημα 4.5) ότι όλες οι εφαρμογές που προέρχονται που είναι Cracked κάνουν μικρότερη χρήση της CPU με την ανάπτυξη νέων εκδόσεων Android. Σημειώνεται ότι στην έκδοση Marshmallow γίνεται μικρότερη χρήση της CPU , μετά ακολουθεί η έκδοση Lollipop και KitKat, όπως και η σειρά έκδοσης τους.

Στα τρία επόμενα γραφήματα που ακολουθούν, περιγράφεται η χρήση της CPU (%) από τις εφαρμογές στις εκδόσεις KitKat, Lollipop και Marshmallow, έτσι μπορούμε να αντιληφθούμε την κατανάλωση CPU που κάνουν οι εφαρμογές ανάλογα με τη προέλευση τους.



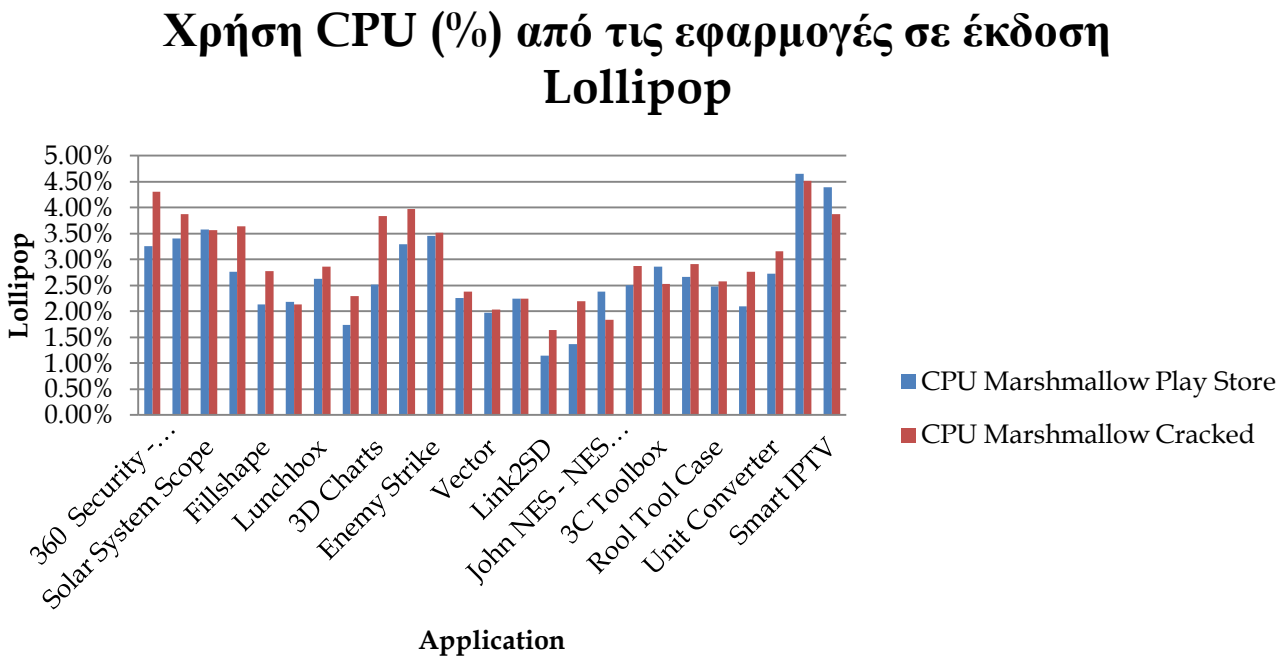
Γράφημα 4.6: Χρήση CPU (%) από τις εφαρμογές στην έκδοση KitKat.

Βλέπουμε στο Γράφημα 4.6, ότι οι εφαρμογές που είναι Cracked σε σχέση με αυτές που προέρχονται από το Play Store και αναλύονται στην έκδοση KitKat κάνουν μεγαλύτερη χρήση της CPU (%).



Γράφημα 4.7: Χρήση CPU (%) από τις εφαρμογές στην έκδοση Lollipop.

Στο Γράφημα 4.7 οι εφαρμογές που είναι Cracked σε σχέση με αυτές που προέρχονται από το Play Store και αναλύονται στην έκδοση Lollipop κάνουν μεγαλύτερη χρήση της CPU (%) με εξαίρεση την εφαρμογή Smart IPTV.

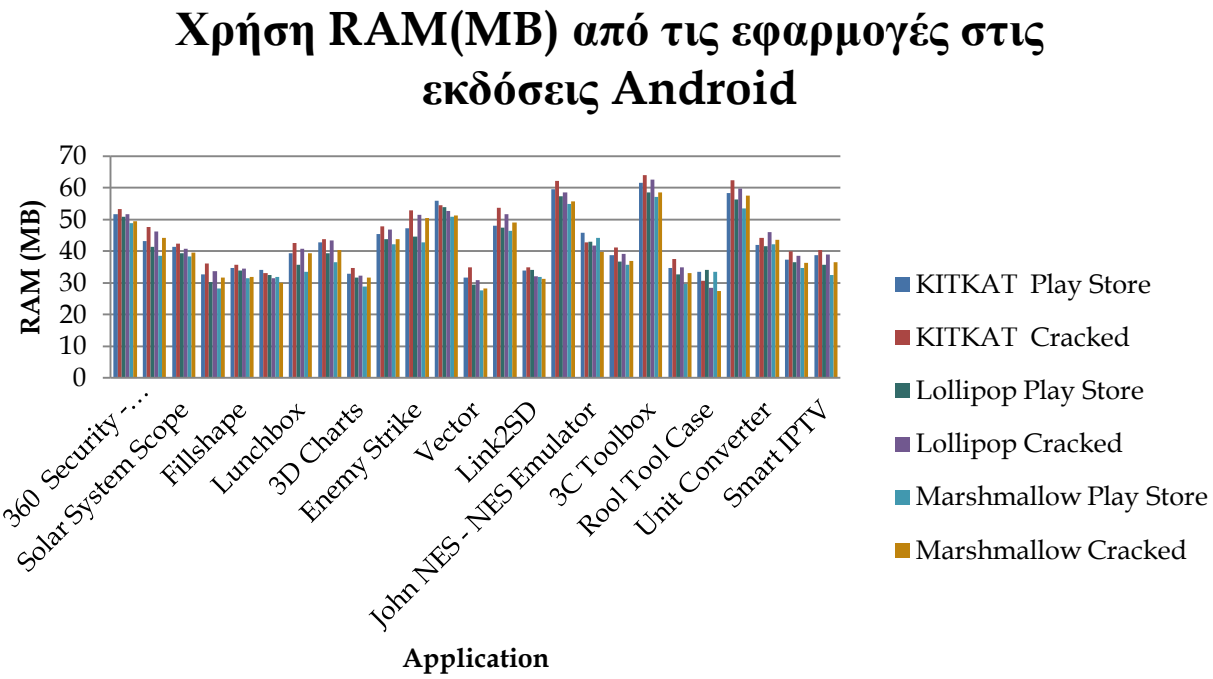


Γράφημα 4.8: Χρήση CPU (%) από τις εφαρμογές στην έκδοση Marshmallow.

Οι εφαρμογές που είναι Cracked σε σχέση με αυτές που προέρχονται από το Play Store και αναλύονται στην έκδοση Marshmallow κάνουν μεγαλύτερη χρήση της CPU (%) με εξαίρεση την εφαρμογή Smart IPTV (Γράφημα 4.8).

4.3.2 Μελέτη της χρήσης RAM από τις εφαρμογές

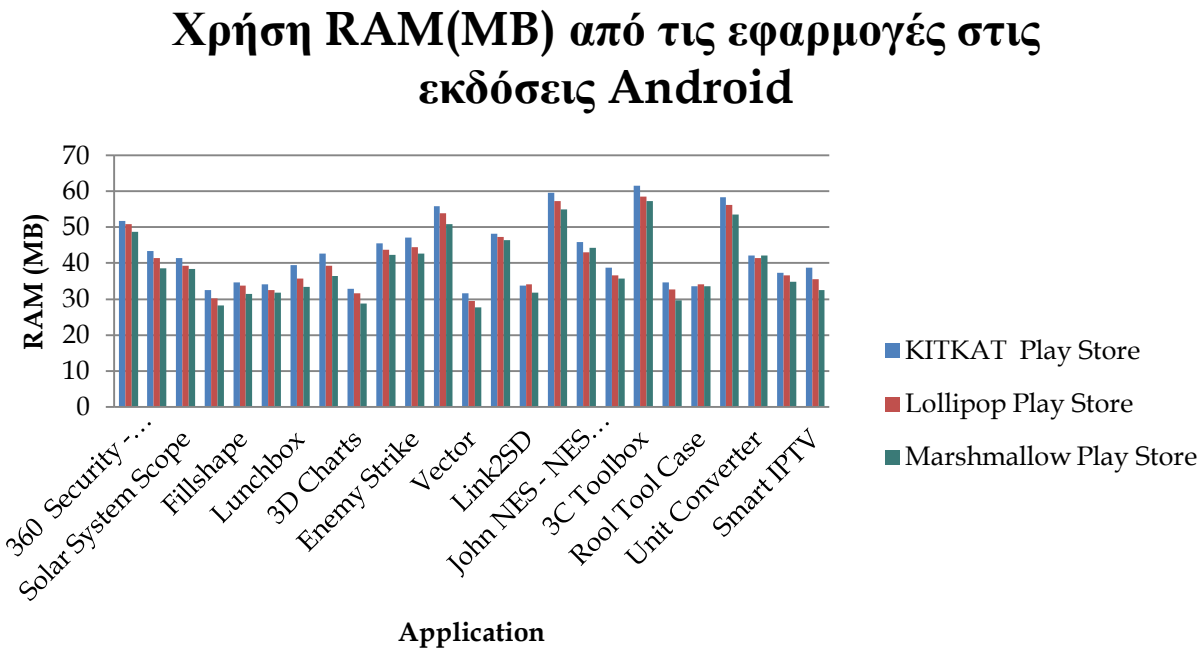
Ακολουθεί η γραφική απεικόνιση της χρήσης RAM (MB) από τις εφαρμογές που προέρχονται από το Play Store και αυτών που είναι Cracked στις τρεις υπό μελέτη εκδόσεις Android, σύμφωνα με τον **Πίνακα 4.3** (Παράρτημα Γ) οποίο φαίνεται η χρήση της RAM (MB) από τις εφαρμογές που προέρχονται από το Play Store και από τις εφαρμογές που είναι Cracked (από Third Party Stores) στην κάθε Android έκδοση.



Γράφημα 4.9: Χρήση RAM (MB) από τις εφαρμογές στις εκδόσεις Android.

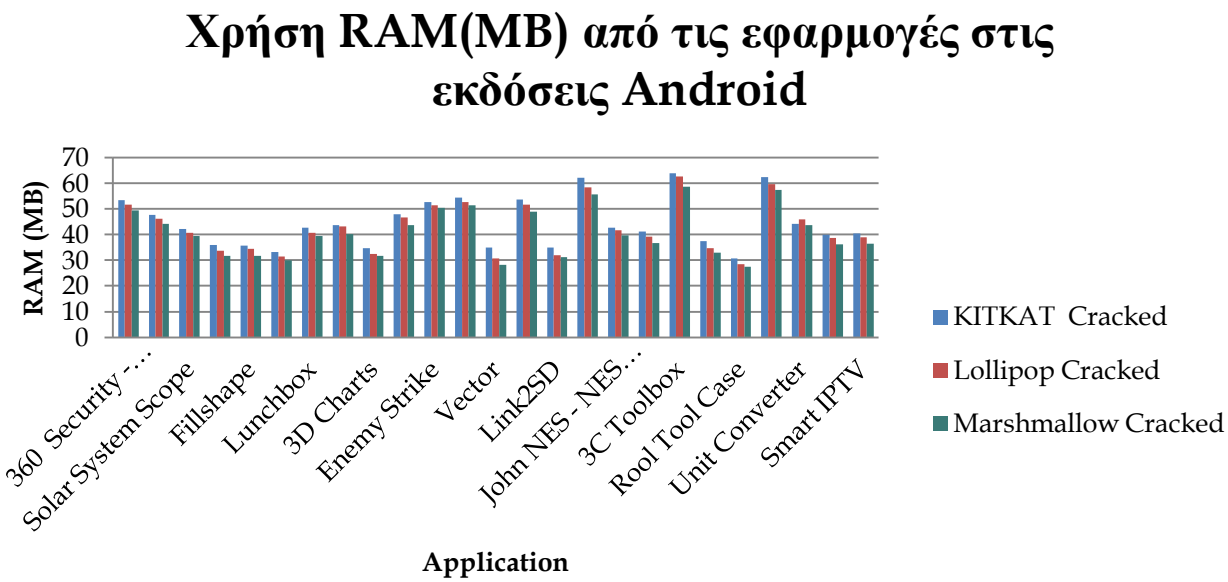
Στην πιο πάνω γραφική παράσταση (Γράφημα 4.9) παρουσιάζεται η χρήση της RAM (MB) όλων των εφαρμογών ανάλογα με την προέλευσης τους σε όλες τις εκδόσεις που μελετώνται. Παρατηρούμε ότι οι κρακαρισμένες εφαρμογές κάνουν μεγαλύτερη χρήση της RAM σε σχέση με αυτές που προέρχονται από το Play Store. Ακόμη αξίζει να σημειωθεί ότι οι εφαρμογές και των δύο προελεύσεων με την ανάπτυξη νέων εκδόσεων Android χρησιμοποιούν λιγότερη RAM (MB).

Ακολουθεί μία πιο αναλυτική μελέτη της χρήσης RAM (MB) από τις εφαρμογές που προέρχονται από το Play Store στις εκδόσεις KitKat, Lollipop και Marshmallow.



Γράφημα 4.10: Χρήση RAM (MB) από τις εφαρμογές που προέρχονται από το Play Store στις εκδόσεις Android.

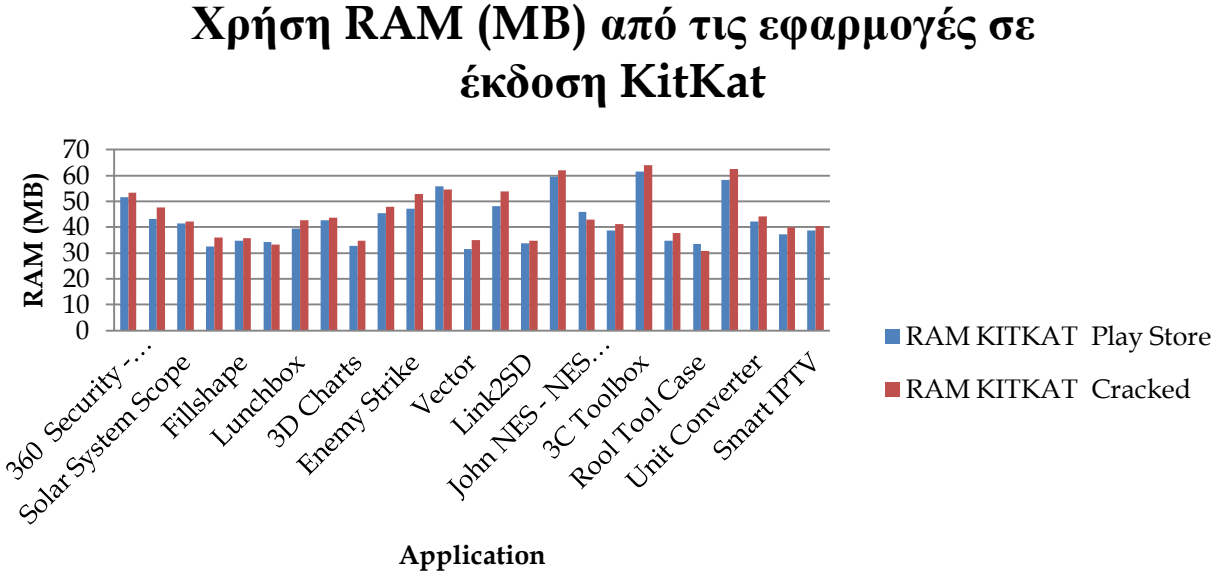
Είναι εμφανές από το Γράφημα 4.10, πώς οι εφαρμογές που προέρχονται από το Play Store κάνουν μικρότερη χρήση της RAM με την ανάπτυξη νέων εκδόσεων Android. Σημειώνεται ότι στην έκδοση Marshmallow γίνεται μικρότερη χρήση της CPU , μετά ακολουθεί η έκδοση Lollipop και KitKat, όπως και η σειρά έκδοσης τους.



Γράφημα 4.11: Χρήση RAM (MB) από τις εφαρμογές που είναι Cracked στις εκδόσεις Android.

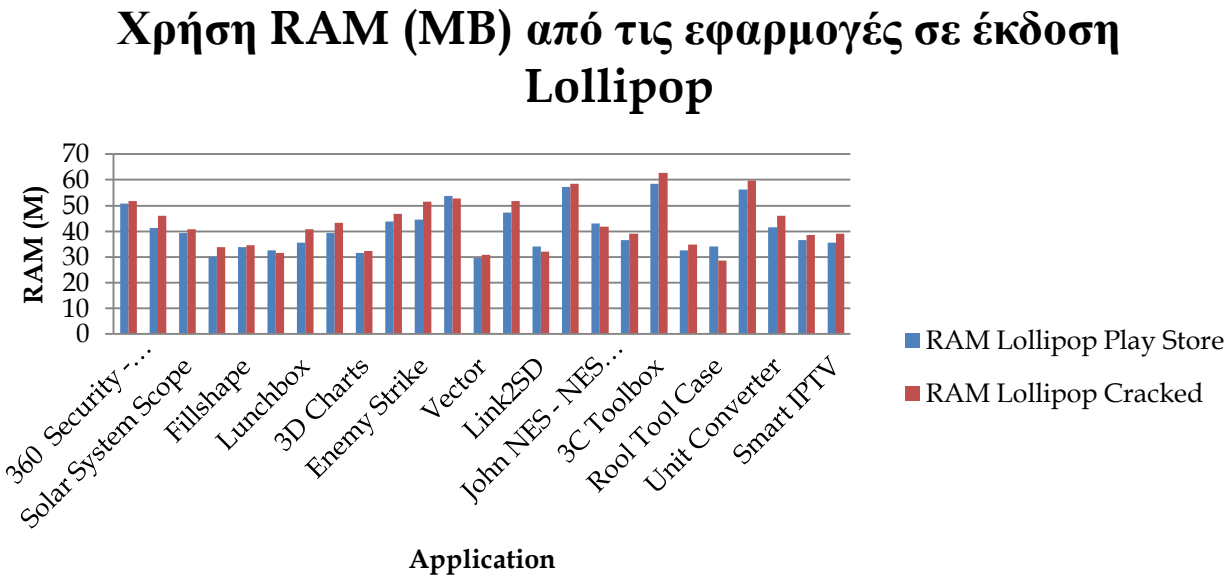
Οι εφαρμογές (Γράφημα 4.11) που προέρχονται που είναι Cracked κάνουν μικρότερη χρήση της RAM με την ανάπτυξη νέων εκδόσεων Android. Σημειώνεται ότι στην έκδοση Marshmallow γίνεται μικρότερη χρήση της CPU , μετά ακολουθεί η έκδοση Lollipop και KitKat, όπως και η σειρά έκδοσης τους.

