Πανεπιστήμιο Πειραιώς, Τμήμα Ψηφιακών Συστημάτων

Π.Μ.Σ. Ασφάλεια Ψηφιακών Συστημάτων

Ακαδημαϊκό έτος 2012-2013

Πτυχιακή Εργασία

Μεταπτυχιακού Διπλώματος Ειδίκευσης

Ασφάλεια στο λειτουργικό σύστημα Android

Σπουδαστής: Δημήτριος Γ.Ν.Σ. Παπαδέας ΜΤΕ/1060

Επιβλέπον καθηγητής: Κωνσταντίνος Λαμπρινουδάκης

Αθήνα 2013

Περίληψη

Η ακόλουθη πτυχιακή εργασία αποτελεί μία μελέτη των δομών ασφάλειας και του ιομορφικού λογισμικού στο λειτουργικό σύστημα κινητών συσκευών Android. Το Android είναι ένα σχετικά καινούριο λειτουργικό σύστημα που δημιουργήθηκε από την Google. Το Android έχει πυρήνα Linux και είναι γραμμένο σε γλώσσα Java επίσης έχει ενισχυθεί με τους δικούς του μηχανισμούς ασφαλείας οι οποίοι είναι προσαρμοσμένοι για ένα περιβάλλον κινητών συσκευών. Κάθε Android εφαρμογή έχει τη δική της ταυτότητα και για να επικοινωνήσει με άλλες εφαρμογές χρησιμοποιεί τους IPC μηχανισμούς που παρέχονται από το σύστημα. Η αρχιτεκτονική ασφάλειας βασίζεται πάνω σε δικαιώματα που ελέγχουν την πρόσβαση σε διαφόρους πόρους και μεταξύ των IPC μηχανισμών. Τα δικαιώματα λειτουργούν ώστε να προστατεύεται ο χρήστης αλλά και η εφαρμογές από κάθε είδους κακόβουλη δραστηριότητα. Το μοντέλο απειλών στις κινητές συσκευές περιλαμβάνει τρεις τύπους απειλών: το malware, το grayware και το spyware. Το ιομορφικό λογισμικό μπορεί να ταξινομηθεί επίσης και με βάση την λειτουργικότητα του ανάλογα με τον τρόπο εγκατάστασης, τον τρόπο ενεργοποίησης και τέλος ανάλογα με τον τύπο του κακόβουλου φορτίου που μπορεί να φέρει. οι συγγραφείς ιομορφικού λογισμικού προτιμούν να ανασκευάζουν δημοφιλείς εφαρμογές προσθέτοντας το κακόβουλο φορτίο και τις αναδημοσιεύουν δωρεάν σε εναλλακτικά repositories με σκοπό να παραπλανηθούν οι χρήστες και να τις εγκαταστήσουν. Το κακόβουλο λογισμικό στο Android εξελίσσεται ταχύτατα σε πιο επικίνδυνο και πιο κρυφό θέτοντας σημαντικές προκλήσεις για τον εντοπισμό του αλλά και για την ευρύτερη ασφάλεια στις συσκευές Android. Η ανάλυση ενός κακόβουλου λογισμικού μπορεί να ποικίλει σε βαθμό δυσκολίας ανάλογα με την πολυπλοκότητα της εφαρμογής και τις απαιτήσεις που τίθενται στον αναλυτή. Υπάρχει ένα σύνολο εργαλείων που καλύπτουν ικανοποιητικά τις ανάγκες τις ανάλυσης. Η οποία κατηγοριοποιείται σε στατική και δυναμική. Η στατική ανάλυση είναι η επισκόπηση του κώδικα του κακόβουλου λογισμικού και μπορεί να δώσει πληροφορίες για μία εφαρμογή αλλά χρειάζεται εξειδικευμένες γνώσεις. Ενώ η δυναμική ανάλυση είναι η ανάλυση της συμπεριφοράς που μελετάει την εφαρμογή καθώς αυτή εκτελείται. Η ανάλυση είτε στατική είτε δυναμική μίας ύποπτης εφαρμογής πρέπει να σχεδιάζεται καλά και να εκτελείται προσεκτικά ανεξάρτητα από το ποσό εύκολη μπορεί να είναι φαινομενικά. Διότι οι συγγραφείς κακόβουλου λογισμικού γίνονται όλο και πιο έμπειροι και αρχίζουν να ενσωματώνουν μηχανισμούς που ελέγχουν εάν οι εφαρμογές τους εκτελούνται σε περιβάλλον ανάλυσης ώστε να αναστέλλεται η λειτουργία τους. Τέλος δύο αξιόλογες μοντελοποιήσεις των απειλών στις κινητές συσκευές υπό την μορφή του top 10 προσφέρονται από την Veracode και τον OWASP με σκοπό την βελτίωση του επιπέδου της ασφάλειας και την ανάπτυξη ασφαλών εφαρμογών.

Ευχαριστίες

Με την ολοκλήρωση της πτυχιακής μου εργασίας, θα ήθελα να ευχαριστήσω όλους τους ανθρώπους οι οποίοι βοήθησαν στην περάτωση αυτής της εργασίας. Θα ήταν παράλειψη να μην αναφερθώ σε όλους εκείνους που μου συμπαραστάθηκαν σε αυτήν την προσπάθεια. Κατά κύριο λόγο, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου από το Τμήμα Ψηφιακών Συστημάτων, κ. Κωνσταντίνο Λαμπρινουδάκη, ο οποίος με υποστήριξε και με καθοδήγησε καθ' όλη τη διάρκεια της πτυχιακής εργασίας. Θα ήθελα να ευχαριστήσω ιδιαιτέρως τους γονείς μου, Γεώργιο-Νικηφόρο-Σταμάτη και Αγγελική για την εμπιστοσύνη και την υποστήριξη τους σε κάθε επιλογή μου. Επίσης θα ήθελα να ευχαριστήσω τους Άρη Καράμ, Ηλιάνα Δανέζη, Στέργιο Νικολαρέα, Βασιλική Αρέθα, Οδυσσέα Παπαδέα και Ελένη Παπαδέα για την βοήθεια και την απεριόριστη ψυχολογική υποστήριξη και κατανόηση που μου παρείχαν όποτε αυτή χρειάστηκε.