Στα τρία επόμενα γραφήματα που ακολουθούν, περιγράφεται η χρήση της CPU (%) από τις εφαρμογές στις εκδόσεις KitKat, Lollipop και Marshmallow, έτσι μπορούμε να αντιληφθούμε την κατανάλωση CPU που κάνουν οι εφαρμογές ανάλογα με τη προέλευση τους.



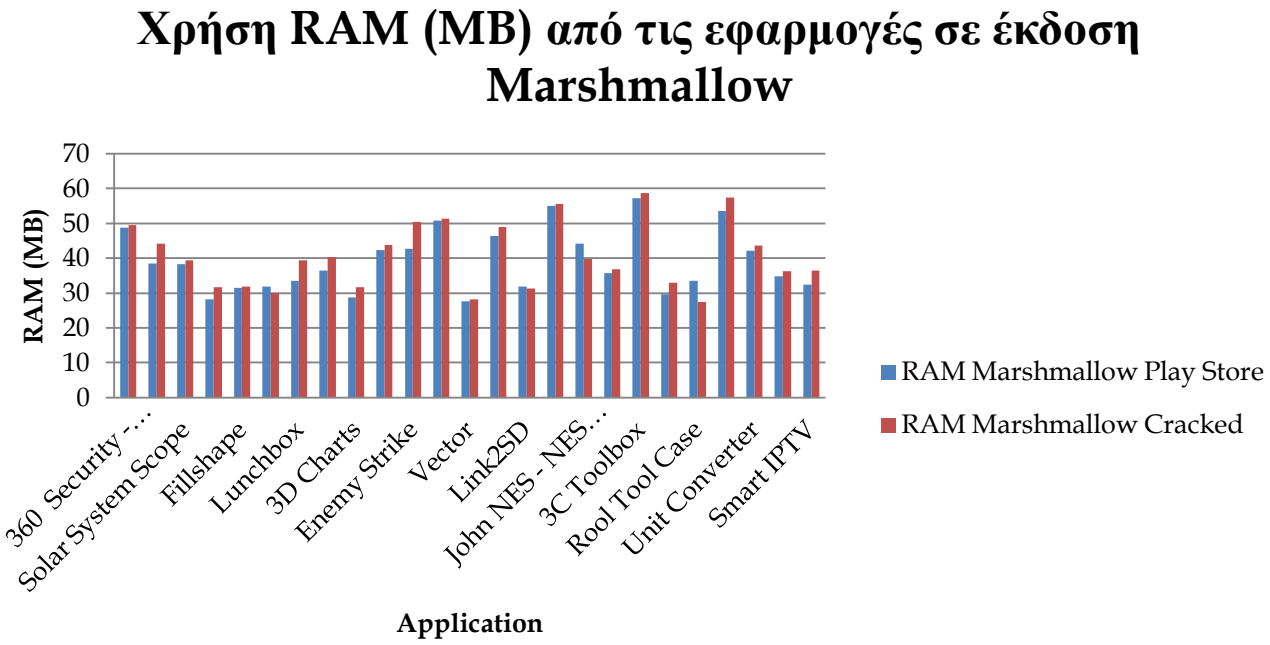
Γράφημα 4.12: Χρήση RAM (MB) από τις εφαρμογές στην έκδοση KitKat.

Βλέπουμε στο Γράφημα 4.12, ότι οι εφαρμογές που είναι Cracked σε σχέση με αυτές που προέρχονται από το Play Store και αναλύονται στην έκδοση KitKat κάνουν μεγαλύτερη χρήση της RAM (MB).



Γράφημα 4.13: Χρήση RAM (MB) από τις εφαρμογές στην έκδοση Lollipop.

Οι Cracked εφαρμογές σύμφωνα με το Γράφημα 4.13 σε σχέση με τις εφαρμογές που προέρχονται από το Play Store και αναλύονται στην έκδοση Lollipop κάνουν μεγαλύτερη χρήση της RAM (MB).

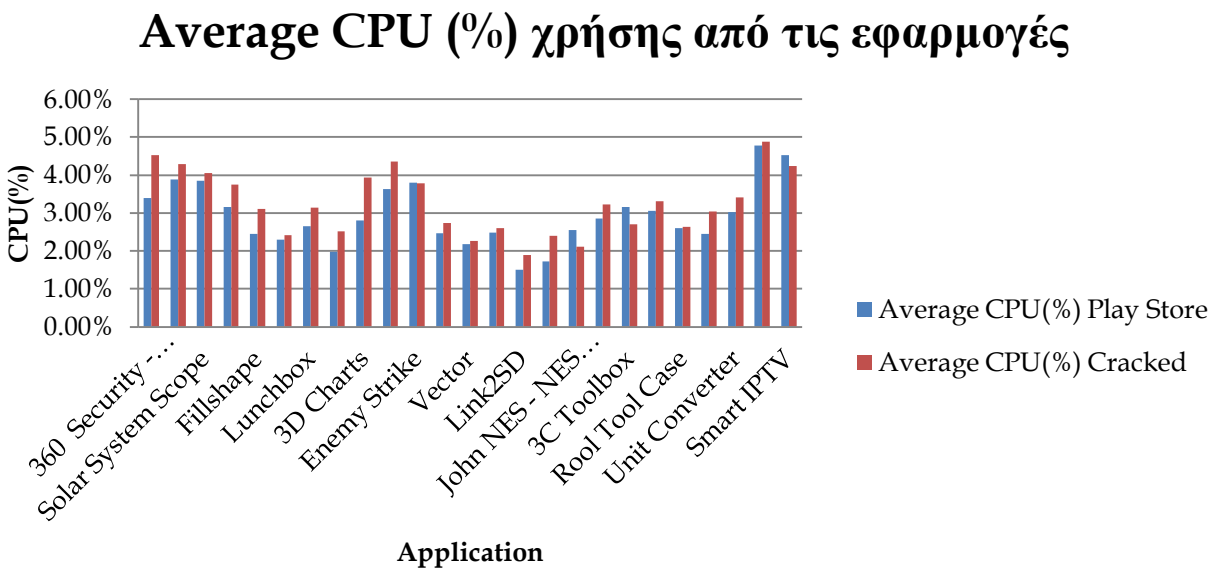


Γράφημα 4.14: Χρήση RAM (MB) από τις εφαρμογές στην έκδοση Marshmallow.

Κατά το Γράφημα 4.14 οι εφαρμογές που είναι Cracked σε σχέση με αυτές που προέρχονται από το Play Store και αναλύονται στην έκδοση Lollipop κάνουν μεγαλύτερη χρήση της RAM (MB).

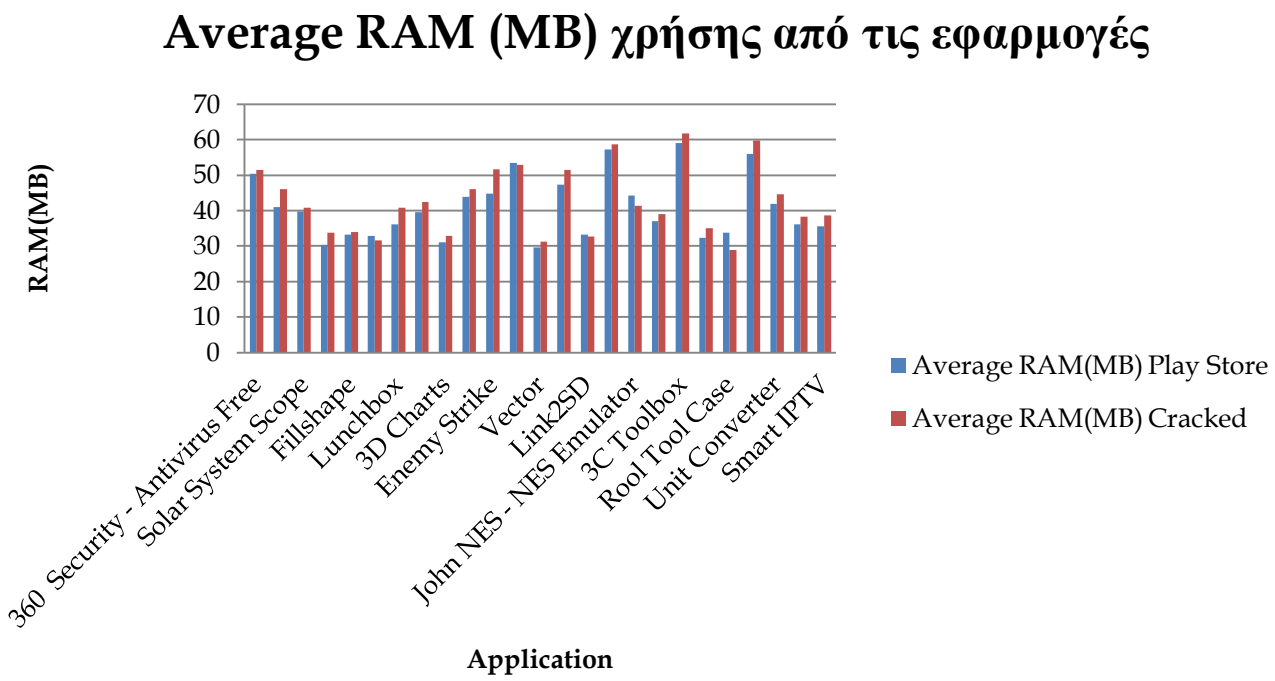
4.3.3 Ανάλυση μέσης χρήσης CPU και RAM

Τα δεδομένα του **Πίνακα 4.4** (Παράρτημα Γ) παρουσιάζουν τη μέση τιμή χρήσης της RAM και της CPU που κάνουν οι εφαρμογές που προέρχονται από το Play Store αλλά των εφαρμογών που είναι Cracked (από Third Party Markets), σύμφωνα με την ανάλυση που γίνεται και στις τρείς εκδόσεις Android. Έτσι τα γραφήματα 4.15, 4.16 παρουσιάζουν τη Μέση τιμή χρήσης CPU (%) και RAM (MB των Play Store και Cracked εφαρμογών.



Γράφημα 4.15: Μέση τιμή χρήσης CPU (%) των Play Store και Cracked εφαρμογών.

Παρατηρούμε (Γράφημα 4.15) ότι η μέση τιμή της CPU (%) για τις των εφαρμογές που προέρχονται από το Play Store σε σχέση με αυτές που είναι Cracked είναι μικρότερη με εξαίρεση την εφαρμογή Smart IPTV.



Γράφημα 4.16: Μέση τιμή χρήσης RAM (MB) των Play Store και Cracked εφαρμογών.

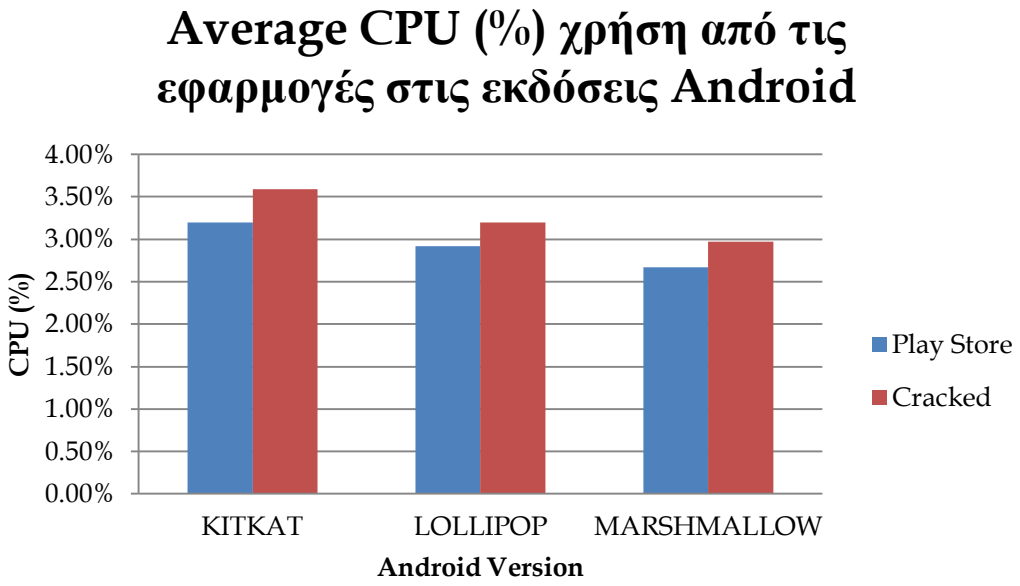
Φαίνεται (Γράφημα 4.16)ότι η μέση τιμή της RAM (MB) για τις των εφαρμογές που προέρχονται από το Play Store σε σχέση με αυτές που είναι Cracked είναι μικρότερη με εξαίρεση την εφαρμογή Smart IPTV.

Ο πιο κάτω πίνακας παρουσιάζει την μέση τιμή της ποσότητας χρήσης της RAM (MB) και του ποσοστού χρήσης της CPU(%) από τις κακόβουλες και καλοήθεις εφαρμογές στις τρείς εκδόσεις Android: KITKAT, Lollipop και Marshmallow. Ο πίνακας 4.5 μας βοηθά να αντιληφθούμε ποια είναι η χρήση της RAM(MB) και της CPU(%) που γίνεται από τις εφαρμογές ανάλογα με το είδος τους αν είναι κακόβουλες ή καλοήθεις.

Application	Average					
	RAM(MB)			CPU(%)		
	Android Version			Android Version		
	KITKAT	Lollipop	Marshmallow	KITKAT	Lollipop	Marshmallow
Play Store	42,60	40,79	39,03	3,20%	2,92%	2,67%
Cracked	44,52	42,76	40,68	3,59%	3,20%	2,97%

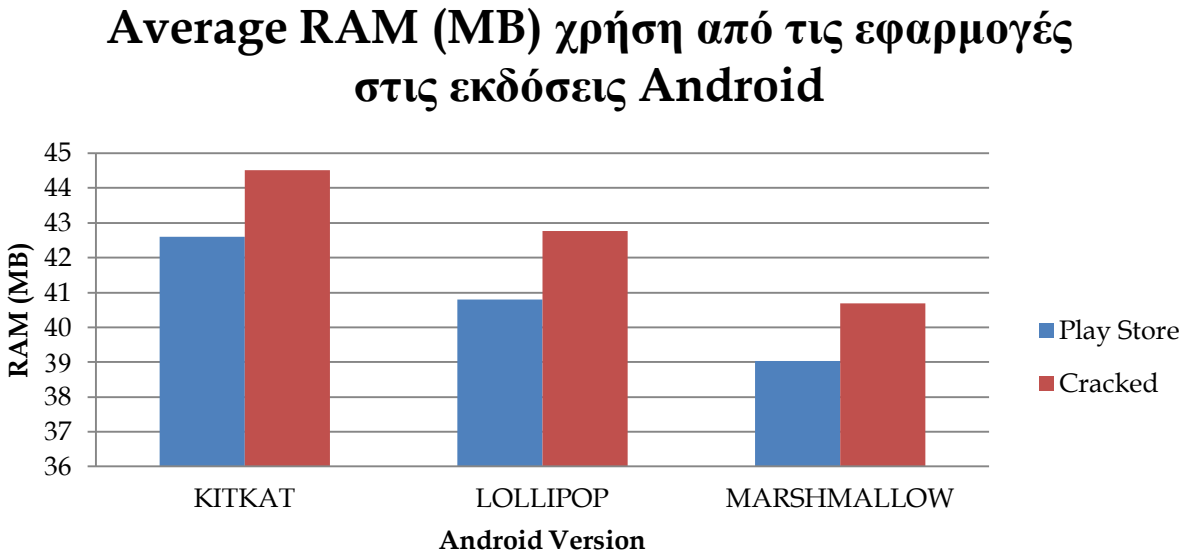
Πίνακας 4.5: Μέση τιμή χρήσης της RAM και CPU από τις εφαρμογές ανάλογα με την προέλευση τους και σύμφωνα με την έκδοση Android στην οποία τρέχουν.

Ακολουθούν οι γραφικές απεικονίσεις της μέσης χρήσης της CPU και RAM ανάλογα με την προέλευση των εφαρμογών.



Γράφημα 4.17: Μέσης χρήση της CPU σύμφωνα με την προέλευση των εφαρμογών.

Φαίνεται από το Γράφημα 4.17, ότι οι εφαρμογές που είναι Cracked κάνουν μεγαλύτερη χρήση της CPU (%) σε σχέση με τις εφαρμογές που προέρχονται από το Play Store.



Γράφημα 4.18: Μέσης χρήση της RAM σύμφωνα με την προέλευση των εφαρμογών.

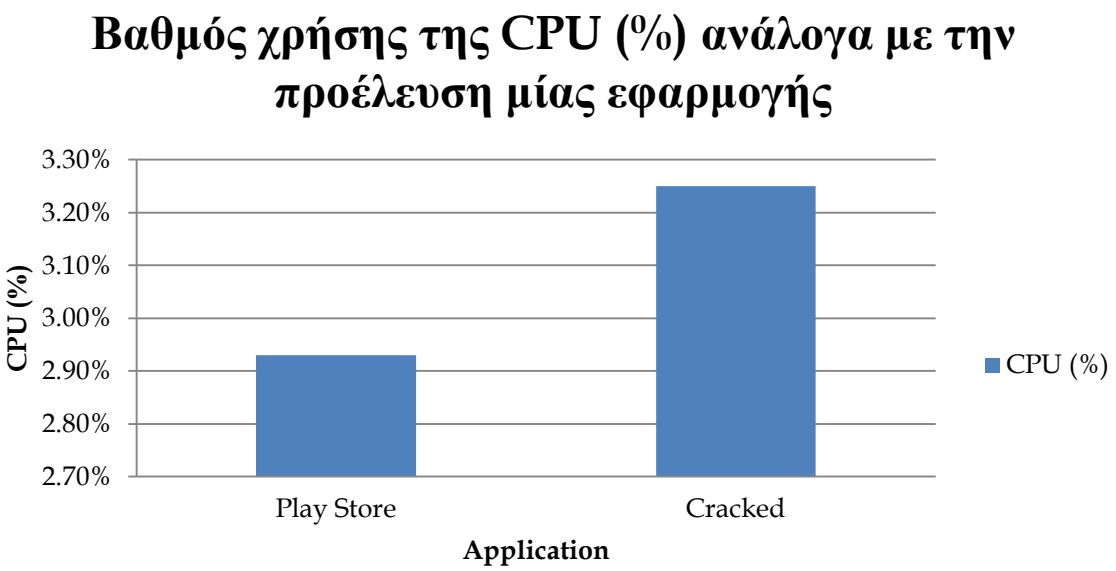
Είναι εμφανές στο Γράφημα 4.18, πως οι εφαρμογές που είναι Cracked κάνουν μεγαλύτερη χρήση της RAM (MB) σε σχέση με τις εφαρμογές που προέρχονται από το Play Store.

Ο πιο κάτω πίνακας παρουσιάζει τη μέση τιμή χρήσης της ποσότητας RAM(MB) και του ποσοστού χρήσης της CPU(%) των τριών εκδόσεων τόσο των Play Store και Cracked εφαρμογών, έτσι μπορούμε να αντιληφθούμε τη χρήση της RAM και της CPU που γίνεται από τις εφαρμογές ανάλογα με την προέλευση τους.

Application	Average	
	RAM(MB)	CPU (%)
Play Store	40,81	2.93%
Cracked	42,65	3.25%

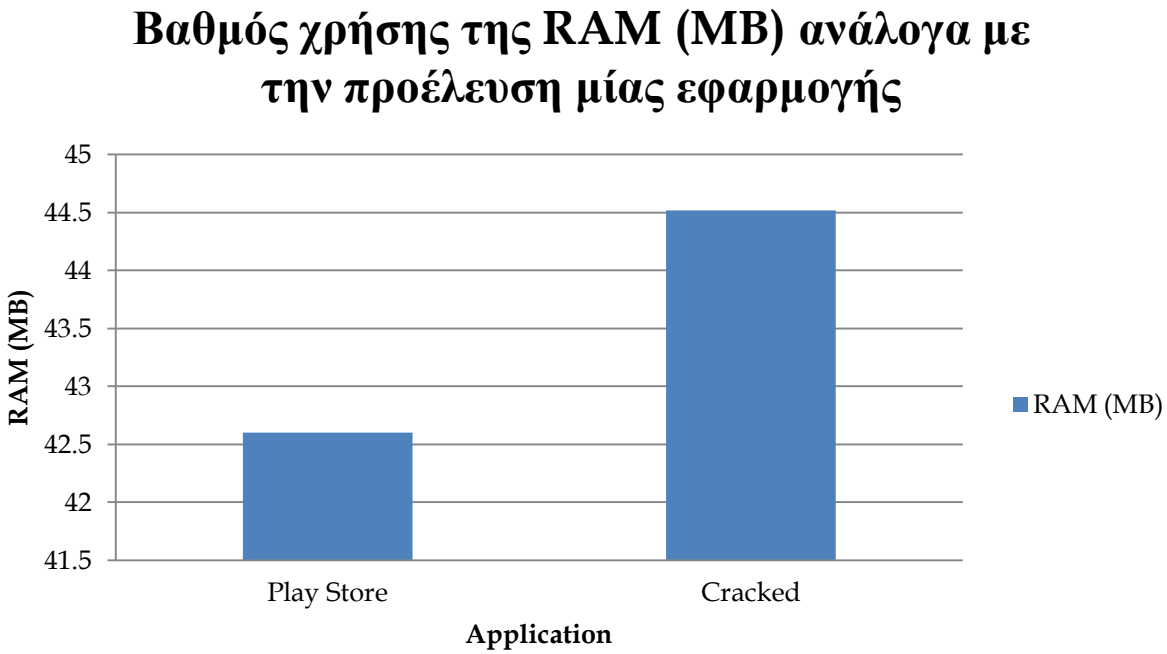
Πίνακας 4.6: Μέση χρήση της RAM(MB) και της CPU(%) των τριών εκδόσεων, τόσο των Play Store και Cracked εφαρμογών.

Οι επόμενες δύο γραφικές απεικονίσεις παρουσιάζουν τη μέση χρήση της RAM(MB) και της CPU(%) των τριών εκδόσεων, τόσο των Play Store και Cracked εφαρμογών.



Γράφημα 4.19: Μέση χρήση της CPU(%) των τριών εκδόσεων, τόσο των Play Store και Cracked εφαρμογών.

Παρατηρούμε πως η προέλευση της εφαρμογής παίζει ρόλο στο ποσοστό χρήσης της CPU, με τις Cracked εφαρμογές να κάνουν μεγαλύτερη χρήση(Γράφημα 4.19).



Γράφημα 4.20: Μέση χρήση της RAM(MB) των τριών εκδόσεων, τόσο των Play Store και Cracked εφαρμογών.

Είναι ξεκάθαρό σύμφωνα με το Γράφημα 4.20 πως η προέλευση της εφαρμογής παίζει ρόλο στη χρήσης της RAM, με τις Cracked εφαρμογές να κάνουν μεγαλύτερη χρήση.

4.4 Ανάλυση του Network Traffic

Το Network Traffic ή το Data Traffic είναι η ποσότητα δεδομένων που μετακινούνται σε ένα δίκτυο σε ένα δεδομένο χρονικό σημείο [106]. Τα δεδομένα δικτύου σε δίκτυα υπολογιστών είναι κυρίως ενθυλακωμένα σε πακέτα δικτύου, τα οποία παρέχουν το φορτίο στο δίκτυο. Η κυκλοφορία δικτύου είναι το κύριο συστατικό στοιχείο για τη μέτρηση της κυκλοφορίας δικτύου, τον έλεγχο της κυκλοφορίας δικτύου και την προσομοίωση.

Η σωστή ανάλυση της κυκλοφορίας δικτύου παρέχει στον οργανισμό την ασφάλεια του δικτύου ως πλεονέκτημα, η ασυνήθιστη ποσότητα κίνησης σε ένα δίκτυο είναι ένα πιθανό σημάδι μιας

επίθεσης. Οι αναφορές κυκλοφορίας δικτύου παρέχουν πολύτιμες πληροφορίες για την πρόληψη τέτοιων επιθέσεων [106].

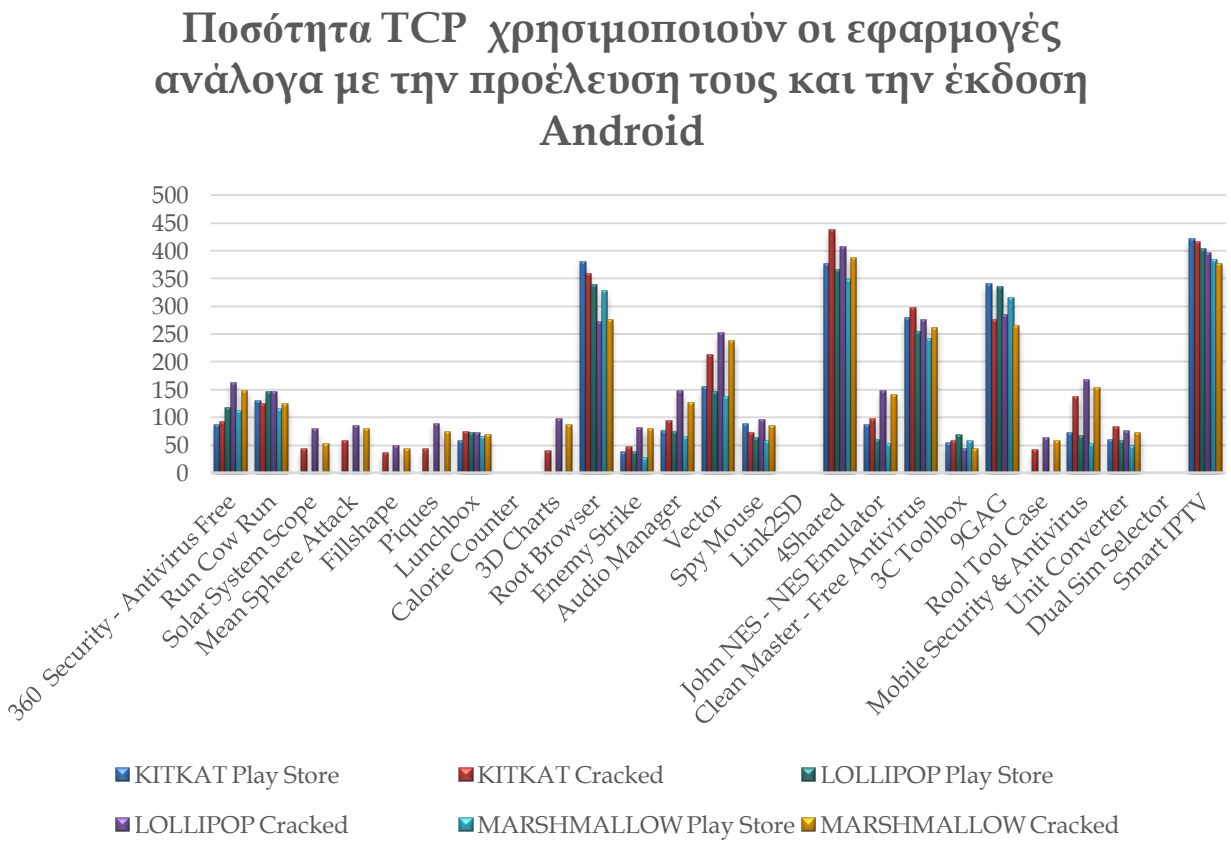
Ο όγκος κυκλοφορίας είναι ένα μέτρο της συνολικής εργασίας που πραγματοποιείται από έναν πόρο ή μια εγκατάσταση, συνήθως πάνω από 24 ώρες, και μετράται σε μονάδες erlang-hour. Ορίζεται ως το προϊόν της μέσης έντασης της κυκλοφορίας και της χρονικής περιόδου της μελέτης. Ένας όγκος κυκλοφορίας μιας ώρας ανά δευτερόλεπτο μπορεί να προκληθεί από δύο κυκλώματα που καταλαμβάνονται συνεχώς επί μισή ώρα ή από ένα κύκλωμα που είναι μισό κατειλημμένο (0.5 erlang) για μια περίοδο δύο ωρών. Οι τηλεπικοινωνιακοί φορείς ενδιαφέρονται ζωηρά για τον όγκο της κυκλοφορίας, διότι υπαγορεύουν άμεσα τα έσοδά τους.

Traffic volume = Traffic intensity × time

Επιλέχθηκαν να μελετηθούν τα TCP και HTTP ports που χρησιμοποιούν οι εφαρμογές που αναλύθηκαν. Στη συνέχεια επεξηγείται ο ρόλος των δύο πρωτοκόλλων.
Το TCP/IP (Transmission Control Protocol/Internet Protocol στα ελληνικά αποδίδεται ως Πρωτόκολλο Ελέγχου Μετάδοσης ή Πρωτόκολλο Διαδικτύου) [107] είναι συλλογή πρωτοκόλλων επικοινωνίας που βασίζεται το Διαδίκτυο αλλά και μεγάλο ποσοστό των εμπορικών δικτύων. Σύμφωνα με το μοντέλου OSI το TCP/IP ανήκει στο επίπεδο Μεταφορά (Transport Layer), που το καθιστά υπεύθυνο για την μεταφορά μηνυμάτων, ανεξαρτήτως του υποκείμενου δικτύου, με έλεγχο σφαλμάτων (Error Control), κατάτμηση (fragmentation) και ρύθμιση ροής (Flow Control). Το HTTP (HyperText Transfer Protocol) στα ελληνικά αποδίδεται ως Πρωτόκολλο Μεταφοράς Υπερκειμένου και αποτελεί ένα πρωτόκολλο επικοινωνίας [108]. Είναι το κύριο πρωτόκολλο που χρησιμοποιείται στους φυλλομετρητές του Παγκοσμίου Ιστού για να μεταφέρει δεδομένα ανάμεσα σε έναν διακομιστή (server) και έναν πελάτη (client). Η διαδικασία που ακολουθούσε το αρχικό πρωτόκολλο ήταν η εξής: Σύνδεση στον εξυπηρετητή, Ερώτηση προς τον εξυπηρετητή και τέλος Απάντηση από τον εξυπηρετητή. Σήμερα χρησιμοποιεί πολύ περισσότερα χαρακτηριστικά τα οποία παρέχουν ακόμα και τη δυνατότητα στο πρόγραμμα-πελάτης να στέλνει δεδομένα στον εξυπηρετητή.

Στον **Πίνακας 4.7** (Παράρτημα Γ) φαίνεται ο αριθμός των TCP και HTTP ports που χρησιμοποιούν οι εφαρμογές που προέρχονται από το Play Store και αυτές που είναι Cracked στις εκδόσεις Android: KitKat, Lollipop, Marshmallow. Ακολουθούν γραφήματα που παρουσιάζουν την

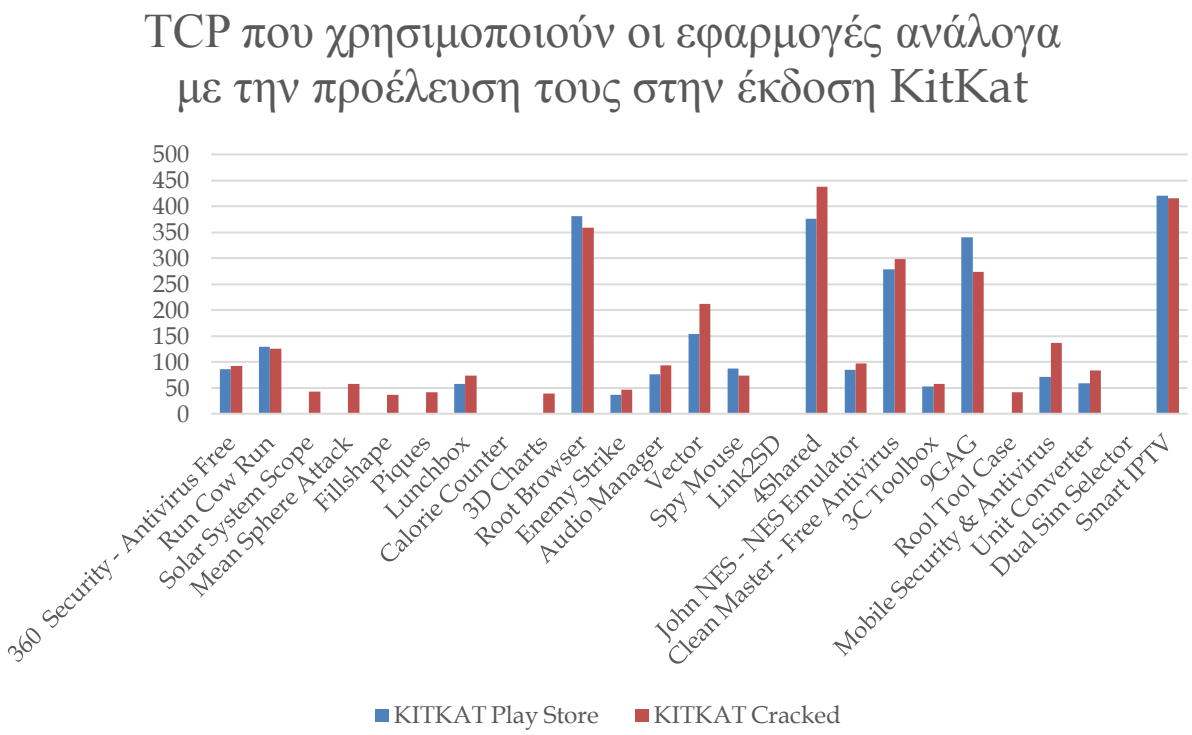
ποσότητα των TCP ports που κάνουν οι εφαρμογές ανάλογα με την προέλευση τους ανά έκδοση Android.



Γράφημα 4.21: Πόσα TCP χρησιμοποιούν οι εφαρμογές ανάλογα με την προέλευση τους και την έκδοση Android.