Αθήνα , Μάιος 2013

Παπαδέας Γ.Ν.Σ. Δημήτριος

Περιεχόμενα

Εισαγωγή

Η εξέλιξη των δικτύων των κινητών τηλεφώνων με την άφιξη των 3G δικτύων σε συνδυασμό με την ανάπτυξή της επιστήμης των υπολογιστών συνετέλεσε στην γέννηση των έξυπνων τηλεφώνων (Smartphones). Στα εν λόγω τηλέφωνα η επικοινωνία δεν υφίσταται πλέον μόνο μέσω φωνής ή μηνυμάτων (sms) αλλά και μέσω Διαδικτύου. Το γεγονός αυτό κατέστησε τα Smartphones εξαιρετικά δημοφιλή στον κόσμο. Στις μέρες μας, πολλές προσωπικές αλλά και επιχειρηματικές συναλλαγές πραγματοποιούνται μέσω Smartphone συσκευών. Οι οποίες έχουν σχεδόν τις ίδιες δυνατότητες με έναν επιτραπέζιο ή φορητό υπολογιστή. Από την στιγμή που σημαντικοί όγκοι πληροφορίας συναλλάσσονται όπως στους υπολογιστές γίνεται κατανοητό ότι τα Smartphones έχουν αντίστοιχα οφέλη αλλά και κινδύνους.
Η πλατφόρμα Android της Google αποτελεί το πιο δημοφιλές λειτουργικό σύστημα σε αυτές τις κινητές συσκευές. Το μερίδιο 37.23%[1] των κινητών συσκευών ανά τον κόσμο έχουν σαν λειτουργικό σύστημα το Android, με 800.000 επίσημες εφαρμογές και χιλιάδες ανεπίσημες. Παράλληλα όμως το Android κατέχει και την πρωτιά στους ιούς στις κινητές συσκευές με ποσοστό της τάξης 64.1% (2012) και αύξηση κατά 18% από το 2011. Με τους ιούς αυτούς να εξελίσσονται σε όλο και πιο ευφυείς, πιο επικίνδυνοι και πιο αποτελεσματικοί η ανάγκη για προστασία των δομών της εμπιστευτικότητας, της ακεραιότητας και της διαθεσιμότητας της πληροφορίας που διακινείται είναι πιο επιτακτική από κάθε άλλη φορά από ποτέ. Για να αντιμετωπιστούν οι ιοί αποτελεσματικά πρέπει πρέπει να μελετηθούν σε βάθος ώστε να προσδιοριστούν οι τεχνοτροπίες τους, οι σκοποί τους, τι ευπάθειες χρησιμοποιούν και πώς τις εκμεταλλεύονται. Όπως επίσης πρέπει να κατανοηθούν οι στόχοι τους καθώς επίσης και ο τρόπος διάδοσης τους. Οι παραπάνω ενέργειες, απαιτούν την κατανόηση του τρόπου λειτουργίας τους διάδοσης και εφαρμογής τους.
Σε ένα συνεχόμενα εναλλασσόμενο περιβάλλον οι μηχανικοί ασφάλειας καλούνται να δώσουν βιώσιμες λύσεις σε αυτό το κρίσιμο ζήτημα. Ο σκοπός της εργασίας αυτής είναι η μελέτη των ιών στο Android και οι τρόποι ανάλυσης τους.

1 37.23% global stats Top 8 mobile operating system
http://gs.statcounter.com/#mobile_os-ww-monthly-201203-201303

Κεφάλαιο 1 Android

Εικόνα 1, Το Android

1.1 Εισαγωγή στο Android

Το **Android [1]** είναι λειτουργικό σύστημα για συσκευές κινητής τηλεφωνίας το οποίο τρέχει τον Linux πυρήνα 2.6. Αρχικά αναπτύχθηκε από την Google και αργότερα από την Open Handset Alliance[2]. Επιτρέπει στους κατασκευαστές λογισμικού να συνθέτουν κώδικα με την χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας την συσκευή μέσω βιβλιοθηκών λογισμικού ανεπτυγμένων από την Google.

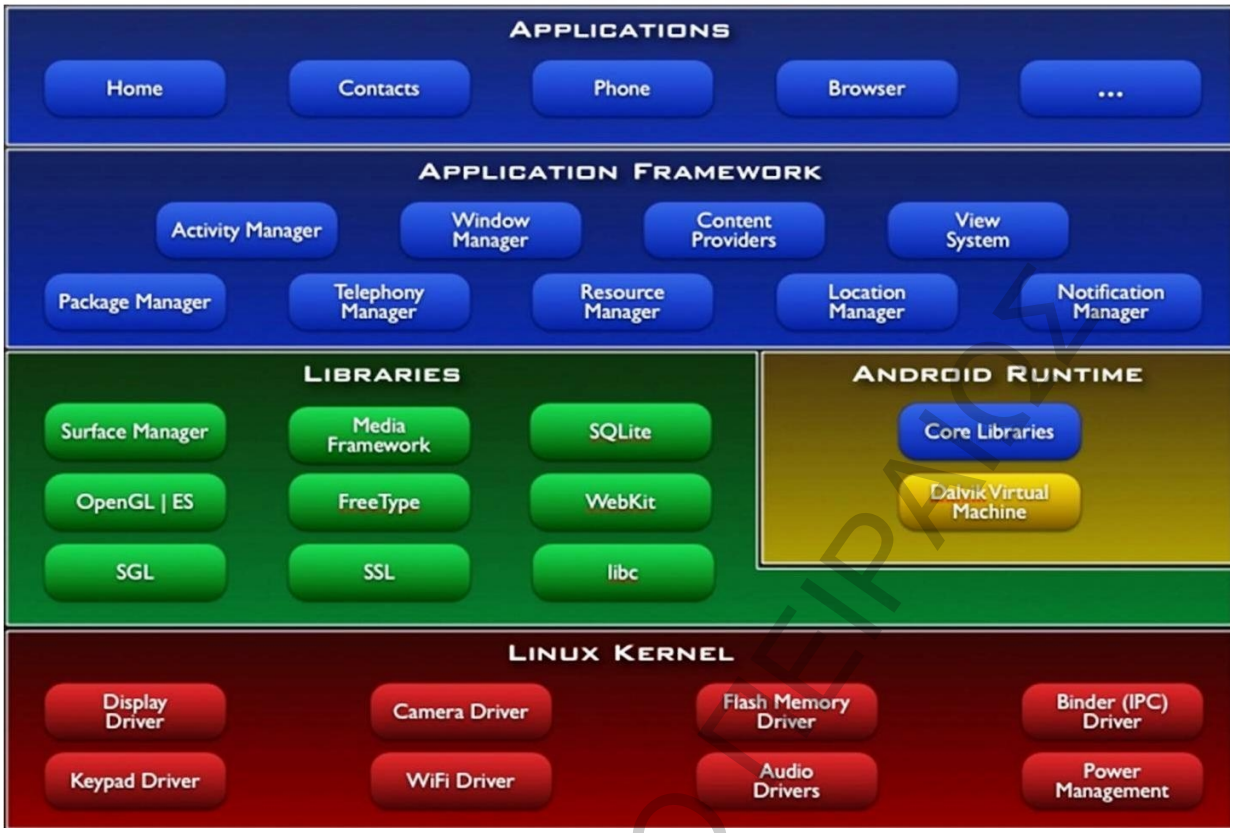
Η πρώτη παρουσίαση της πλατφόρμας Android έγινε στις 5 Νοεμβρίου 2007, παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού Open Handset Alliance, μιας κοινοπραξίας 48 τηλεπικοινωνιακών εταιριών, εταιριών λογισμικού καθώς και κατασκευής hardware, οι οποίες είναι αφιερωμένες στην ανάπτυξη και εξέλιξη ανοιχτών προτύπων στις συσκευές κινητής τηλεφωνίας. Η Google δημοσίευσε το μεγαλύτερο μέρος του κώδικα του Android υπό τους όρους του Apache License της ελεύθερης άδειας λογισμικού.