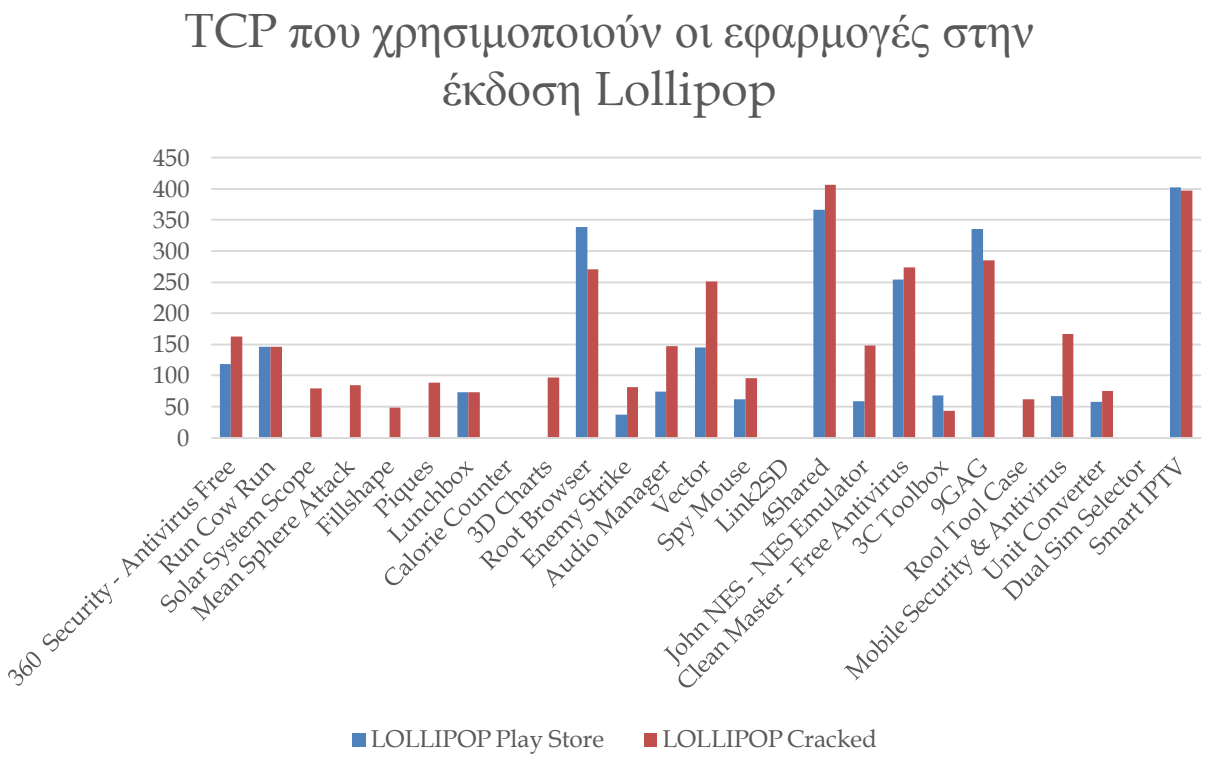
Στο Γράφημα 4.21 φαίνονται ποια TCP χρησιμοποιούν οι εφαρμογές ανάλογα με την προέλευση τους στις εκδόσεις Android KitKat, Lollipop, Marshmallow. Είναι εμφανές ότι κάθε φορά χρησιμοποιούν διαφορετικό αριθμό.

Τα επόμενα τρία γραφήματα παρουσιάζουν την ποσότητα TCP που χρησιμοποιούν οι εφαρμογές ανάλογα με την προέλευση τους σε κάθε έκδοση Android



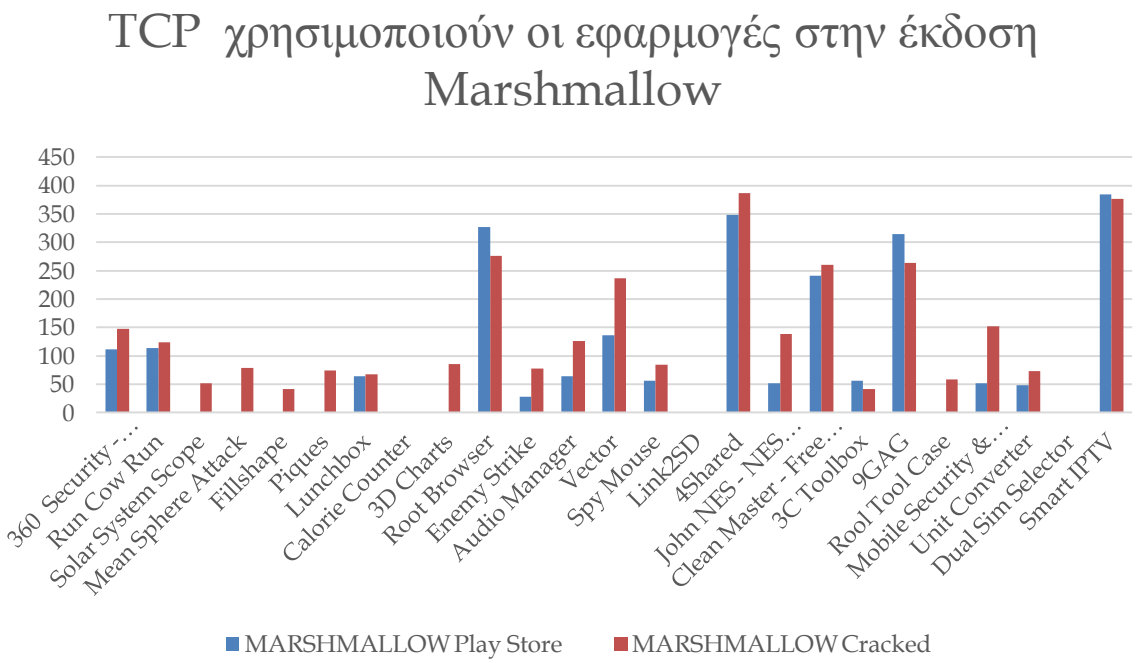
Γράφημα 4.22: TCP που χρησιμοποιούν οι εφαρμογές ανάλογα με την προέλευση τους στην έκδοση KitKat.

Στο Γράφημα 4.22 φαίνονται τα TCP χρησιμοποιούν οι εφαρμογές ανάλογα με την προέλευση τους στην έκδοση Android KitKat, δεν είναι απαραίτητο οι Cracked να χρησιμοποιούν περισσότερα TCP, αλλά παρατηρούμε οι εφαρμογές Solar System Scope, Mean Sphere Attack, Fillshape, Piques, 3D Charts και Rool Tool Case που είναι Cracked να χρησιμοποιούν TCP ενώ αυτές του Play Store όχι.



Γράφημα 4.23: TCP που χρησιμοποιούν οι εφαρμογές ανάλογα με την προέλευση τους στην έκδοση Android Lollipop.

Στο πιο πάνω (Γράφημα 4.23) φαίνονται τα TCP χρησιμοποιούν οι εφαρμογές ανάλογα με την προέλευση τους στην έκδοση Android Lollipop, δεν είναι απαραίτητο οι Cracked να χρησιμοποιούν περισσότερα TCP, αλλά παρατηρούμε οι εφαρμογές Solar System Scope, Mean Sphere Attack, Fillshape, Piques, 3D Charts και Rool Tool Case που είναι Cracked να χρησιμοποιούν TCP ενώ αυτές του Play Store όχι.

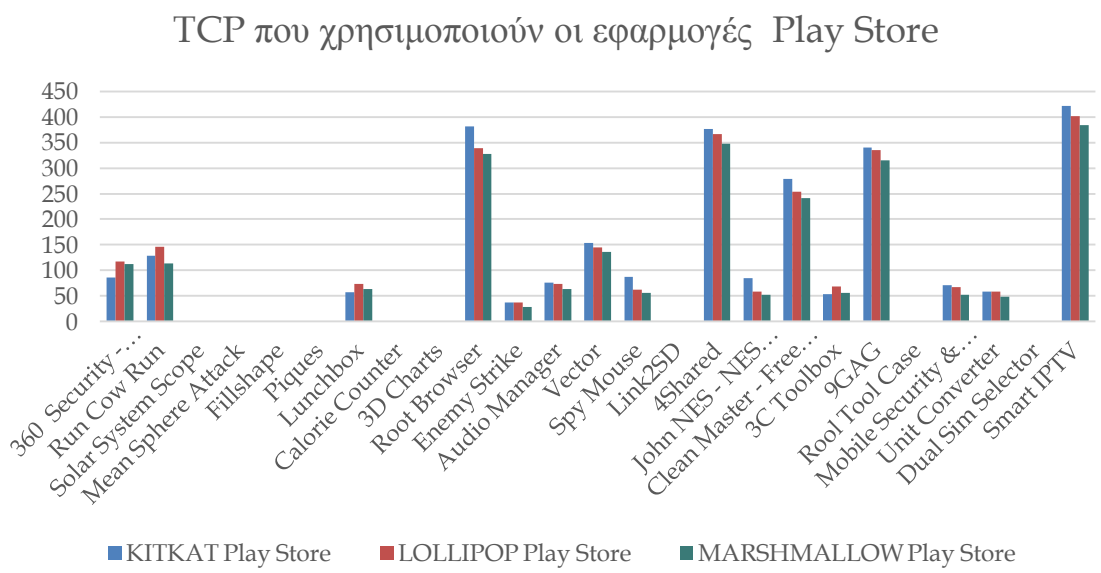


Γράφημα 4.24: TCP που χρησιμοποιούν οι εφαρμογές ανάλογα με την προέλευση τους στην έκδοση Android Marshmallow.

Στο πιο πάνω (Γράφημα 4.24) φαίνονται τα TCP χρησιμοποιούν οι εφαρμογές ανάλογα με την προέλευση τους στην έκδοση Android Marshmallow, δεν είναι απαραίτητο οι Cracked να χρησιμοποιούν περισσότερα TCP, αλλά παρατηρούμε οι εφαρμογές Solar System Scope, Mean Sphere Attack, Fillshape, Piques, 3D Charts και Rool Tool Case που είναι Cracked να χρησιμοποιούν TCP ενώ αυτές του Play Store όχι.

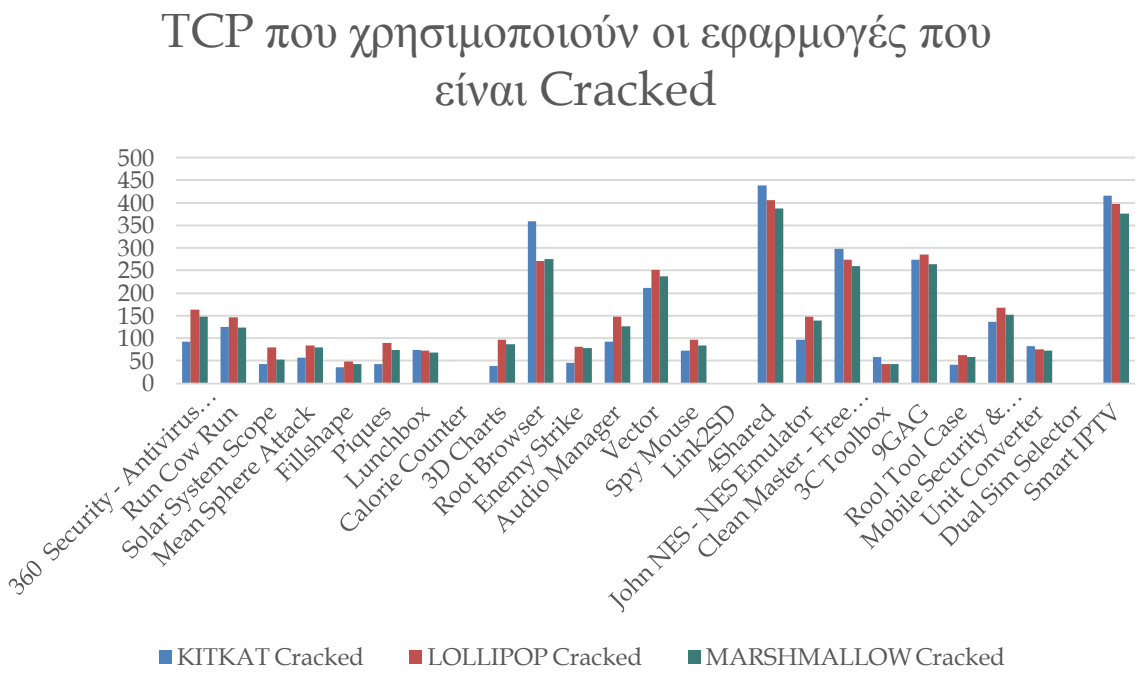
Οι εφαρμογές Solar System Scope, Mean Sphere Attack, Fillshape, Piques, 3D Charts και Rool Tool Case που είναι Cracked σε όλες τις εκδόσεις χρησιμοποιούν περισσότερα TCP.

Πιο κάτω ακολουθούν δύο γραφήματα που παρουσιάζουν την ποσότητα των TCP που χρησιμοποιούν οι εφαρμογές Play Store και Cracked στις τρείς εκδόσεις.



Γράφημα 4.25: TCP χρησιμοποιούν οι εφαρμογές Play Store στις τρείς εκδόσεις.

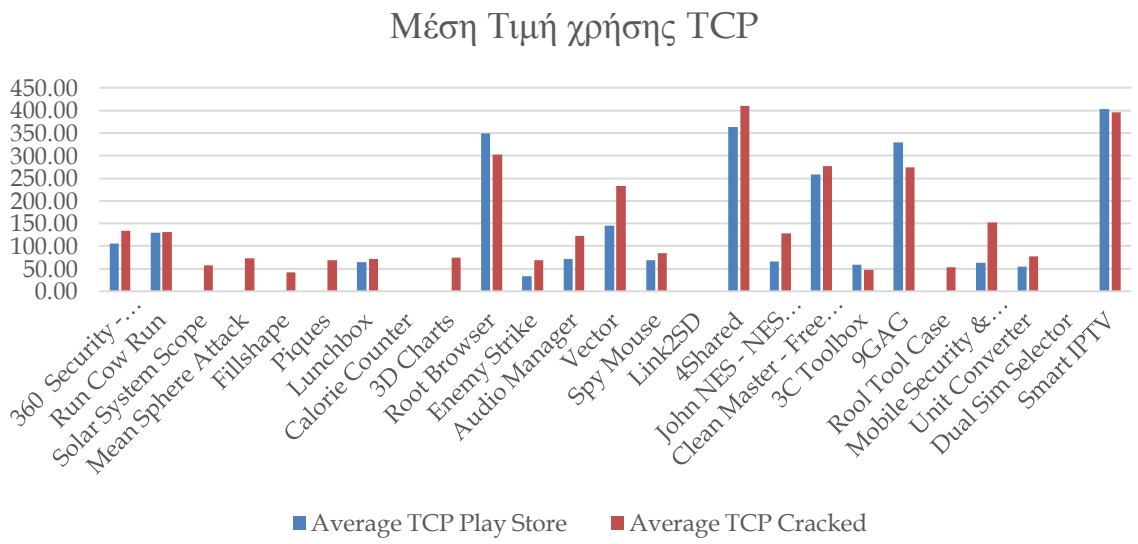
Κάθε φορά τα TCP χρησιμοποιούν οι εφαρμογές από το Play Store στις τρείς εκδόσεις Android είναι διαφορετικά εκτός από τις εφαρμογές Solar System Scope, Mean Sphere Attack, Fillshape, Piques, Calorie Counter, 3D Charts, Link2SD, Rool Tool Case και Dual Sim Selector όπου δεν χρησιμοποιούν TCP (Γράφημα 4.25).



Γράφημα 4.26: TCP χρησιμοποιούν οι εφαρμογές Cracked στις τρείς εκδόσεις.

Τα TCP σύμφωνα με το γράφημα που χρησιμοποιούν οι εφαρμογές που είναι Cracked στις τρείς εκδόσεις Android είναι διαφορετικά εκτός από τις εφαρμογές Calorie Counter, Link2SD και Dual Sim Selector (Γράφημα 4.26).

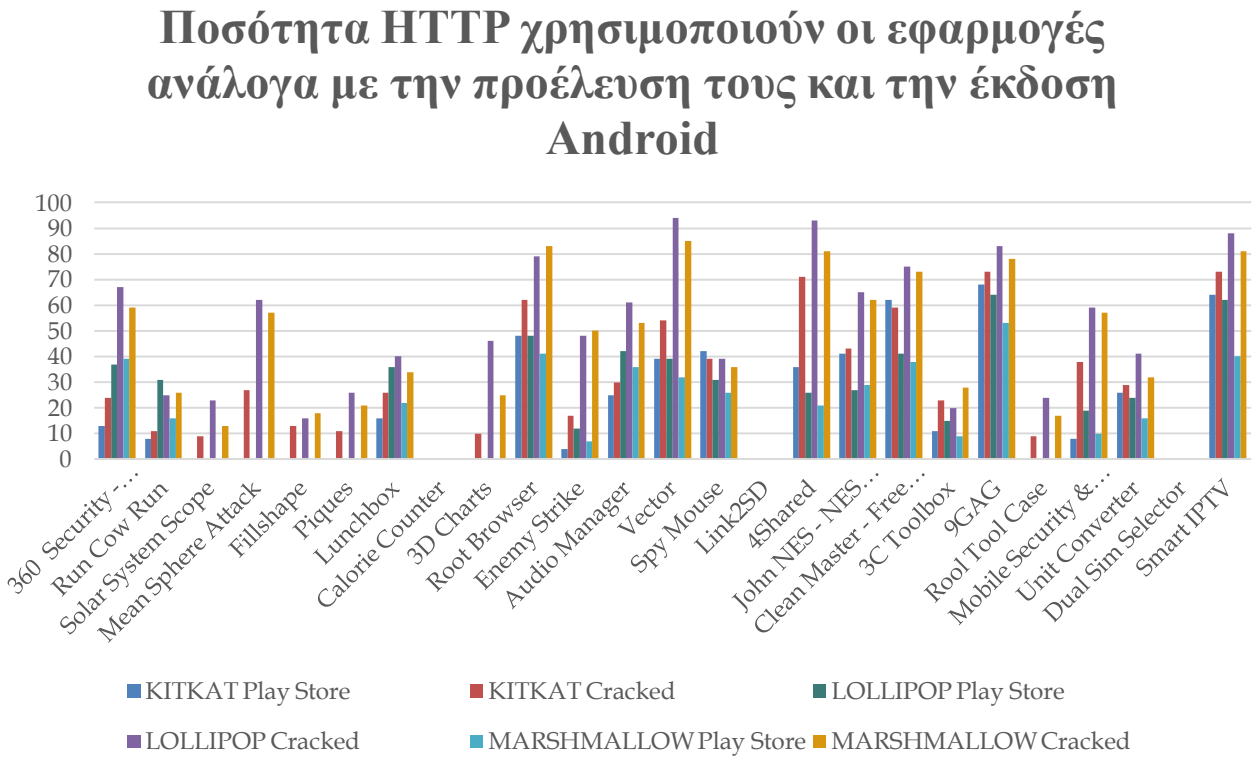
Στο πιο κάνω Γράφημα 4.27 φαίνεται η μέση τιμή χρήσης του TCP ανάλογα με τη προέλευση της εφαρμογής σε όλες τις εκδόσεις Android.



Γράφημα 4.27: Μέση τιμή χρήσης του TCP ανάλογα με τη προέλευση της εφαρμογής

Το Γράφημα 4.27 μας δείχνει πως οι εφαρμογές Cracked χρησιμοποιούν περισσότερο το TCP port.

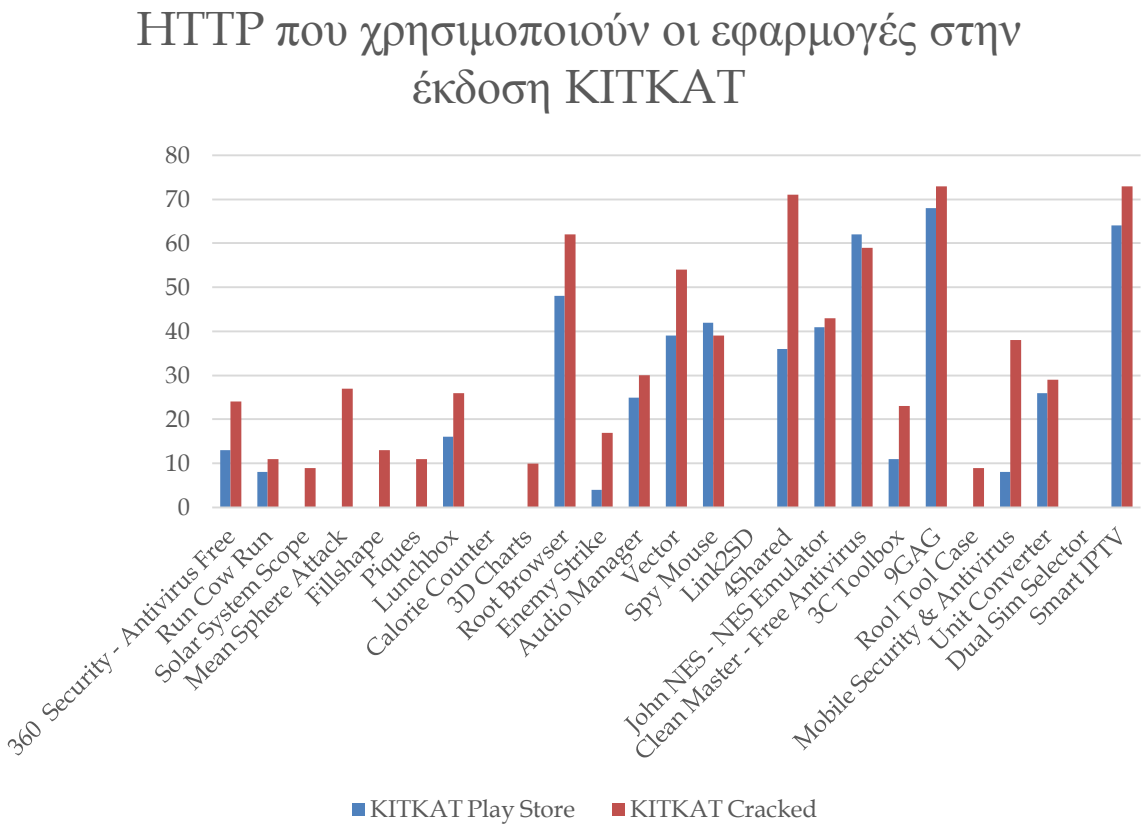
Ακολουθούν γραφήματα που παρουσιάζουν την μεταβολή χρήσης του HTTP ports που κάνουν οι εφαρμογές ανάλογα με την προέλευση τους ανά έκδοση Android.



Γράφημα 4.28: HTTP χρησιμοποιούν οι εφαρμογές ανάλογα με την προέλευση τους και την έκδοση Android.

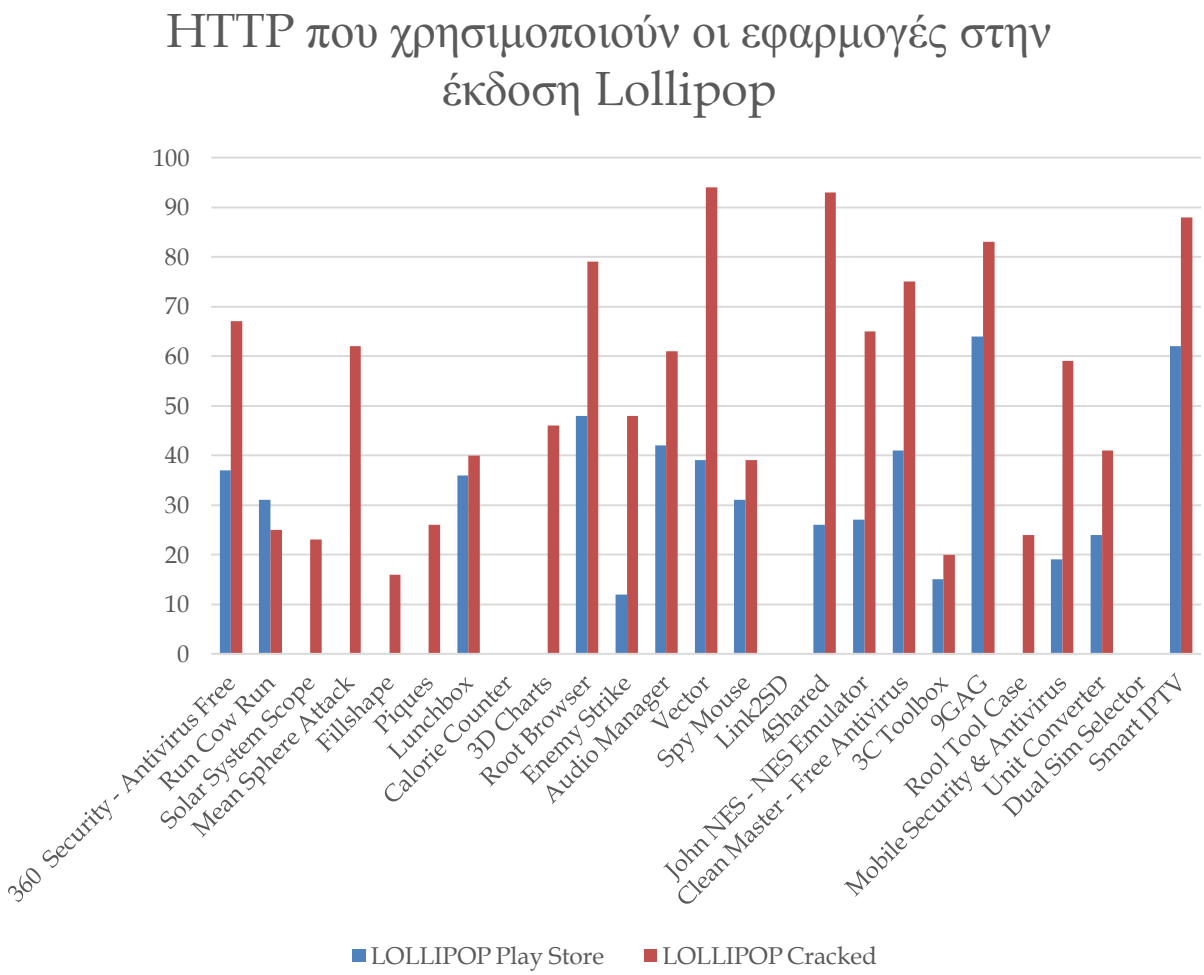
Στο πιο πάνω Γράφημα 4.28 φαίνονται πόσα HTTP χρησιμοποιούν οι εφαρμογές ανάλογα με την προέλευση τους στις εκδόσεις Android KitKat, Lollipop, Marshmallow. Είναι εμφανές ότι κάθε φορά χρησιμοποιούν διαφορετικό port.

Τα επόμενα τρία γραφήματα παρουσιάζουν την ποσότητα των HTTP που χρησιμοποιούν οι εφαρμογές ανάλογα με την προέλευση τους σε κάθε έκδοση Android



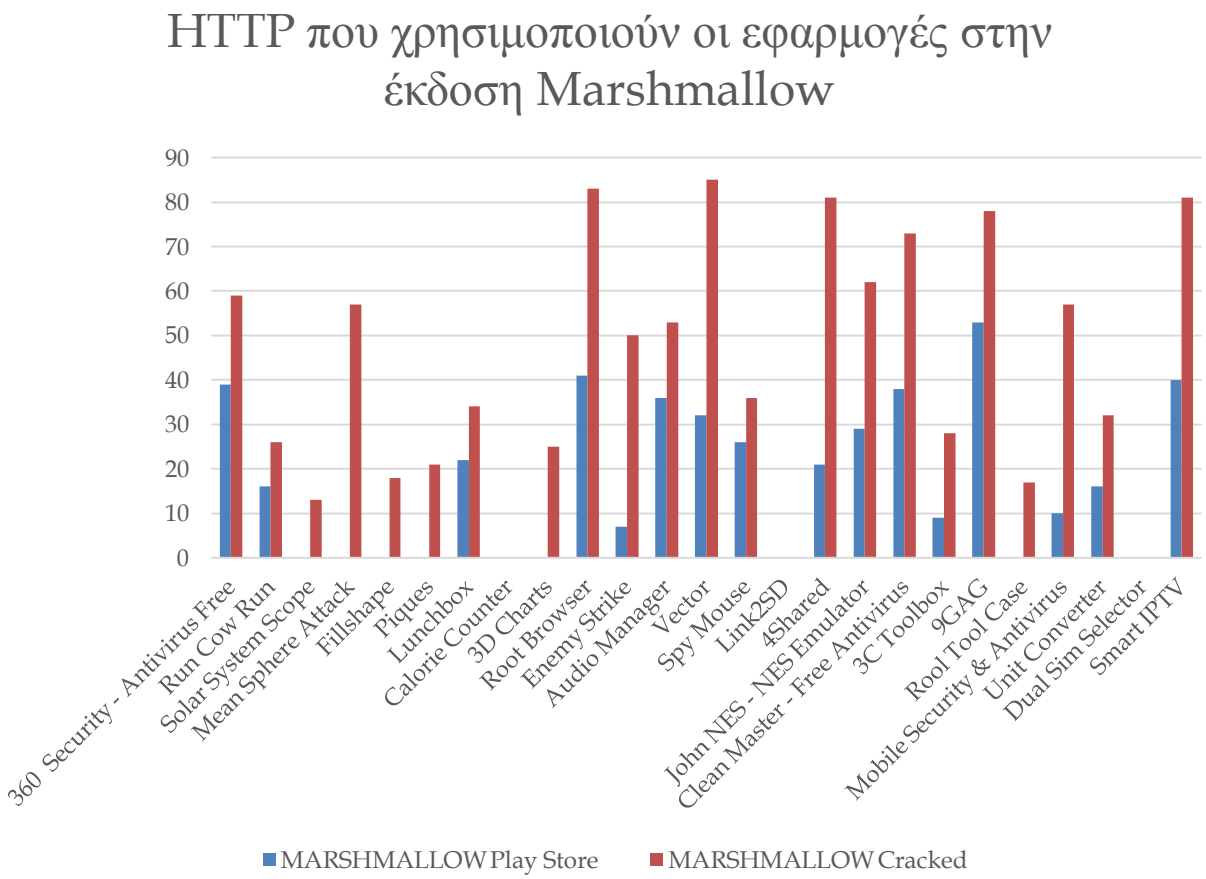
Γράφημα 4.29: HTTP που χρησιμοποιούν οι εφαρμογές ανάλογα με την προέλευση τους στην έκδοση KitKat.

Ο αριθμός των HTTP που χρησιμοποιούν οι εφαρμογές ανάλογα με την προέλευση τους στην έκδοση Android KitKat κάθε φορά είναι διαφορετικός (Γράφημα 4.29) εκτός από τις εφαρμογές Calorie Counter, Link2SD και Dual Sim Selector, που είναι μηδαμινός. Οι Cracked τείνουν να χρησιμοποιούν περισσότερα .



Γράφημα 4.30: HTTP που χρησιμοποιούν οι εφαρμογές ανάλογα με την προέλευση τους στην έκδοση Lollipop.

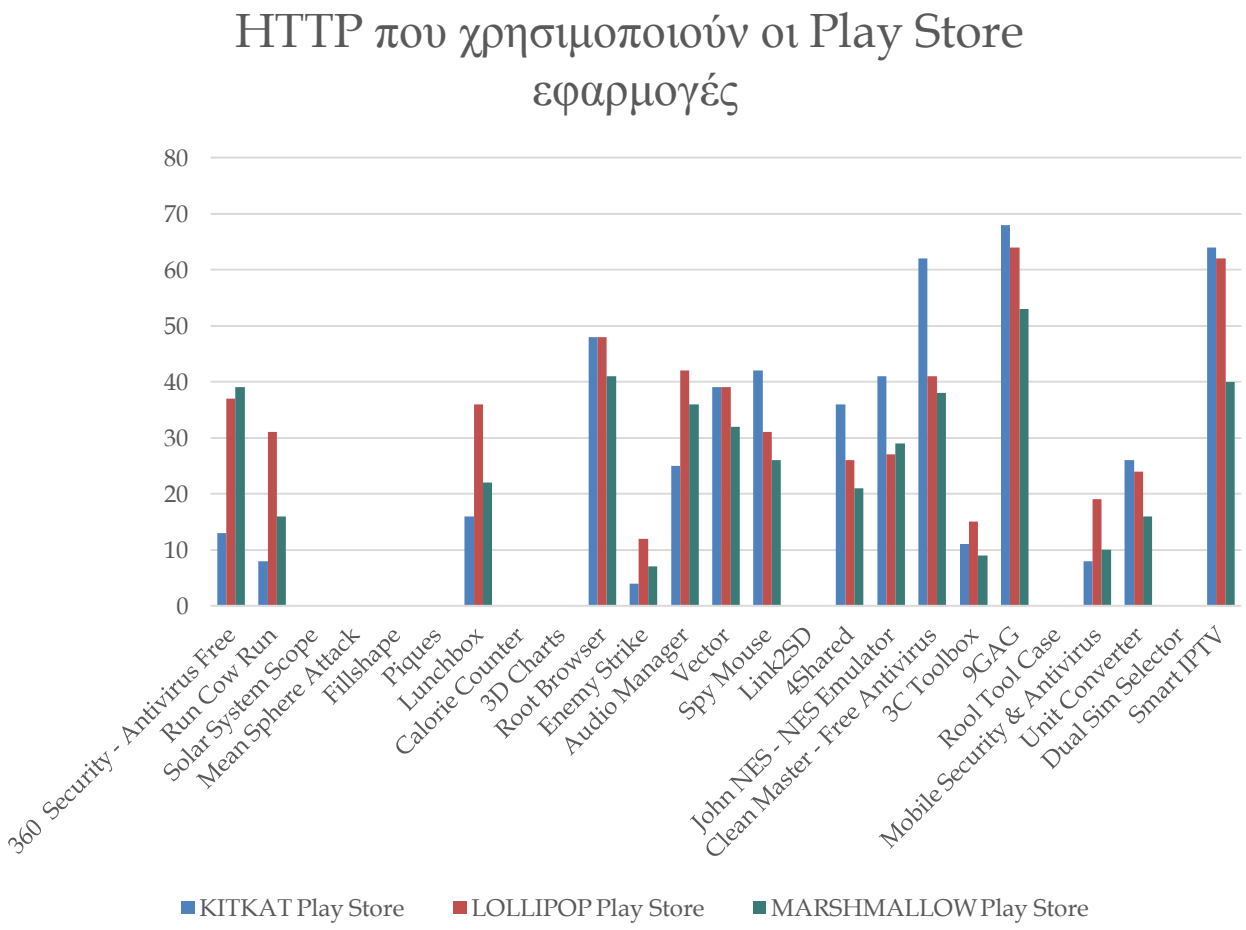
Τα HTTP που χρησιμοποιούν οι εφαρμογές ανάλογα με την προέλευση τους στην έκδοση Android Lollipop κάθε φορά είναι διαφορετικά (Γράφημα 4.30). Εκτός από τις εφαρμογές Calorie Counter, Link2SD και Dual Sim Selector που δεν παρουσιάζουν χρήση και παρατηρούμε οι Cracked να σημειώνουν αυξημένη χρήση συγκριτικά με τις εφαρμογές του Play Store.



Γράφημα 4.31: HTTP ports που χρησιμοποιούν οι εφαρμογές ανάλογα με την προέλευση τους στην έκδοση Marshmallow.

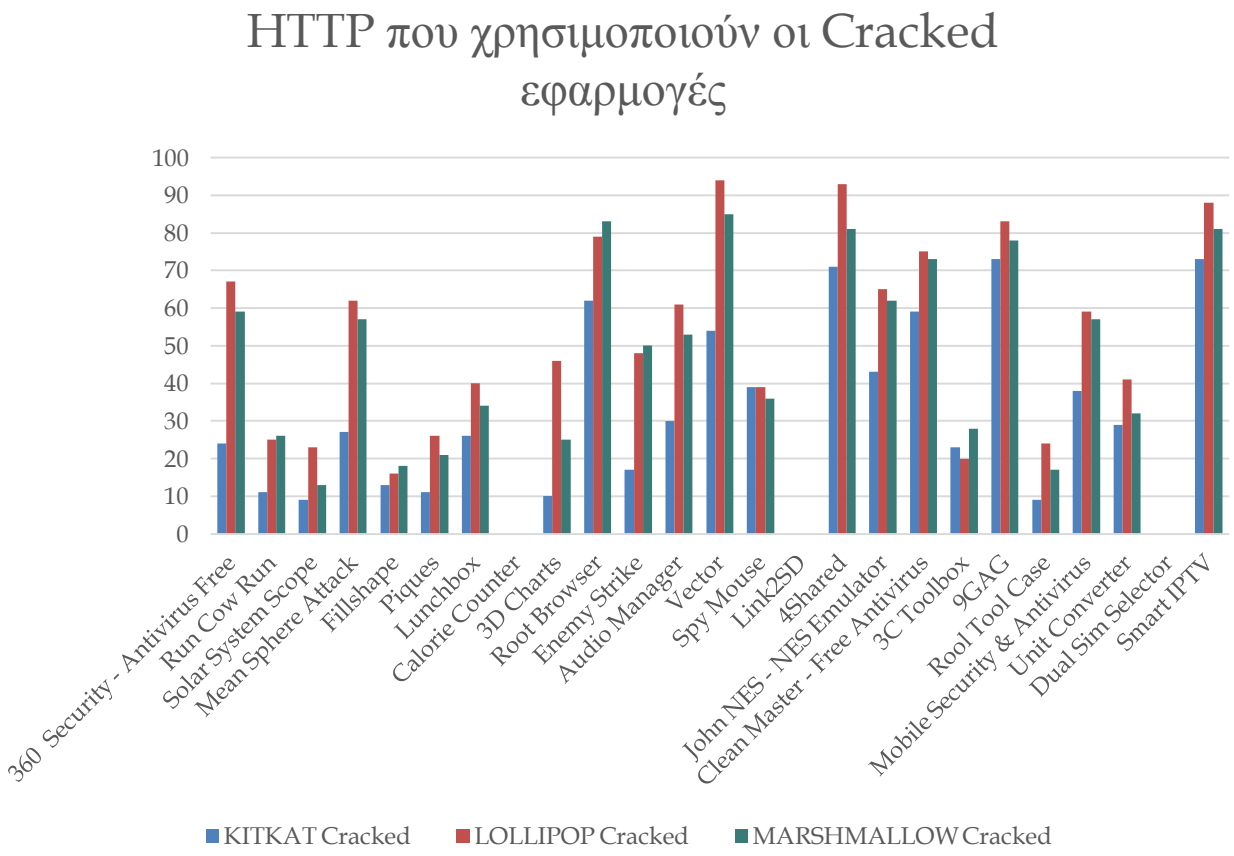
Τα HTTP που χρησιμοποιούν οι εφαρμογές Cracked είναι περισσότερα από των Play Store στην έκδοση Android Marshmallow κάθε φορά είναι διαφορετικά (Γράφημα 4.31), εκτός από τις εφαρμογές Calorie Counter, Link2SD και Dual Sim Selector που κάνουν μηδενική χρήση.