Η αρχιτεκτονική του λειτουργικού:

Η πλατφόρμα του λειτουργικού Android χωρίζεται στα ακόλουθα επίπεδα, των οποίων η λειτουργία και ιδιότητες πρόκειται στη συνέχεια να επεξηγηθούν :

1. Εφαρμογές (Applications)
2. Πλαίσιο εφαρμογών (Application Framework)
3. Βιβλιοθήκες (Libraries)
4. Βάση του λειτουργικού (Android Runtime)
5. Πυρήνας του λειτουργικού (Linux Kernel)

2 www.openhandsetalliance.com/



Εικόνα 2. Τα επίπεδα του της πλατφόρμας του Android

Οι **Εφαρμογές (Applications)** είναι μια δέσμη προγραμμάτων, όπως ο διαχειριστής SMS, το ημερολόγιο, το web-browser, η διαχείριση επαφών κ.ά, που ενσωματώνονται ως βασικά προγράμματα στην Android συσκευή.

Το **Πλαίσιο εφαρμογών (Application Framework)** έχει σχεδιαστεί για να προωθηθεί η επαναχρησιμοποίηση των στοιχείων, έτσι ώστε κάθε εφαρμογή να μπορεί να εξάγει τις διασυνδέσεις της και να αλληλεπιδρά με άλλες εφαρμογές. Οι αλληλεπιδράσεις αυτές μπορεί να είναι δραστηριότητες (Activities), παροχή περιεχομένου, διαχείριση δραστηριότητας κτλ.

Οι **Βιβλιοθήκες (Libraries)** προορίζονται για τους προγραμματιστές και περιλαμβάνουν ένα σύστημα που βασίζεται στη βιβλιοθήκη BSD derived System-C.

Η **Βάση του λειτουργικού** (Android Runtime) παρέχει τα χαρακτηριστικά γνωρίσματα προγραμματισμού Java και την εικονική μηχανή.

Το Android χρησιμοποιεί τον **Πυρήνα (Kernel)** Linux 2,6 για τις διαδικασίες χαμηλού επιπέδου του συστήματος, όπως η διαχείριση της μνήμης, των διαδικασιών και της στοίβας του δικτύου.

1.2 Σύστημα αρχείων

Το λειτουργικό Android επιτρέπει την αποθήκευση των δεδομένων τοπικά ως αρχεία. Χρησιμοποιεί επίσης ένα σύστημα αρχείων [2] για την αποθήκευση και για το χειρισμό των ρυθμίσεων της κάθε εφαρμογής (Προτιμήσεις) αλλά και για την βάση δεδομένων SQLite. Για κάθε εφαρμογή, το λειτουργικό Android δημιουργεί έναν κατάλογο στο "data/data/-το όνομα του πακέτου της εφαρμογής- ". Τα αρχεία αποθηκεύονται στον φάκελο “files” και οι ρυθμίσεις των εφαρμογών στον φάκελο "shared_prefs" με την μορφή XML αρχείου. Εκεί η κάθε εφαρμογή έχει τον δικό της υποφάκελλο καθώς και δικαιώματα τροποποίησης, ανάγνωσης αλλά και εκχώρησης αυτών των δικαιωμάτων σε 3^{ες} εφαρμογές. Η πρόσβαση στο σύστημα αρχείων γίνεται μέσω του πακέτου της Java: `java.io`

Το Android επίσης παρέχει βοηθητικές κλάσεις για τη δημιουργία και την πρόσβαση σε νέα αρχεία και καταλόγους. Για παράδειγμα η μέθοδος getDir(String, int) θα δημιουργήσει ή θα παρέχει πρόσβαση, η openFileInput(String s) θα ανοίξει ένα αρχείο για είσοδο και η openFileOutput(String s, int) θα δημιουργήσει ένα αρχείο

Το όρισμα int καθορίζει τα δικαιώματα :

- MODE_PRIVATE – Καμία πρόσβαση από άλλες εφαρμογές
- MODE_WORLD_READABLE – Μονό ανάγνωση από άλλες εφαρμογές
- MODE_WORLD_WRITABLE - Μονό εγγραφή από άλλες εφαρμογές
- MODE_WORLD_READABLE | MODE_WORLD_WRITABLE – Ανάγνωση και εγγραφή

1.2.1 Περιορισμοί της σχεδίασης που επιβάλλονται από την πλατφόρμα

Η πλατφόρμα Android δημιουργήθηκε για συσκευές με περιορισμένη επεξεργαστική ισχύ, μνήμη και αποθήκευση. Οι ελάχιστες απαιτήσεις για μια συσκευή Android είναι ανάλογες με την έκδοση του λειτουργικού.

1.3 Android Phone Μνήμη και αποθηκευτικά μέσα

Μία συσκευή Android μπορεί να έχει διαφορετικούς τύπους μνήμης και διαφορετικά αποθηκευτικά μέσα. Όπως για παράδειγμα: το LG Optimus το οποίο έχει μνήμη ROM 512MB, μνήμη RAM 512MB και δέχεται εξωτερικές μνήμες micro SDHC slot επεκτάσιμη έως 32GB.

1.3.1 Η μνήμη RAM

Η μνήμη τυχαίας προσπέλασης (Random Access Memory) η οποία βρίσκεται σχεδόν σε όλους τους υπολογιστές και αλλά στα smart phones. Στην μνήμη RAM φορτώνονται και εκτελούνται όλα τα προγράμματα γρήγορα. Μετά τον τερματισμό της συσκευής τα δεδομένα εξασθενούν και ύστερα από κάποιο χρονικό διάστημα σβήνονται.