Πιο κάτω ακολουθούν δύο γραφήματα που παρουσιάζουν την ποσότητα των HTTP που χρησιμοποιούν οι εφαρμογές Play Store και Cracked στις τρείς εκδόσεις.



Γράφημα 4.32: HTTP που χρησιμοποιούν οι εφαρμογές Play Store στις τρείς εκδόσεις.

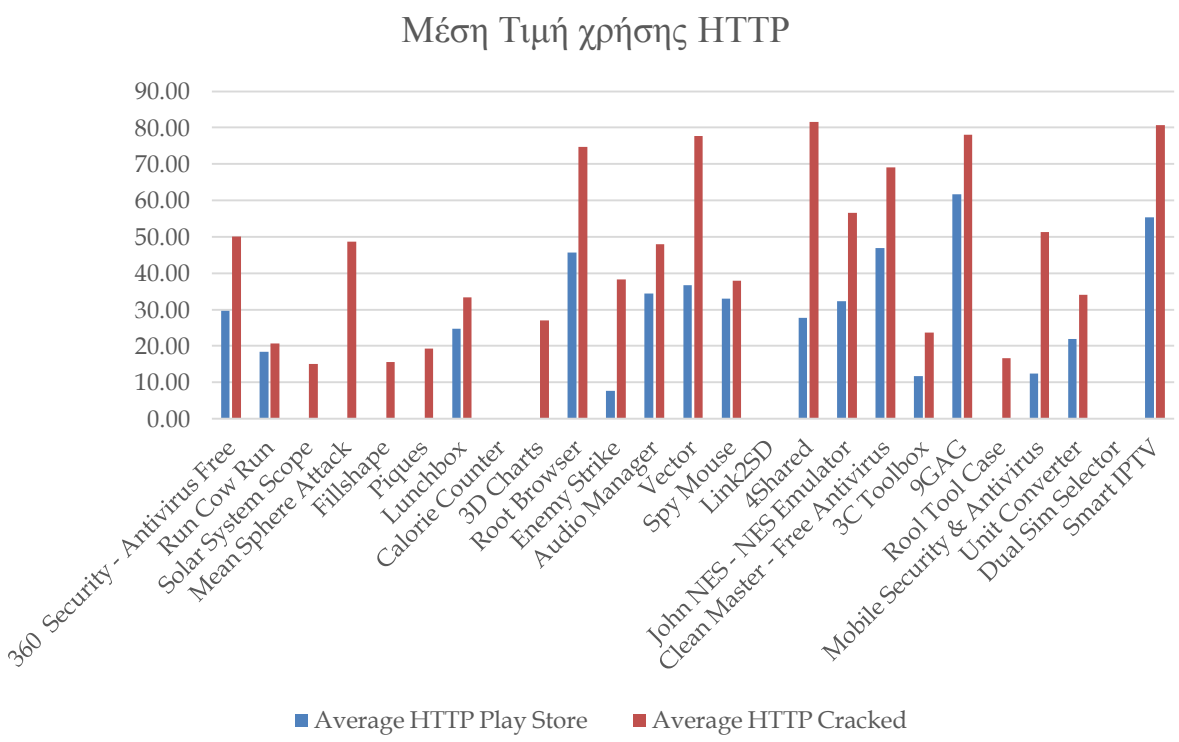
Κάθε φορά τα HTTP ports που χρησιμοποιούν οι εφαρμογές από το Play Store στις τρείς εκδόσεις Android είναι διαφορετικά εκτός από τις εφαρμογές Solar System Scope, Mean Sphere Attack, Fillshape, Piques, Calorie Counter, 3D Charts, Link2SD, Rool Tool Case και Dual Sim Selector (Γράφημα 4.32).



Γράφημα 4.33: HTTP που χρησιμοποιούν οι εφαρμογές Cracked στις τρείς εκδόσεις.

Κάθε φορά τα HTTP ports που χρησιμοποιούν οι εφαρμογές Cracked στις τρείς εκδόσεις Android είναι διαφορετικά εκτός από τις εφαρμογές Calorie Counter, Link2SD και Dual Sim Selector (Γράφημα 4.33).

Στο πιο κάνω Γράφημα 4.34 φαίνεται η μέση τιμή χρήσης του HTTP ανάλογα με τη προέλευση της εφαρμογής σε όλες τις εκδόσεις Android.



Γράφημα 4.34: Μέση τιμή χρήσης του HTTP ανάλογα με τη προέλευση της εφαρμογής

Γενική παρατήρηση είναι πως οι περισσότερες Play Store και Cracked εφαρμογές και στις τρείς υπό ανάλυση εκδόσεις Android χρησιμοποιούν διαφορετικά αριθμό TCP και το HTTP. Επιπρόσθετα παρατηρούμε ότι οι Cracked εφαρμογές ανεξαρτήτως έκδοσης κάνουν μεγαλύτερη χρήση των TCP και HTTP ports συγκριτικά με τις εφαρμογές που προέρχονται από το Play Store. Ακόμη αξίζει να σημειωθεί πως ενώ οι Play Store εφαρμογές μπορεί να μην σημειώνουν καμία χρήση των TCP και HTTP σε σχέση με τις όμοιες τους τις Cracked που όμως σημειώνουν όπως συμβαίνει στις εφαρμογές Solar System Scope, Mean Sphere Attack, Fillshape, Piques, 3D Charts, Rool Tool Case.

Κεφάλαιο 5
Επίλογος

5.1 Συμπεράσματα

Πασιφανές για όλους μας καθημερινά είναι πως η τεχνολογία έχει γίνει μία καθημερινή συνήθεια με τις θετικές αλλά την ίδια στιγμή και τις αρνητικές πτυχές της. Ειδικότερα στην εποχή μας αυτό είναι εμφανές με την χρήση της στα smartphones που αποτελούν προέκταση του εαυτού μας μέσω των οποίων πραγματοποιούμε πολλές προσωπικές δραστηριότητες και επιχειρηματικές συναλλαγές. Με αποτέλεσμα να διατρέχουμε σοβαρούς κινδύνους με την εγκατάσταση διαφόρων εφαρμογών σε αυτά και έτσι να πλήττεται η εμπιστευτικότητα, η ακεραιότητα και η διαθεσιμότητα. Το Android malware εξαπλώνεται με ιλιγγιώδη ταχύτητα, πράγμα που σηματοδοτεί το γεγονός πως αποτελεί πραγματικά επικίνδυνη απειλή για εκατομμύρια χρήστες ανά το παγκόσμιο. Το Android malware ανιχνεύεται όλο και πιο δύσκολα από το μέσο χρήστη smartphone και προκύπτει από Android εφαρμογές.

Σημάδια που δείχνουν ότι το Android Smartphone έχει μολυνθεί με κακόβουλο λογισμικό και αποτελούν ενδείξεις αποκάλυψης της παρουσία του κακόβουλου λογισμικού είναι ο Bad Battery

Life: Οι Android χρήστες που δεν εκτελούν πολλές δραστηριότητες με φίλτρα έχουν αίσθηση της διάρκειας της μπαταρίας τους. Το malware επηρεάζει σε μεγάλο βαθμό τη ζωή της μπαταρίας. Οι Dropped Calls and Disruptions: Το mobile malware μπορεί να επηρεάσει τις τρέχουσες ή εισερχόμενες κλήσεις. Οι κλήσεις μπορεί να πέσουν ή μπορεί να υπάρξουν διαταραχές κατά τη διάρκεια μίας συνομιλίας, πράγμα που υποδεικνύει την ύπαρξη malware. Inordinately Large Phone Bills: Το Android malware μπορεί συχνά να μολύνει τις συσκευές και να ξεκινήσει την αποστολή sms σε αριθμούς με υψηλό κόστος. Data Plan Spikes: Malware που διακινεί παράνομα δεδομένα από τη συσκευή σε τρίτους, μπορεί να ανιχνευθεί με την εξέταση του Data Plan λογαριασμού. Clogged Performance: Ανάλογα με τις προδιαγραφές του υλικού της συσκευής η μόλυνση κακόβουλου λογισμικού μπορεί να προκαλέσει σοβαρά προβλήματα στην απόδοση, όταν προσπαθεί κάποιος να διαβάσει, να γράψει ή να εκπέμψει δεδομένα από το smartphone. Αυτή η μόλυνση είναι οικεία με αυτή της μόλυνσης ενός υπολογιστή με malware. Η επανεκκίνηση μίας συσκευής πολλές φορές την ημέρα οφείλεται στην εκτέλεση κακόβουλου λογισμικού στο παρασκήνιο, πράγμα που σημαίνει κατανάλωση πάρα πολλής επεξεργαστικής ισχύς ώστε να λειτουργούν σωστά οι διάφορες εφαρμογές. Επομένως η γνώση της πραγματικής πρόθεσης των εφαρμογών που εγκαθίστανται στις συσκευές κρίνεται αναγκαία, οι στόχοι μίας καλοήθους εφαρμογής είναι να εξυπηρετεί τον χρήστη της ενώ μίας κακόβουλής είναι να εκμεταλλευτεί τον χρήστη της.

Πρωτεύον στόχος της συγκεκριμένης μελέτης ήταν η μελέτη των δικαιωμάτων που αιτούνται οι εφαρμογές για την εγκατάσταση τους σε μία Android συσκευή και πώς αυτά αποτελούν ένα τρόπο ώστε να χαρακτηριστεί μία εφαρμογή ως κακόβουλη ή καλοήθης. Για την επίτευξη αυτού, μελετήθηκαν γνωστές εφαρμογές ως προς τα δικαιώματα τους οι οποίες προέρχονται από το Play Store αλλά και οι ίδιες που είναι Cracked και προέρχονται από Third Party Markets. Οι εφαρμογές που διατίθενται στο Play Store είναι ήδη ελεγμένες και θα μπορούσαμε να πούμε ότι θεωρούνται καλοήθεις. Οι Play Store εφαρμογές είναι εμφανές πως χρησιμοποιούν τα απολύτως απαραίτητα δικαιώματα που απαιτούνται για να εκτελέσουν τη λειτουργία ή τις λειτουργίες για τις οποίες έχουν δημιουργηθεί. Συνήθως όλες οι καλοήθεις αιτούνται δικαιώματα που χρησιμοποιούν στοιχεία του χρήστη όπως όνομα, επίθετο, ημερομηνία γέννησης, ηλεκτρονικό ταχυδρομείο. Αιτούνται αποδοχή δικαιώματος για σύνδεση στο διαδίκτυο γιατί μερικές εφαρμογές έχουν ανάγκη σύνδεσης με το διαδίκτυο για να λειτουργήσουν. Δικαιώματα για χρήση φωτογραφιών ή άλλων μέσων π.χ. βίντεο ή ήχος, χρήσης κάμερας και δικαιώματα για εντοπισμό τοποθεσίας, αυτά μπορεί να απαιτούνται από εφαρμογές Μέσων Κοινωνικής Δικτύωσης. Για την πραγματοποίηση αγορών σε εφαρμογές όπως το eBay και το eBanking

απαιτούν δικαιώματα για την χρήση τραπεζικών λογαριασμών. Από την άλλη πλευρά οι κακόβουλες εφαρμογές δεν αιτούνται μόνο τα δικαιώματα που απαιτούνται για την εκτέλεση των λειτουργιών που υπόσχονται αλλά αιτούνται και δικαιώματα που έχουν ως απώτερο στόχο να επηρεάσουν τη συσκευή και κατ' επέκτασιν τον χρήστη αφού οι προθέσεις των δημιουργών του είναι εξαρχής κακόβουλες. Τα περισσότερα έχουν ως στόχο την υποκλοπή προσωπικών δεδομένων του χρήστη, την αποστολή κακόβουλών μηνυμάτων που μπορεί να περιέχουν ιούς. Μπορεί να κλέβουν τις συντεταγμένες τοποθεσίας της συσκευής, να εγκαθιστούν άλλες εφαρμογές. Συχνό φαινόμενο είναι η υποκλοπή τραπεζικών πληροφοριών όπως αριθμούς τραπεζικών λογαριασμών, πολλές φορές έχουν μέχρι και την δυνατότητα για root Access. Σύμφωνα με την ανάλυση που έγινε οι Cracked αιτούνται περισσότερα δικαιώματα από τις εφαρμογές που προέρχονται από το Play Store. Οι Cracked εφαρμογές αιτούνται δικαιώματα που δεν τα αιτούνται οι εφαρμογές που προέρχονται από το Play Store, τέτοια δικαιώματα μπορεί να είναι: READ_SMS που δίνει τη δυνατότητα να διαβάζει μηνύματα SMS, RECEIVE_SMS η εφαρμογή λαμβάνει μηνύματα SMS, με επίπεδο προστασίας επικίνδυνο, SEND_SMS η εφαρμογή στέλνει μηνύματα SMS, WRITE_CONTACTS γράφει τα δεδομένα επαφών του χρήστη, CALL_PHONE μπορεί να ξεκινήσει μια τηλεφωνική κλήση η εφαρμογή χωρίς να περάσει από τη διεπαφή χρήστη του Dialer για να επιβεβαιώσει την κλήση ο χρήστης, RECORD_AUDIO επιτρέπει σε μια εφαρμογή την εγγραφή ήχου και το δικαίωμα VIBRATE επιτρέπει την πρόσβαση στον δονητή. Επομένως, όντως τα δικαιώματα που αιτούνται οι εφαρμογές για την εγκατάσταση τους σε μία συσκευή μπορούν να καταδείξουν τον χαρακτήρα της αφού δείχνουν και τις προθέσεις της μέσα από τα διάφορα σημεία της συσκευής που επιθυμεί να χρησιμοποιήσει. Άρα με τη γνώση των δικαιωμάτων μίας εφαρμογής μπορούμε να χαρακτηρίσουμε μία εφαρμογή ως καλοήθη ή ως κακόβουλη. Συνειδητοποιούμε ότι η ελαχιστοποίηση της έκτασης της ζημιάς εξαρτάται από την απαίτηση άδειας του χρήστη για προγράμματα που πιθανόν να αιτούνται ενέργειες που μπορεί να αποβούν ιδιαιτέρως επικίνδυνες.

Δεύτερος στόχος μας ήταν να μελετήσουμε πως επηρεάζονται οι Android εκδόσεις KITKAT, Lollipop και Marshmallow ανάλογα με το είδος μίας εφαρμογής και έτσι να αντιληφθούμε πως επηρεάζεται μία συσκευή Android γενικότερα. Αντιλαμβανόμαστε με αυτό τον τρόπο την εξέλιξη που έχει σημειωθεί σε επίπεδο ασφάλειας στα Android. Έτσι λοιπόν καταγράψαμε τη χρήση της RAM και της CPU από τις εφαρμογές καλοήθεις και κακόβουλες στις τρείς υπό μελέτη διαφορετικές εκδόσεις. Η RAM και η CPU επηρεάζουν την ασφάλεια του χρήστη, του λογισμικού και των συσκευών. Όταν η RAM και η CPU σημειώνουν υψηλά επίπεδα χρήσης, σπαταλούν περισσότερη μπαταρία και βαρυφορτώνουν το λογισμικό καθιστώντας το πιο ευάλωτο στις

επιθέσεις από το κακόβουλο λογισμικό, μπορεί να καταστρέφει η μπαταρία, ο επεξεργαστής και υπερθερμανθεί η συσκευή. Μερικές εφαρμογές επιθυμούν την καταστροφή των συσκευών με άμεση επιρροή της ασφάλειας γιατί χάνονται χρήσιμα στοιχεία που έχει αποθηκευμένα ο χρήστης στη συσκευή. Μέσα από τις μετρήσεις που έγιναν είναι εμφανές πως με την ανάπτυξη νέων εκδόσεων έχουμε μία γενικότερα μειωμένη χρήση της RAM και της CPU, πράγμα που σηματοδοτεί ότι οι κατασκευαστές είναι ιδιαίτερα ευαισθητοποιημένοι στο θέμα ασφάλειας και επιδιώκουν την περαιτέρω ενίσχυση του με την πάροδο των ετών. Κατά την ανάλυση που έγινε είναι πως οι εφαρμογές που είναι Cracked σε σχέση με αυτές που προέρχονται από το Play Store κάνουν περισσότερη χρήση της RAM και της CPU. Όπως προκύπτει μέσα από τους μέσους όρους χρήσης της RAM και της CPU στις εκδόσεις KITKAT, Lollipop και Marshmallow είναι πως οι Cracked κάνουν μειωμένη χρήση σε σχέση με τις Play Store.