1.3.2 Η μνήμη ROM

“Η μνήμη μόνο για ανάγνωση” (Read-Only Memory). Τα δυαδικά ψηφία που είναι αποθηκευμένα σε αυτόν τον τύπο μνήμης δεν μπορούν ποτέ να τροποποιηθούν από την στιγμή που φεύγουν από το εργοστάσιο, το νέο είδος ROM είναι το EEPROM, που αντικατέστησε ROM, το

οποίο επιτρέπει την αντικατάσταση των δεδομένων όταν εφαρμόζεται υψηλότερη τάση έτσι ώστε να μπορεί να εξυπηρετήσει ενημερώσεις. Το EEPROM αντικαθίστανται σταδιακά από την Flash Memory ως επί το πλείστον, η οποία επιτρέπει την αντικατάσταση εύκολα. Πλέον η μνήμη ROM σε ένα smart phone είναι Flash Memory. Η μνήμη ROM στις τηλεφωνικές συσκευές συχνά χωρίζεται σε πολλά διαμερίσματα (partitions). Στα Android, το ένα partition είναι για την εγκατάσταση του λειτουργικού συστήματος, το οποίο συνήθως προστατεύεται και δεν επιτρέπεται η εγγραφή σε αυτό. Το Rooting σημαίνει η λήψη όλων των δικαιωμάτων των χρηστών του λειτουργικού συστήματος, συνεπώς με το Rooting επιτρέπεται η ανάγνωση και η εγγραφή στο διαμέρισμα του λειτουργικού συστήματος. Τα παραπάνω επιτρέπουν στον χρήστη, που έχει πάρει τα δικαιώματα αυτά, να εκτελέσει διαχειριστικές διεργασίες όπως να τροποποιήσει το λειτουργικό ή ακόμα και να εγκαταστήσει ένα διαφορετικό στην θέση του. Το άλλο partition είναι για τα δεδομένα των χρηστών, συμπεριλαμβανομένων των ληφθέντων εφαρμογών και των αποθηκευμένων δεδομένων τους. Αυτό το partition δεν χρησιμοποιείται μόνο για για ανάγνωση. Το τμήμα αυτό ονομάζεται επίσης εσωτερική μνήμη του τηλεφώνου. Δηλαδή κάτι ανάλογο με τον C: drive στα Windows. Τέλος, ο φάκελος των Windows είναι αόρατος και στο partition του OS.

Για παράδειγμα το LG Optimus, έχει αρχικά διαθέσιμο χώρο 180MB. Ο διαθέσιμος χώρος ελαττώνεται με την εγκατάσταση νέων εφαρμογών. Και αυτό φαίνεται στο : **Settings | SD card & phone storage settings | Internal Phone Storage.** Όταν ο χώρος γίνει ελάχιστος δεν επιτρέπεται η περαιτέρω εγκατάσταση νέων εφαρμογών. Πλέον στα σύγχρονα smart phones η εσωτερική μνήμη του τηλεφώνου είναι συνήθως μεγαλύτερη από 1 GB και συνδυάζεται με εξωτερικές κάρτες μνήμης.

1.3.3 Εξωτερικές κάρτες μνήμης Micro SD / SDHC

Η χρήση εξωτερικών καρτών μνήμης (**[3], [4]**) αποτελεί τη μόνη δυνατότητα αύξησης του αποθηκευτικού χώρου στη συσκευή. Αυτές μοιάζουν με τον εξωτερικό σκληρό δίσκο ενός υπολογιστή. (Για παράδειγμα το Optimus από κατασκευής έχει μια τοποθετημένη κάρτα SD 2GB η οποία μπορεί να αντικατασταθεί με οποιαδήποτε κάρτα έως 32GB). Η micro SD είναι τοποθετημένη στον φάκελο / mnt / sdcard. Και οι ρυθμίσεις της είναι **Settings | SD card & phone storage settings | SD card**.

Σε αυτή την κάρτα μπορούν να αποθηκευθούν δεδομένα και αρχεία ανεξάρτητα όπως πχ. ταινίες, μουσική, φωτογραφίες. Η κάρτα SD μπορεί επίσης να χρησιμοποιηθεί και για μεταφορά από SD κάρτα σε SD κάρτα ή από συσκευή σε συσκευή. Μετά το Android 2.1 παρέχεται η δυνατότητα σε ένα μέρος μίας εγκατεστημένης εφαρμογής να μπορεί να μετακινηθεί από την εσωτερική στην εξωτερική μνήμη, εξοικονομώντας με αυτό τον τρόπο πολύτιμο εσωτερικό χώρο στην εσωτερική μνήμη. Βέβαια δεν είναι δυνατό αυτό για όλες τις εφαρμογές ή τουλάχιστον για κάποια συγκεκριμένα τμήματα εφαρμογών να μπορούν να μετακινηθούν από την εσωτερική μνήμη στην κάρτα SD. Έτσι, η προσθήκη μιας μεγαλύτερης κάρτας SD δεν είναι απαραίτητο ότι θα βοηθήσει πολύ αν εσωτερική μνήμη είναι γεμάτη. Επίσης υπάρχουν εφαρμογές που παρέχουν πολλές δυνατότητες επεξεργασίας και ενημέρωσης σχετικά με τις εξωτερικές κάρτες.

1.4 Η εικονική μηχανή Dalvik

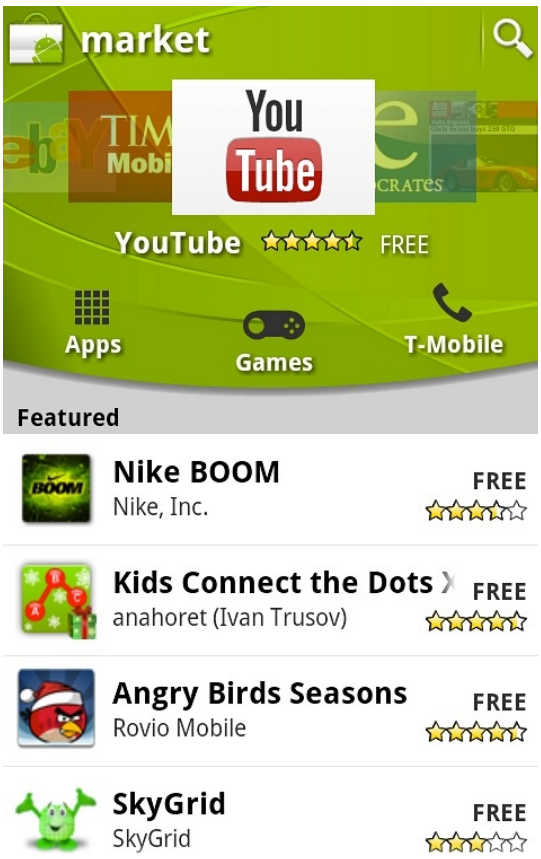
Δεδομένου ότι η βάση του λειτουργικού Android (application runtime) πρέπει να υποστηρίζει ένα ευρύ σύνολο συσκευών σε συνδυασμό με το ότι οι εφαρμογές πρέπει να είναι απομονωμένες (sandboxed) για λόγους ασφάλειας, επιδόσεων και αξιοπιστίας μια εικονική μηχανή φαίνεται σαν μια προφανής επιλογή. Ωστόσο, ένα λειτουργικό που βασίζεται σε μια εικονική μη-

χανή δεν είναι απαραίτητο ότι θα μπορέσει να ισορροπήσει τις απαιτήσεις αυτές με περιορισμένη ταχύτητα επεξεργαστή και μνήμη RAM τις οποίες έχουν οι περισσότερες φορητές συσκευές.

Για την αντιμετώπιση όλων αυτών των κάπως αντικρουόμενων απαιτήσεων η Google δημιούργησε την εικονική μηχανή Dalvik **[5]**. Συνοπτικά το παραπάνω εγχείρημα στοχεύει στην υλοποίηση ενός περιβάλλοντος εκτέλεσης εφαρμογών με τον περιορισμό ότι κάθε εφαρμογή Android τρέχει σαν αυτόνομη διαδικασία, με το δικό της τμήμα της εικονικής μηχανής Dalvik. Το Dalvik έχει γραφτεί έτσι ώστε μια συσκευή να έχει την δυνατότητα εκτέλεσης πολλαπλών Vms(Virtual machine). Το Dalvik VM εκτελεί εκτελέσιμα αρχεία Dalvik (.DEX) μορφής, τα οποία έχουν βελτιστοποιηθεί για ελάχιστο ίχνος μνήμης. Το VM είναι βασισμένο στα συστήματα καταχωρητών και τρέχει προγράμματα γραμμένα σε Java τα οποία έχουν μετατραπεί σε μορφή .dex. Το Dalvik VM βασίζεται στον πυρήνα Linux για την υποκείμενη λειτουργικότητα, όπως threading και χαμηλού επιπέδου διαχείρισης μνήμης. Δεδομένου ότι κάθε εφαρμογή τρέχει στη δική της διαδικασία μέσα στην ίδια εικονική μηχανή της, όχι μόνο θα πρέπει η λειτουργία των πολλαπλών VMs να είναι αποτελεσματική αλλά και δημιουργία νέων VMs πρέπει να είναι γρήγορη.

Το Google Play Android market

Το Google Play (πρώην Android Market) [7] είναι ένα online κατάστημα λογισμικού που αναπτύχθηκε από την Google για συσκευές Android. Αποτελεί μία προεγκατατεστειμμένη εφαρμογή που ονομάζεται «Market» στις περισσότερες συσκευές Android. Η τελευταία επιτρέπει στους χρήστες να περιηγούνται και να κατεβάζουν άλλες εφαρμογές που δημοσιεύονται από τρίτους προγραμματιστές. Οι χρήστες μπορούν επίσης να αναζητούν και να διαβάζουν αναλυτικές πληροφορίες σχετικά με τις εφαρμογές στην ιστοσελίδα του Android Market.



Εικόνα 4, Το Android market

Εικόνα 3, Το νέο Google play

Αναπτύσσοντας εφαρμογές Android

Για να αναπτύξει ένας προγραμματιστής εφαρμογές Android χρειάζεται να γνωρίζει Java όπως επίσης και τα εργαλεία Android SDK **[9]**, Eclipse με το plug-inAndroid Development Tool. Για την δημοσίευση εφαρμογών στο Android Market απαιτείται ένας λογαριασμός Android marketplace developer ο οποίος προϋποθέτει ένα τέλος 25$.

Εικόνα 5, Μήνυμα εγγραφής

Εικόνα 6, Η εγγραφή ολοκληρώθηκε

Οι εφαρμογές πρέπει να πληρούν μερικές προϋποθέσεις προκειμένου να δημοσιευθούν στο Android Market. Απαιτήσεις που αφορούν κυρίως το πώς αυτές οι προϋποθέσεις ταξινομούνται

στις κατηγορίες εφαρμογών και πως κρυπτογραφούνται. Να σημειωθείς πως κάθε εφαρμογή υποβάλλεται με ένα κύριο εικονίδιο και κάποια ενδεικτικά screenshots.

Κεφάλαιο 2 Η Ασφάλεια στο Android

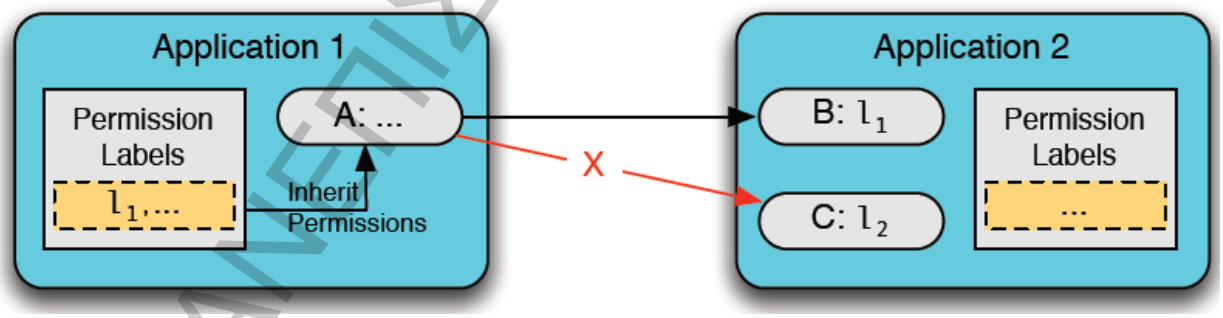
2.1 Η αρχιτεκτονική ασφάλειας του Android

Το Android είναι ένα λειτουργικό σύστημα με πυρήνα Linux, που έχει προγραμματιστεί σε Java και έχει ενισχυθεί με τους δικούς του μηχανισμούς ασφαλείας οι οποίοι είναι προσαρμοσμένοι για ένα περιβάλλον κινητών συσκευών.