Ο τρίτος στόχος μας ήταν η μελέτη του Network Traffic, που επιβεβαιώνει την υπόθεση μας πως οι εφαρμογές που είναι Cracked δημιουργούν μεγαλύτερη συμφόρηση στο δίκτυο σε σχέση με τις εφαρμογές που προέρχονται από το Play Store. Συγκριτικά τα δύο είδη εφαρμογών λόγω της προέλευσης τους σημειώνουν διαφορετικές ποσότητες χρήσης των TCP και HTTP ports με τις Cracked να υπερτερούν. Πράγμα που σημαίνει ότι οι εφαρμογές Cracked αφού δημιουργούν κίνηση πέραν της κανονικής έχουν και μία ύποπτη συμπεριφορά, με αποτέλεσμα να ελλοχεύουν κάποιοι κίνδυνοι.

Επομένως θα μπορούσαμε να χαρακτηρίσουμε μία εφαρμογή ως καλοήθη ή ως κακοήθη ανάλογα με τα δικαιώματα που αιτείται από τον χρήστη για να την εγκαταστήσει στην συσκευή του. Ακόμη αν κατά την χρήση μίας εφαρμογής καταναλώνεται πολύ γρήγορα η μπαταρία, κάποιες άλλες λειτουργίες μπλοκάρονται, σβήνει το τηλέφωνο ή οτιδήποτε άλλο που δεν καθιστά μία συσκευή εύχρηστη τότε είναι ένας εύκολος τρόπος να χαρακτηριστεί μία εφαρμογή ως κακοήθης. Οι μετρήσεις που έγιναν σε πρακτικό επίπεδο επιβεβαιώνουν τις υποσχέσεις εξέλιξης σε επίπεδο ασφάλειας της κάθε νέας έκδοσης, ανεξαρτήτως προέλευσης των εφαρμογών. Αυτό δείχνει πως το Android όντως ενισχύει το λογισμικό της κάθε φορά με πιο έξυπνη και ασφαλή τεχνολογία. Ταυτόχρονα επιβεβαιώνεται το γεγονός ότι η προέλευση μίας εφαρμογής παίζει σημαντικό ρόλο στην ασφάλειας. Δε δύναται μία εφαρμογή που προσφέρεται κρακαρισμένη και δωρεάν να μην αποβλέπει σε κάποιο στόχο, ενώ νόμιμα για να την αποκτήσεις μπορεί να απαιτείται και κάποιο χρηματικό ποσό.

Εν κατακλείδι θα πρέπει να είμαστε ιδιαίτερα προσεκτικοί με τις εφαρμογές που χρησιμοποιούμε ώστε να μπορούμε να διαφυλάξουμε την ασφάλεια των συσκευών μας και την ταυτόχρονα την δική μας.

5.2 Μελλοντική Εργασία

Στην παρούσα πειραματική διαδικασία που ακολουθήθηκε εντοπίστηκαν το τα συνηθέστερα δικαιώματα που αιτούνται τόσο οι καλοήθεις όσο και οι κακόβουλες εφαρμογές και τα πιθανά μέρη μίας Android συσκευής που μπορεί να επηρεάζονται. Ακόμη μελετήθηκε η επιρροή που δέχονται οι Android εκδόσεις KITKAT, Lollipop και Marshmallow ανάλογα με το είδος της εφαρμογής. Μελλοντική έρευνα θα μπορούσε επεκταθεί σε πιο συγκεκριμένες κατηγορίες κακόβουλων συμπεριφορών που αφορούν κακόβουλο λογισμικό και κακόβουλες τεχνικές διείσδυσης. Επιπλέον μπορούν να μελετηθούν και άλλες εκδόσεις Android αλλά και άλλα λειτουργικά συστήματα, ώστε να διαπιστωθεί το επίπεδο ασφάλειας που παρέχεται από κάθε κατασκευαστική εταιρεία. Τα αποτελέσματα της μεταπτυχιακής διατριβής θα μπορούσαν να αποτελέσουν το έναυσμα ώστε να δημιουργηθεί antimalware εφαρμογή που να ανιχνεύει κακόβουλες συμπεριφορές σε συσκευές Android, να ειδοποιεί τον χρήστη και έτσι να λαμβάνει τα μέτρα του.

Βιβλιογραφία

[01] "AAFER, Y., W. DU and H. YIN. Droidapiminer: Mining api-level features for robust malware detection in androidAnonymous *International Conference on Security and Privacy in Communication Systems*, 2013.

[02] ALLIANCE, O.H., 2011. Industry leaders announce open platform for mobile devices, 2007. *URL Http://www.Openhandsetalliance.com/press_110507.Html*.

[03] ASHRAF, Z., 2013. DIY: Android Malware Analysis–Taking apart OBAD (part 1). *Security Intelligence*.

[04] BARTEL, A., J. KLEIN, Y. LE TRAON and M. MONPERRUS. Dexpler: converting android dalvik bytecode to jimple for static analysis with sootAnonymous *Proceedings of the ACM SIGPLAN International Workshop on State of the Art in Java Program analysis*, 2012.

[05] BLÄSING, T., L. BATYUK, A. SCHMIDT, S.A. CAMTEPE and S. ALBAYRAK. An android application sandbox system for suspicious software detectionAnonymous *Malicious and unwanted software (MALWARE), 2010 5th international conference on*, 2010.

[06] BOSE, A., X. HU, K.G. SHIN and T. PARK. Behavioral detection of malware on mobile handsetsAnonymous *Proceedings of the 6th international conference on Mobile systems, applications, and services*, 2008.

[07] BROWSER, O., 2013. Search Engine including Mobile Market Share. *StatCounter Global Stats*.

[08] BURGUERA, I., U. ZURUTUZA and S. NADJM-TEHRANI. Crowdroid: behavior-based malware detection system for androidAnonymous *Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices*, 2011.

[09] BURNS, M., 2012. Eric Schmidt:" There are now 1.3 million Android device activations per day". *Tech Crunch*, vol. 5.

[10] CADENHEAD, R., 2002. *Sams teach yourself Java 2 in 24 hours.* Sams Publishing.

[11] CASTILLO, C.A., 2011. Android malware past, present, and future. *White Paper of McAfee Mobile Security Working Group*, vol. 1, pp. 16.

[12] CHENG, J., S.H. WONG, H. YANG and S. LU. Smartsiren: virus detection and alert for smartphonesAnonymous *Proceedings of the 5th international conference on Mobile systems, applications and services*, 2007.

[13] CHIN, E., A.P. FELT, K. GREENWOOD and D. WAGNER. Analyzing inter-application communication in AndroidAnonymous *Proceedings of the 9th international conference on Mobile systems, applications, and services*, 2011.

[14] CHRISTODORESCU, M. and JHA, S., 2006. *Static Analysis of Executables to Detect Malicious Patterns*.

[15] CINAR, O., 2012. *Android apps with Eclipse.* Apress.

[16] DAGON, D., MARTIN, T. and STARNER, T., 2004. Mobile phones as computing devices: The viruses are coming!. *IEEE Pervasive Computing*, vol. 3, no. 4, pp. 11-15.

[17] DAWSON, M., 2015. *New Threats and Countermeasures in Digital Crime and Cyber Terrorism.* IGI Global.

[18] DESNOS, A., 2011. Androguard: Reverse engineering, malware and goodware analysis of android applications... and more (ninja!). *Retrieved June*, vol. 10, pp. 2014.

[19] DIXON, B., Y. JIANG, A. JAIANTILAL and S. MISHRA. Location based power analysis to detect malicious code in smartphonesAnonymous *Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices*, 2011.

[20] EGELE, M., SCHOLTE, T., KIRDA, E. and KRUEGEL, C., 2012. A survey on automated dynamic malware-analysis techniques and tools. *ACM Computing Surveys (CSUR)*, vol. 44, no. 2, pp. 6.

[21] ENCK, W., GILBERT, P., HAN, S., TENDULKAR, V., CHUN, B., COX, L.P., JUNG, J., MCDANIEL, P. and SHETH, A.N., 2014. TaintDroid: an information-flow tracking system for realtime privacy monitoring on smartphones. *ACM Transactions on Computer Systems (TOCS)*, vol. 32, no. 2, pp. 5.

[22] ENCK, W., M. ONGTANG and P. MCDANIEL. On lightweight mobile phone application certificationAnonymous *Proceedings of the 16th ACM conference on Computer and communications security*, 2009.

[23] FARUKI, P., V. GANMOOR, V. LAXMI, M.S. GAUR and A. BHARMAL. AndroSimilar: robust statistical feature signature for Android malware detectionAnonymous *Proceedings of the 6th International Conference on Security of Information and Networks*, 2013.

[24] FELT, A.P., E. CHIN, S. HANNA, D. SONG and D. WAGNER. Android permissions demystifiedAnonymous *Proceedings of the 18th ACM conference on Computer and communications security*, 2011.

[25] FELT, A.P., M. FINIFTER, E. CHIN, S. HANNA and D. WAGNER. A survey of mobile malware in the wildAnonymous *Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices*, 2011.

[26] FENG, Y., S. ANAND, I. DILLIG and A. AIKEN. Apposcopy: Semantics-based detection of android malware through static analysisAnonymous *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering*, 2014.

[27] FRANCESCHI-BICCHIERAI, L., 2015. Goodbye, Android. *Motherboard.Vice.Retrieved August*, vol. 2.

[28] FUCHS, A.P., CHAUDHURI, A. and FOSTER, J.S., 2009. *Scandroid: Automated Security Certification of Android*.

[29] GANAPATI, P., 2012. Study Shows Some Android Apps Leak User Data Without Clear Notifications| Gadget Lab. *Wired.Com.Http://www.Wired.com/gadgetlab/2010/09/data-Collection-Android/.Retrieved*, pp. 01-30.

[30] GEIER, E., 2016. How to Keep Your PC Safe With Sandboxing. *PCWorld.Ladattu*, vol. 20, pp. 2016.

[31] GOLDBERG, I., D. WAGNER, R. THOMAS and E.A. BREWER. A secure environment for untrusted helper applications: Confining the wily hackerAnonymous *Proceedings of the 6th conference on USENIX Security Symposium, Focusing on Applications of Cryptography*, 1996.

[32] HISPASEC SISTEMAS, S., 2012. *Virustotal Public API*.

[33] HOUMANSADR, A., S.A. ZONOUZ and R. BERTHIER. A cloud-based intrusion detection and response system for mobile phonesAnonymous *Dependable Systems and Networks Workshops (DSN-W), 2011 IEEE/IFIP 41st International Conference on*, 2011.

[34] HUANG, C., TSAI, Y. and HSU, C., 2013. Performance evaluation on permission-based detection for android malware. In: *Advances in Intelligent Systems and Applications-Volume 2*Springer, pp. 111-120.

[35] KIM, H., J. SMITH and K.G. SHIN. Detecting energy-greedy anomalies and mobile malware variantsAnonymous *Proceedings of the 6th international conference on Mobile systems, applications, and services*, 2008.
[36] KIM, J., YOON, Y., YI, K., SHIN, J. and CENTER, S., 2012. ScanDal: Static analyzer for detecting privacy leaks in android applications. *Most*, vol. 12.
[37] KINGSLEY-HUGHES, A., 2015. The toxic hellstew survival guide. *ZDnet.Retrieved August*, vol. 2.
[38] KOLTER, J.Z. and MALOOF, M.A., 2006. Learning to detect and classify malicious executables in the wild. *Journal of Machine Learning Research*, vol. 7, no. Dec, pp. 2721-2744.
[39] KOZMA, L., 2008. k Nearest Neighbors algorithm (kNN). *Helsinki University of Technology*.
[40] LO, R.W., LEVITT, K.N. and OLSSON, R.A., 1995. MCF: A malicious code filter. *Computers & Security*, vol. 14, no. 6, pp. 541-566.
[41] LOCKHEIMER, H., 2012. *Android and Security-Official Google Mobile Blog*.
[42] LOHR, S. Suit Opens a Window Into Google. *The New York Times.ISSN*, pp. 0362-4331.
[43] MAIER, D., T. MÜLLER and M. PROTSENKO. Divide-and-conquer: Why android malware cannot be stoppedAnonymous *Availability, Reliability and Security (ARES), 2014 Ninth International Conference on*, 2014.
[44] MYLONAS, A., S. DRITSAS, B. TSOUMAS and D. GRITZALIS. Smartphone security evaluation The malware attack caseAnonymous *Security and Cryptography (SECRYPT), 2011 Proceedings of the International Conference on*, 2011.
[45] PALLER, G., 2011. *Dedexer User's Manual*.
[46] PROTALINSKI, E., 2012. Android malware numbers exploded to 25,000 in June 2012. *ZDNet, July*, vol. 17.
[47] RAPHAEL, J., 2012. Exclusive: Inside Android 4.2's powerful new security system. *Computerworld Blogs*.
[48] RAVEENDRANATH, R., V. RAJAMANI, A.J. BABU and S.K. DATTA. Android malware attacks and countermeasures: Current and future directionsAnonymous *Control, Instrumentation, Communication and Computational Technologies (ICCICCT), 2014 International Conference on*, 2014.
[49] RIVERA, J. and VAN DER MEULEN, R. Gartner says annual smartphone sales surpassed sales of feature phones for the first time in 2013 (feb. 2014). *Dostopno Na: Http://www.Gartner.com/newsroom/id/2665715*.
[50] Sable Research Group, 2016. *Soot: A Framework for Analyzing and Transforming Java and Android Applications*.
[51] SADEGHI, A., BAGHERI, H. and GARCIA, J., 2016. A Taxonomy and Qualitative Comparison of Program Analysis Techniques for Security Assessment of Android Software. *IEEE Transactions on Software Engineering*.
[52] SANZ, B., I. SANTOS, C. LAORDEN, X. UGARTE-PEDRERO, P.G. BRINGAS and G. ÁLVAREZ. Puma: Permission usage to detect malware in androidAnonymous *International Joint Conference CISIS'12-ICEUTE´ 12-SOCO´ 12 Special Sessions*, 2013.
[53] SATO, R., CHIBA, D. and GOTO, S., 2013. Detecting Android malware by analyzing manifest files. *Proceedings of the Asia-Pacific Advanced Network*, vol. 36, no. 23-31, pp. 17.
[54] SCHULTZ, M.G., E. ESKIN, F. ZADOK and S.J. STOLFO. Data mining methods for detection of new malicious executablesAnonymous *Security and Privacy, 2001. S&P 2001. Proceedings. 2001 IEEE Symposium on*, 2001.

[55] SHABTAI, A., FLEDEL, Y., KANONOV, U., ELOVICI, Y. and DOLEV, S., 2009. Google android: A state-of-the-art review of security mechanisms. *ArXiv Preprint arXiv:0912.5101*.

[56] SHABTAI, A., FLEDEL, Y., KANONOV, U., ELOVICI, Y., DOLEV, S. and GLEZER, C., 2010. Google android: A comprehensive security assessment. *IEEE Security & Privacy*, vol. 8, no. 2, pp. 35-44.

[57] SHABTAI, A., KANONOV, U., ELOVICI, Y., GLEZER, C. and WEISS, Y., 2012. "Andromaly": a behavioral malware detection framework for android devices. *Journal of Intelligent Information Systems*, vol. 38, no. 1, pp. 161-190.

[58] SHIN, W., S. KIYOMOTO, K. FUKUSHIMA and T. TANAKA. Towards formal analysis of the permission-based security model for androidAnonymous *Wireless and Mobile Communications, 2009. ICWMC'09. Fifth International Conference on*, 2009.

[59] SPREITZENBARTH, M. The Evil Inside a Droid—Android Malware: past, present and futureAnonymous *Proceedings of the 1st Baltic Conference on Network Security & Forensics*, 2012.

[60] STRAZZARE, T., 2011. Security Alert: Zsone Trojan found in Android Market. *Online] may*, vol. 11.

[61] TANG, W., G. JIN, J. HE and X. JIANG. Extending Android security enforcement with a security distance modelAnonymous *Internet Technology and Applications (iTAP), 2011 International Conference on*, 2011.

[62] TESAURO, G.J., KEPHART, J.O. and SORKIN, G.B., 1996. Neural networks for computer virus recognition. *IEEE Expert*, vol. 11, no. 4, pp. 5-6.

[63] TUNG, L., 2015. Android security a'market for lemons' that leaves 87 percent vulnerable. *Zdnet.Com.ZDNet.Retrieved*, pp. 10-14.

[64] WEI, X., L. GOMEZ, I. NEAMTIU and M. FALOUTSOS. Permission evolution in the android ecosystemAnonymous *Proceedings of the 28th Annual Computer Security Applications Conference*, 2012.

[65] WHITWAM, R., 2015. *Android Antivirus Apps are useless—here's what to do Instead*.

[66] WOGNSEN, E.R., KARLSEN, H.S., OLESEN, M.C. and HANSEN, R.R., 2014. Formalisation and analysis of Dalvik bytecode. *Science of Computer Programming*, vol. 92, pp. 25-55.

[67] WU, D., C. MAO, T. WEI, H. LEE and K. WU. Droidmat: Android malware detection through manifest and api calls tracingAnonymous *Information Security (Asia JCIS), 2012 Seventh Asia Joint Conference on*, 2012.

[68] WU, F., NARANG, H. and CLARKE, D., 2014. An overview of mobile malware and solutions. *Journal of Computer and Communications*, vol. 2, no. 12, pp. 8.

[69] XIE, L., X. ZHANG, J. SEIFERT and S. ZHU. pBMDS: a behavior-based malware detection system for cellphone devicesAnonymous *Proceedings of the third ACM conference on Wireless network security*, 2010.

[70] YAN, L. and H. YIN. DroidScope: Seamlessly Reconstructing the OS and Dalvik Semantic Views for Dynamic Android Malware Analysis.Anonymous *USENIX security symposium*, 2012.

[71] YAN, P., 2014. A look at repackaged apps and their effect on the mobile threat landscape. *TrendLabs Security Intelligence Blog*.

[72] YEE, B., D. SEHR, G. DARDYK, J.B. CHEN, R. MUTH, T. ORMANDY, S. OKASAKA, N. NARULA and N. FULLAGAR. Native client: A sandbox for portable, untrusted x86 native codeAnonymous *Security and Privacy, 2009 30th IEEE Symposium on*, 2009.