Το Android OS συνδυάζει χαρακτηριστικά, όπως αποτελεσματική κοινόχρηστη μνήμη, multi-tasking, αναγνωριστικά χρήστη Unix (UIDs[3]) καθώς και δικαιώματα αρχείων. Το μοντέλο ασφάλειας που προκύπτει είναι πολύ περισσότερο σαν ένας multi-user διακομιστής παρά σαν ένα sandbox που συναντάται στην J2ME ή στις πλατφόρμες Blackberry. Σε αντίθεση με ένα desktop περιβάλλον όπου οι εφαρμογές ανήκουν σε έναν χρήστη και λειτουργούν με το ίδιο UID, στο Android οι εφαρμογές είναι τελείως απομονωμένες και αποθηκεύονται χωριστά η μία από την άλλη. Οι εφαρμογές Android εκτελούνται ως ξεχωριστές διαδικασίες με διαφορετικά UIDs και διαφορετικά δικαιώματα η καθεμία. Μια εφαρμογή δεν μπορεί συνήθως ούτε να διαβάσει ούτε να γράψει δεδομένα ή κώδικα σε κάποια άλλη και για την ανταλλαγή δεδομένων μεταξύ των εφαρμογών πρέπει να γίνει ρητή εκχώρηση συγκεκριμένου δικαιώματος. Αυτή η ιδιότητα ονομάζεται Inter Component Communication (ICC). Το γραφικό περιβάλλον Android έχει κάποια νέα χαρακτηριστικά ασφαλείας που βοηθούν στην υποστήριξη αυτής της απομόνωσης. Οι mobile πλατφόρμες αποκτούν ολοένα και περισσότερη σημασία και έχουν σύνθετες απαιτήσεις συμπεριλαμβανομένων των κανονισμών συμμόρφωσης. Το Android υποστηρίζει τη δημιουργία εφαρμογών που χρησιμοποιούν τις λειτουργίες του τηλεφώνου, ενώ παράλληλα προστατεύει τους χρήστες ελαχιστοποιώντας τις συνέπειες των σφαλμάτων, προστατεύοντας τους από κακόβουλο λογισμικό.

Η απομόνωση των διαδικασιών του Android εξαλείφει την ανάγκη για περίπλοκα αρχεία διαμόρφωσης πολιτικής για sandbox[4]. Αυτό δίνει την ευελιξία στις εφαρμογές να χρησιμοποιούν τον πηγαίο τους κώδικα χωρίς να διακυβεύεται η ασφάλεια του λειτουργικού ή κάποια τροποποίηση δικαιωμάτων. ([11])

Το ICC του Android συνοψίζεται στην ακόλουθη εικόνα:



Εικόνα 7, Το ICC

Τα δικαιώματα στο Android ορίζουν προκαθορισμένες ρητές άδειες που έχουν δοθεί σε μία εφαρμογή και της επιτρέπουν πρόσβαση σε πόρους όπως πχ να χρησιμοποιήσει το GPS ή την κάμερα ή την δυνατότητα τηλεφωνικών κλήσεων. Όταν εγκαθίσταται μία εφαρμογή, της δίνεται ένα μοναδικό UID και η εφαρμογή στο εξής θα εκτελείται πάντα με αυτό το UID στη συγκεκριμένη συσκευή. Το UID της εφαρμογής χρησιμοποιείται για την προστασία των δεδομένων και γι' αυτό οι προγραμματιστές πρέπει να είναι σαφείς σχετικά με την ανταλλαγή δεδο-

3 Ακρώνυμο του User Identifier

4 Πίο αναλυτικά για το Sandbox στο Κεφάλαιο 4

μένων με άλλες εφαρμογές. Οι εφαρμογές γενικά μπορούν να διασκεδάσουν τους χρήστες με τα γραφικά,την μουσική ή και τη έναρξη άλλων μικρο-εφαρμογών χωρίς ειδικά δικαιώματα.
Το κακόβουλο λογισμικό είναι μία "ατυχής πραγματικότητα" στις δημοφιλείς πλατφόρμες. Τις αρνητικές επιπτώσεις του κακόβουλου λογισμικού προσπαθεί να τις εξαλείψει το Android, μέσω των χαρακτηριστικών που διαθέτει. Ωστόσο, ακόμη και χωρίς δικαιώματα ένα κακόβουλο λογισμικό έχει τη δυνατότητα να εγκατασταθεί σε μια συσκευή Android (συχνά παριστάμενο μια άλλη χρήσιμη εφαρμογή) περιορίζοντας έστω και προσωρινά την εμπειρία του χρήστη. Οι χρήστες σε αυτή την ατυχή περίπτωση πρέπει να εντοπίσουν και να εξαλείψουν την εχθρική εφαρμογή. Το Android βοηθά τους χρήστες να το κάνουν αυτό ελαχιστοποιώντας την έκταση της κακοποίησης, όσο αυτό είναι δυνατό, απαιτώντας την άδεια του χρήστη για τα προγράμματα που αιτούνται επικίνδυνες ενέργειες όπως:

- Απευθείας κλήση κλήσεων (στην οποία μπορεί να υπάρχουν περαιτέρω χρεώσεις).
-Αποκάλυψη ή παραβίαση προσωπικών δεδομένων του χρήστη
- Καταστροφή δεδομένων, διευθύνσεων, επαφών email, κλπ.

Γενικά η ενδεδειγμένη αντίδραση ενός χρήστη απέναντι σε ένα πρόβλημα, σε κάποιο σφάλμα στον κώδικα ή σε κάποιο κακόβουλο λογισμικό θα πρέπει είναι απλά η απεγκατάσταση, εάν το ελαττωματικό λογισμικό διακόπτει την ομαλή λειτουργία του τηλεφώνου. (Είναι γεγονός ότι ο χρήστης δεν μπορεί να το απ' εγκαταστήσει, δύναται όμως να επανεκκινήσει το τηλέφωνο -προαιρετικά σε ασφαλή λειτουργία, ώστε να τρέχει μόνο ο κώδικας του λειτουργικού- και στη συνέχεια να αφαιρέσει το μη επιθυμητό λογισμικό εφόσον πλέον δεν εκτελείται).

2.1.1 Ευθύνες ασφαλείας των προγραμματιστών

Οι προγραμματιστές που γράφουν εφαρμογές για το Android πρέπει να εξετάσουν τον κωδικά τους για το πώς η εφαρμογή θα κρατήσει τους χρήστες ασφαλείς απέναντι καθώς και το πώς θα διαχειριστεί την περιορισμένη μνήμη, την επεξεργαστική ισχύ αλλά και την ενέργεια της μπαταρίας. Οι προγραμματιστές οφείλουν να μεριμνούν για την ασφάλεια των δεδομένων που εισάγουν οι χρηστές στη συσκευή μέσω μηχανισμών ασφαλείας. Επίσης πρέπει να διαφυλάσσεται η εφαρμογή από την πρόσβαση malware (ειδικά στα δικαιώματα ή τα προνόμια της). Το πως θα επιτευχθεί αυτό εν μέρει σχετίζεται με τα χαρακτηριστικά της πλατφόρμας τα οποία χρησιμοποιεί η εφαρμογή, καθώς και τις τυχόν επεκτάσεις που έχουν γίνει στην πλατφόρμα της εκάστοτε διανομής Android. Ένα από τα σημαντικότερα πράγματα που οφείλει να καταλάβει κανείς για το μοντέλο ασφάλειας στο Android, όπως προαναφέρθηκε, είναι ότι κάθε εφαρμογή τρέχει με διαφορετικό UID. Συνήθως σε έναν επιτραπέζιο υπολογιστή κάθε χρήστης διαθέτει ένα μοναδικό UID και όλες οι εφαρμογές εκτελούνται ως διεργασίες με το UID του χρήστη. Στο Android, το σύστημα δίνει UID σε κάθε εφαρμογή και όχι κάθε χρήστη. Για παράδειγμα, κατά την εκκίνηση μίας νέας διεργασίας (δηλαδή την έναρξη μιας δραστηριότητας - *Activity*[5]), η νέα διαδικασία δεν πρόκειται να εκτελεστεί ως διεργασία του χρήστη, αλλά αυτόνομα με δική της ταυτότητα. Είναι σημαντικό σε περιπτώσεις που η εφαρμογή ή κάποια δραστηριότητα εκκινείται με λάθος παραμέτρους[6] ο προγραμματιστής να έχει εξασφαλίσει ότι η εφαρμογή δεν θα βλάψει το λειτουργικό ούτε την συσκευή και γενικά θα αποφεύγεται οποιαδήποτε μη επιθυμητή τροποποίηση. Κάθε εφαρμογή μπορεί να ζητήσει από τον Διαχειριστή Δραστηριότητας (Activity Manager) να ξεκινήσει σχεδόν οποιαδήποτε άλλη εφαρμογή η οποία θα τρέχει με το UID της εφαρμογής [9]. Ευτυχώς, τα μη αξιόπιστα σημεία εισόδου σε μία εφαρμογή περιορίζονται στα ιδιαίτερα χαρακτηριστικά της πλατφόρμας που θα επιλέγουν από τον προγραμματιστή να χρησιμοποιηθούν ενώ είναι και δυνατή η διασφάλιση τους με συνεπή τρόπο.

5 Κεφάλαιο 2.2.2 Activities

6 Οι εφαρμογές Android συνήθως δεν έχουν καθόλου παραμέτρους.

Οι Android εφαρμογές δεν έχουν την παραδοσιακή main μέθοδο από οπού ξεκινάν να εκτελούνται. Αντί αυτού, το αρχικό σημείο εισόδου τους βασίζεται στην καταχώρηση των *Activites*, *Services*, *BroadcastReceivers και ContentProviders* [7]στο σύστημα. Το Android απαιτεί από τους προγραμματιστές να υπογράψουν ψηφιακά τον κωδικά τους.
Η υπογραφή του Android κώδικα χρησιμοποιεί συνήθως αυτό-υπογεγραμμένο(self signed) πιστοποιητικό, το οποίο οι προγραμματιστές μπορούν να δημιουργήσουν χωρίς τη βοήθεια ή την άδεια κάποιας έμπιστης οντότητας. Ένας λόγος για αυτήν την υπογραφή κώδικα είναι η δυνατότητα στους προγραμματιστές να ενημερώνουν την εφαρμογή τους χωρίς να δημιουργούν περίπλοκα interfaces και δικαιώματα. Οι εφαρμογές που έχουν υπογραφεί με το ίδιο κλειδί (και ως εκ τούτου είναι από τον ίδιο δημιουργό) μπορούν να αιτηθούν εκτέλεση με το ίδιο UID. Αυτό επιτρέπει στους προγραμματιστές να αναβαθμίσουν ή να επιδιορθώσουν το λογισμικό τους εύκολα, συμπεριλαμβανομένης της αντιγραφής δεδομένων από υπάρχουσες εκδόσεις. Η διαδικασία της υπογραφής είναι διαφορετική από ένα κανονικό Jar αρχείο ή μία Authenticode υπογραφή[8], ωστόσο, όμως η πραγματική ταυτότητα του δημιουργού δεν είναι κατ' ανάγκη επικυρωμένη από μία τρίτη έμπιστη οντότητα. Οι προγραμματιστές κερδίζουν καλή φήμη δημιουργώντας καλά προϊόντα και τα πιστοποιητικά αυτά αποδεικνύουν την πατρότητα των έργων τους. Ένας προγραμματιστής δεν γίνεται έμπιστος μόνο και μόνο επειδή πλήρωσε λίγα χρήματα σε κάποια αρχή. Αυτή η τακτική έχει υιοθετηθεί από την Google και μπορεί κάλλιστα να πετύχει, παρόλο που δεν θα ήταν τεχνικά δύσκολο να προστεθούν αξιόπιστοι κανόνες υπογράφων σε κάποια διανομή Android, αν αποδειχθεί επιθυμητή η δυνατότητα έμπιστων προγραμματιστών.