[73] YOU, I. and K. YIM. Malware obfuscation techniques: A brief surveyAnonymous *Broadband, Wireless Computing, Communication and Applications (BWCCA), 2010 International Conference on*, 2010.
[74] YUHUI, F. and X. NING. The Behavioral Analysis of Android MalwareAnonymous *3rd International Conference on Next Generation Computer and Information Technology (NGCIT 2014)*, 2014.
[75] ZHAO, M., F. GE, T. ZHANG and Z. YUAN. Antimaldroid: An efficient svm-based malware detection framework for androidAnonymous *International Conference on Information Computing and Applications*, 2011.
[76] ZHENG, M., M. SUN and J.C. LUI. Droid analytics: a signature based analytic system to collect, extract, analyze and associate android malwareAnonymous *Trust, Security and Privacy in Computing and Communications (TrustCom), 2013 12th IEEE International Conference on*, 2013.
[77] ZHOU, W., Y. ZHOU, X. JIANG and P. NING. Detecting repackaged smartphone applications in third-party android marketplacesAnonymous *Proceedings of the second ACM conference on Data and Application Security and Privacy*, 2012.
[78] ZHOU, Y. and X. JIANG. Dissecting android malware: Characterization and evolutionAnonymous *Security and Privacy (SP), 2012 IEEE Symposium on*, 2012.
[79] ZHOU, Y., Z. WANG, W. ZHOU and X. JIANG. Hey, you, get off of my market: detecting malicious apps in official and alternative android markets.Anonymous *NDSS*, 2012.
[80] Wikipedia, "Android (operating system)," [Online]. Available: http://en.wikipedia.org/wiki/Android_%28operating_system%29. [Accessed 10 02 2017].
[81] Wikipedia, the free encyclopedia –Web Mercator. https://en.wikipedia.org/wiki/Web_Mercator (Accessed 16 03 2017).
[82] https://en.wikipedia.org/wiki/Android_version_history (Accessed 16 03 2017).
[83] “Number of available Android applications - AppBrain.” [Online]. Available: http://www.appbrain.com/stats/number-of-android-apps. [Accessed: 28 01 2017].
[84] Staff (September 7, 2013). "Privacy Scandal: NSA Can Spy on Smart Phone Data". Retrieved September 7, 2013.
[85] James Ball (January 28, 2014). "Angry Birds firm calls for industry to respond to NSA spying revelations | World news". theguardian.com. Retrieved February 2, 2014.
[86] "Mobile malware exaggerated by "charlatan" vendors, says Google engineer". PC Advisor. November 24, 2011. Retrieved November 9, 2012.
[87] "Android 4.2 brings new security features to scan sideloaded apps". Android Central. Retrieved November 9, 2012.
[88] "Android malware perspective: only 0.5% comes from the Play Store". Phonearena.com. Retrieved March 14, 2013.
[89] "Samsung Armors Android to Take On BlackBerry". The New York Times. February 28, 2013.
[90] "AppAnalysis.org: Real Time Privacy Monitoring on Smartphones". Retrieved February 21, 2012.
[91] "Android Security Overview". Android Open Source Project. Retrieved February 20, 2012.
[92] "Antivirus for Android Smartphones". AVG. Retrieved February 16, 2012.

[93] “Mind the (Security) Gaps: The 1H 2015 Mobile Threat Landscape - Security News - Trend Micro USA.” [Online]. Available: http://www.trendmicro.com/vinfo/us/security/news/mobile-safety/mindthe-security-gaps-1h-2015-mobile-threat-landscape. [Accessed: 08-Dec- 2015].
[94] “The Mobile Landscape Roundup: 1H 2014 - Security News - Trend Micro USA.” [Online]. Available: http://www.trendmicro.com/vinfo/us/security/news/mobile-safety/themobile-landscape-roundup-1h-2014. [Accessed: 08-Dec-2015].
[95] “root exploits.” [Online]. Available: http://www.selinuxproject.org/~jmorris/lss2011_slides/caseforseandroid. pdf. [Accessed: 15-Dec-2015].
[96] “contagio mobile: Backdoor.AndroidOS.Obad.a.” [Online]. Available: http://contagiominidump.blogspot.in/2013/06/backdoorandroidosobada. html. [Accessed: 28-Oct-2015].
[97] “Riskware | Internet Security Threats.” [Online]. Available: http://usa.kaspersky.com/internet-security-center/threats/riskware#.Vm-5IUp97IU. [Accessed: 15-Dec-2015].
[98] “NotCompatible Android Trojan: What You Need to Know | PCWorld.” [Online]. Available: http://www.pcworld.com/article/254918/notcompatible_android_trojan_what_you_need_to_know.html. [Accessed: 15-Dec-2015].
[99] “[Utility][Tool][Windows] Baksmali / Smali Ma… | Android Development and Hacking.” [Online]. Available: http://forum.xdadevelopers.com/showthread.php?t=2311766. [Accessed: 22-Dec-2015].
[100] “strace download | SourceForge.net.” [Online]. Available: http://sourceforge.net/projects/strace/. [Accessed: 22-Dec-2015].
[101] Department of Computer Science Friedrich-Alexander-University Erlangen-Nuremberg. Mobile-Sandbox. http://www.mobile-sandbox.com, January 2012.
[102] http://www.crackapk.com
[103] http://www.appcake.net/
[104] https://developer.android.com/guide/topics/permissions/requesting.html#normal-dangerous
[105] https://developer.android.com/reference/android/Manifest.permission.html
[106] https://www.techopedia.com/definition/29917/network-traffic
[107] https://en.wikipedia.org/wiki/Transmission_Control_Protocol
[108] https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol

Παράρτημα Α
Εικόνες

Παράρτημα Β
Γραφήματα

Παράρτημα Γ
Πίνακες

Πίνακας 4.1: : Τα δικαιώματα που αιτούνται οι εφαρμογές από το Play Store (μπλε χρώμα) και οι εφαρμογές που είναι κρακαρισμένες και είναι από Third Party Markets (πράσινο χρώμα)……54

A/A	Application	Permission																
		android.permission.INTERNET	android.permission.ACCESS_COARSE_LOCATION	android.permission.ACCESS_FINE_LOCATION	android.permission.CAMERA	android.permission.BLUETOOTH	android.permission.READ_EXTERNAL_STORAGE	android.permission.READ_CONTACTS	android.permission.WRITE_EXTERNAL_STORAGE	android.permission.MEDIA_CONTENT_CONTROL	android.permission.READ_SMS	android.permission.RECEIVE_SMS	android.permission.SEND_SMS	android.permission.WRITE_CONTACTS	android.permission.CALL_PHONE	android.permission.RECORD_AUDIO	android.permission.VIBRATE
1	*360 Security – Antivirus Free*	•	•	•	•	•	•	•										
	360 Security – Antivirus Free	•	•	•	•	•	•	•	•		•	•	•	•	•			
2	*Run Cow Run*	•					•		•									
	Run Cow Run	•	•	•			•	•	•								•	
3	*Solar System Scope*		•	•														
	Solar System Scope	•					•	•	•								•	•
4	*Mean Spheres Attack*									•								
	Mean Spheres Attack	•					•	•	•								•	
5	*Fillshape*									•								
	Fillshape	•					•		•	•								
6	*Piques*																	
	Piques	•						•										•
7	*Lunchbox*	•								•								
	Lunchbox	•					•		•	•							•	
8	*Calorie Counter*				•					•								
	Calorie Counter				•		•		•								•	
9	*3D Charts*									•								
	3D Charts	•	•	•				•										
10	*Root Browser*	•								•								
	Root Browser	•					•		•									
11	*Enemy Strike*	•						•		•								
	Enemy Strike	•					•	•	•	•								
12	*Audio Manager*	•	•	•	•					•								
	Audio Manager	•	•	•	•		•		•	•								
13	*Vector*	•								•								
	Vector	•			•			•									•	
14	*Spy Mouse*	•	•	•														
	Spy Mouse	•	•	•			•		•									
15	*Link2SD*									•								
	Link2SD						•		•	•								
16	*4Shared*	•	•	•				•		•								
	4Shared	•	•	•				•		•	•	•	•					
17	*John NES - NES Emulator*	•								•								
	John NES - NES Emulator	•				•		•		•								
18	*Clean Master – Free Antivirus*	•			•					•								
	Clean Master – Free Antivirus	•			•		•		•	•								
19	*3C Toolbox*	•	•	•	•			•	•	•								
	3C Toolbox	•	•	•	•			•		•	•	•	•					
20	*9GAG*									•								
	9GAG	•					•		•	•								
21	*Root Tool Case*									•								
	Root Tool Case		•	•						•								
22	*Mobile Security & Antivirus*	•	•	•	•			•		•								
	Mobile Security & Antivirus	•	•	•	•		•	•	•	•	•	•	•					
23	*Unit Converter*	•																
	Unit Converter	•	•	•														
24	*Dual Sim Selector*							•		•								
	Dual Sim Selector						•	•	•	•								
25	*Smart IPTV*	•																
	Smart IPTV	•	•	•														

Πίνακας 4.2: Ο πιο πάνω πίνακας παρουσιάζει τη χρήση CPU που κάνουν οι εφαρμογές που προέρχονται από το Play Store και αυτών που είναι Cracked στις εκδόσεις Android.....56

CPU						
Application	KITKAT		Lollipop		Marshmallow	
	Play Store	Cracked	Play Store	Cracked	Play Store	Cracked
360 Security - Antivirus Free	3,51%	4,72%	3,43%	4,56%	3,26%	4,31%
Run Cow Run	4,38%	4,81%	3,86%	4,18%	3,41%	3,87%
Solar System Scope	3,95%	4,76%	4,01%	3,84%	3,58%	3,57%
Mean Sphere Attack	3,56%	4,15%	3,14%	3,47%	2,77%	3,64%
Fillshape	2,74%	3,41%	2,47%	3,15%	2,13%	2,78%
Piques	2,46%	2,84%	2,27%	2,28%	2,18%	2,13%
Lunchbox	2,83%	3,49%	2,53%	3,06%	2,63%	2,86%
Calorie Counter	2,30%	2,76%	1,87%	2,49%	1,74%	2,29%
3D Charts	3,17%	3,83%	2,74%	4,13%	2,52%	3,84%
Root Browser	3,94%	4,69%	3,68%	4,39%	3,29%	3,97%
Enemy Strike	4,18%	4,12%	3,75%	3,73%	3,46%	3,52%
Audio Manager	2,68%	3,17%	2,46%	2,65%	2,26%	2,38%
Vector	2,41%	2,59%	2,16%	2,16%	1,97%	2,04%
Spy Mouse	2,80%	3,15%	2,43%	2,42%	2,24%	2,25%
Link2SD	1,93%	2,17%	1,41%	1,85%	1,15%	1,64%
4Shared	2,19%	2,64%	1,64%	2,36%	1,37%	2,19%
John NES - NES Emulator	2,74%	2,38%	2,54%	2,14%	2,38%	1,84%
Clean Master - Free Antivirus	3,18%	3,61%	2,86%	3,17%	2,51%	2,87%
3C Toolbox	3,42%	2,94%	3,19%	2,63%	2,86%	2,53%
9GAG	3,37%	3,76%	3,14%	3,27%	2,67%	2,91%
Rool Tool Case	2,58%	2,81%	2,73%	2,51%	2,48%	2,58%
Mobile Security & Antivirus	2,86%	3,39%	2,40%	2,97%	2,10%	2,76%
Unit Converter	3,27%	3,66%	3,07%	3,41%	2,73%	3,16%
Dual Sim Selector	4,97%	5,26%	4,73%	4,86%	4,65%	4,52%
Smart IPTV	4,62%	4,58%	4,54%	4,25%	4,39%	3,88%

Πίνακας 4.3: Ο πιο πάνω πίνακας παρουσιάζει τη χρήση RAM που κάνουν οι εφαρμογές που προέρχονται από το Play Store και αυτών που είναι Cracked στις εκδόσεις Android.....60

RAM						
Application	KITKAT		Lollipop		Marshmallow	
	Play Store	Cracked	Play Store	Cracked	Play Store	Cracked
360 Security - Antivirus Free	51,66	53,37	50,78	51,74	48,75	49,52
Run Cow Run	43,28	47,62	41,36	46,13	38,53	44,26
Solar System Scope	41,46	42,28	39,25	40,8	38,36	39,45
Mean Sphere Attack	32,58	36,03	30,18	33,76	28,25	31,74
Fillshape	34,62	35,72	33,83	34,59	31,48	31,83
Piques	34,15	33,12	32,47	31,52	31,83	30,07
Lunchbox	39,41	42,59	35,68	40,68	33,47	39,35
Calorie Counter	42,73	43,71	39,31	43,29	36,51	40,27
3D Charts	32,79	34,68	31,7	32,36	28,79	31,64
Root Browser	45,44	47,9	43,74	46,75	42,26	43,81
Enemy Strike	47,16	52,81	44,52	51,52	42,73	50,4
Audio Manager	55,83	54,47	53,84	52,7	50,85	51,36
Vector	31,6	34,96	29,49	30,81	27,64	28,17
Spy Mouse	48,12	53,78	47,36	51,63	46,37	48,95
Link2SD	33,82	34,85	34,07	32,08	31,87	31,25
4Shared	59,64	62,1	57,34	58,46	54,96	55,62
John NES - NES Emulator	45,85	42,79	42,95	41,76	44,17	39,72
Clean Master - Free Antivirus	38,69	41,24	36,62	39,14	35,7	36,84
3C Toolbox	61,48	63,92	58,51	62,58	57,19	58,62
9GAG	34,72	37,58	32,6	34,83	29,68	32,98
Rool Tool Case	33,49	30,64	34,18	28,51	33,49	27,45
Mobile Security & Antivirus	58,31	62,43	56,27	59,73	53,51	57,46
Unit Converter	42,06	44,16	41,48	45,95	42,14	43,59
Dual Sim Selector	37,29	39,85	36,52	38,6	34,76	36,23
Smart IPTV	38,75	40,39	35,61	38,97	32,43	36,41

Application	Average			
	RAM(MB)		CPU(%)	
	Play Store	Cracked	Play Store	Cracked
360 Security - Antivirus Free	50,4	51,54	3,40%	4,53%
Run Cow Run	41,06	46	3,88%	4,29%
Solar System Scope	39,69	40,84	3,85%	4,06%
Mean Sphere Attack	30,34	33,84	3,16%	3,75%
Fillshape	33,31	34,05	2,45%	3,11%
Piques	32,82	31,57	2,30%	2,42%
Lunchbox	36,19	40,87	2,66%	3,14%
Calorie Counter	39,52	42,42	1,97%	2,51%
3D Charts	31,09	32,89	2,81%	3,93%
Root Browser	43,81	46,15	3,64%	4,35%
Enemy Strike	44,8	51,58	3,80%	3,79%
Audio Manager	53,51	52,84	2,47%	2,73%
Vector	29,58	31,31	2,18%	2,26%
Spy Mouse	47,28	51,45	2,49%	2,61%
Link2SD	33,25	32,73	1,50%	1,89%
4Shared	57,31	58,73	1,73%	2,40%
John NES - NES Emulator	44,32	41,42	2,55%	2,12%
Clean Master - Free Antivirus	37	39,07	2,85%	3,22%
3C Toolbox	59,06	61,71	3,16%	2,70%
9GAG	32,33	35,13	3,06%	3,31%
Rool Tool Case	33,72	28,87	2,60%	2,63%
Mobile Security & Antivirus	56,03	59,87	2,45%	3,04%
Unit Converter	41,89	44,57	3,02%	3,41%
Dual Sim Selector	36,19	38,23	4,78%	4,88%
Smart IPTV	35,6	38,59	4,52%	4,24%

Πίνακας 4.7: Ποσότητα TCP και HTTP ports που χρησιμοποιούν οι Play Store και οι Cracked εφαρμογές στις εκδόσεις KitKat, Lollipop και Marshmallow.....69

Application	KITKAT				LOLLIPOP				MARSHMALLOW			
	Play Store		Cracked		Play Store		Cracked		Play Store		Cracked	
	TCP	HTTP	TCP	HTTP	TCP	HTTP	TCP	HTTP	TCP	HTTP	TCP	HTTP
360 Security - Antivirus Free	86	13	92	24	118	37	163	67	112	39	147	59
Run Cow Run	129	8	125	11	146	31	146	25	114	16	124	26
Solar System Scope	0	0	43	9	0	0	79	23	0	0	52	13
Mean Sphere Attack	0	0	57	27	0	0	84	62	0	0	79	57
Fillshape	0	0	36	13	0	0	48	16	0	0	42	18
Piques	0	0	42	11	0	0	89	26	0	0	74	21
Lunchbox	57	16	74	26	73	36	73	40	64	22	68	34
Calorie Counter	0	0	0	0	0	0	0	0	0	0	0	0
3D Charts	0	0	39	10	0	0	97	46	0	0	86	25
Root Browser	381	48	359	62	339	48	271	79	327	41	276	83
Enemy Strike	37	4	46	17	37	12	81	48	28	7	78	50
Audio Manager	76	25	93	30	74	42	147	61	64	36	126	53
Vector	154	39	212	54	145	39	251	94	136	32	237	85
Spy Mouse	87	42	73	39	62	31	96	39	56	26	84	36
Link2SD	0	0	0	0	0	0	0	0	0	0	0	0
4Shared	376	36	438	71	366	26	406	93	348	21	387	81
John NES - NES Emulator	85	41	97	43	59	27	148	65	52	29	139	62
Clean Master - Free Antiviru	279	62	298	59	254	41	274	75	241	38	260	73
3C Toolbox	53	11	58	23	68	15	43	20	56	9	42	28
9GAG	340	68	274	73	335	64	285	83	315	53	264	78
Rool Tool Case	0	0	41	9	0	0	62	24	0	0	58	17
Mobile Security & Antivirus	71	8	137	38	67	19	167	59	52	10	152	57
Unit Converter	59	26	83	29	58	24	75	41	48	16	73	32
Dual Sim Selector	0	0	0	0	0	0	0	0	0	0	0	0
Smart IPTV	421	64	416	73	402	62	397	88	384	40	376	81