2.1.2 Δικαιώματα του χρήστη στο Android

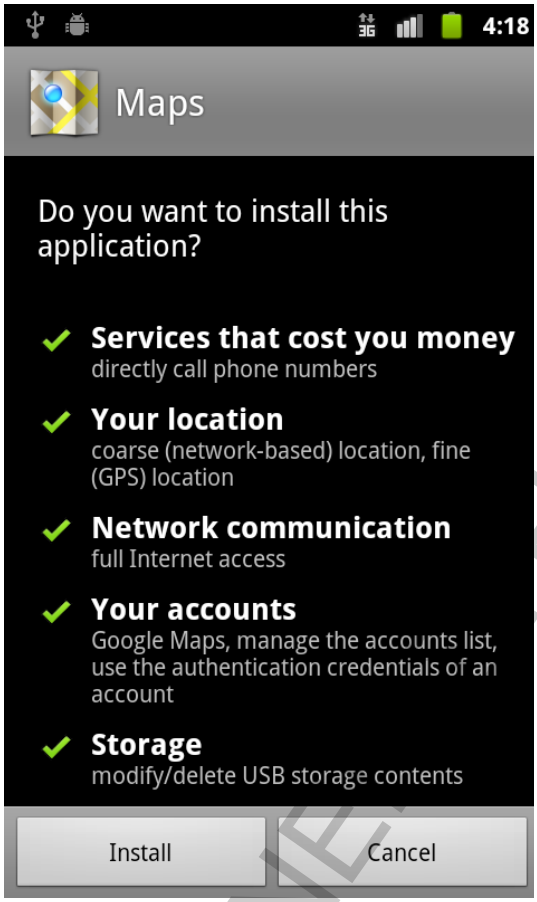
Οι εφαρμογές χρειάζονται την έγκριση από τον χρήστη για να κάνουν τις ενέργειες για τις οποίες έχουν σχεδιαστεί και αναπτυχθεί, ενέργειες όπως η αποστολή μηνυμάτων SMS, η χρήση της κάμερας ή η πρόσβαση στη βάση δεδομένων των επαφών του ιδιοκτήτη. Το Android χρησιμοποιεί manifest permissions για να καθορίσει τι επιτρέπει ο χρήστης να κάνει μία εφαρμογή. Η ανάγκη χρήσης ενός δικαιώματος που απαιτεί μία εφαρμογή δηλώνεται στο αρχείο *AndroidManifest.xml*. Ο χρήστης αποδέχεται ή όχι την χορήγηση αυτών των δικαιωμάτων κατά την εγκατάσταση (Εικόνες 8-9). Εκεί δίνεται η δυνατότητα στους χρήστες να αποφασίσουν εάν θα εμπιστευθούν το λογισμικό με βάση τις αξιολογήσεις, τη φήμη του δημιουργού του έργου, καθώς και τα δικαιώματα που απαιτούνται. Αποφασίζοντας την αποδοχή ή μη των όρων χρήσης κατά την εγκατάσταση, επιτρέπεται στον χρήστη, να επικεντρώνεται στην λειτουργικότητα παρά στην ασφάλεια. Τα δικαιώματα μερικές φορές καλούνται ως manifest permissions ή Android δικαιώματα για τη διάκρισή τους από τα δικαιώματα αρχείων. Για να είναι χρήσιμα τα δικαιώματα θα πρέπει να συνδέονται με κάποιο νόημα που ο χρήστης κατανοεί. Για παράδειγμα, μια εφαρμογή χρειάζεται την άδεια READ_CONTACTS (να διαβάσει τις επαφές του χρήστη στη συσκευή). Μια εφαρμογή διαχείρισης επαφών χρειάζεται την άδεια READ_CONTACTS, (σε αντίθεση με ένα παιχνίδι με τουβλάκια που δεν χρειάζεται αυτό το δικαίωμα) αλλά αντίστοιχα μπορεί να χρειάζεται το δικαίωμα πχ: VIBRATE (εάν δονείται η συσκευή όταν ο παίκτης χάνει) ή το δικαίωμα INTERNET (για περιπτώσεις που έχει αναβαθμίσεις ή διαδικτυακή αλληλεπίδραση). Κρατώντας έτσι το μοντέλο απλό, είναι δυνατόν να διασφαλιστεί η χρήση όλων των εσωτερικών διαφορετικών μηχανισμών επικοινωνίας του λειτουργικού (**Android inter-process communication** -IPC[9]) με ένα μόνο είδος άδειας. Η εκκίνηση

7 Κεφάλαιο: 2.2 Δομικά στοιχεία του Android

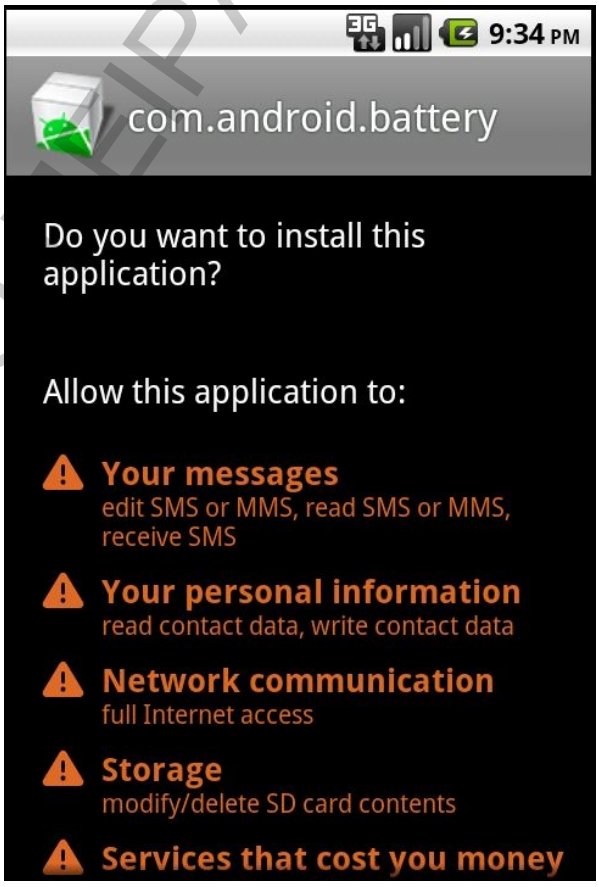
8 Η πολιτική της Microsoft για την υπογραφή κώδικα: http://msdn.microsoft.com/en-us/library/ms537364(VS.85).aspx

9 Οι Android IPC μηχανισμοί δομούν την εσωτερική επικοινωνία των στοιχείων του λειτουργικού.

των *Activities*, η έναρξη ή η σύνδεση με τα *Services*, η πρόσβαση των ContentProviders, η αποστολή και λήψη *Intents* και τα επικαλούμενα interfaces αιτούνται όλα το ίδιο δικαίωμα.
Ως εκ τούτου, οι χρήστες δεν χρειάζεται να καταλάβουν τίποτα περισσότερο από το πως “η εφαρμογή διαχείρισης επαφών πρέπει να μπορεί να διαβάσει τις επαφές και συνεπώς να έχει το δικαίωμα ανάγνωσης επαφών READ_CONTACTS". Συνεπώς οι προγραμματιστές όταν αναπτύσσουν προσαρμοσμένα δικαιώματα οφείλουν να φροντίζουν ώστε το νέο δικαίωμα να έχει συνοπτικό και κατανοητό όνομα σε σχέση με την ιδιότητα του. Μετά την εγκατάσταση, τα δικαιώματα μιας εφαρμογής δεν δύνανται να αλλάξουν. Ελαχιστοποιώντας τα δικαιώματα που χρησιμοποιεί μια εφαρμογή ελαχιστοποιούνται αυτόματα και οι συνέπειες των πιθανών κενών ασφαλείας με αποτέλεσμα οι χρήστες να αισθάνονται καλύτερα κατά την εγκατάστασή της. Όπου βλέπουν τι δικαιώματα αιτείται η εφαρμογή σε ένα παράθυρο παρόμοιο με αυτό που φαίνεται στις εικόνες 8-9. Η εγκατάσταση του λογισμικού μπορεί να κρύβει κινδύνους και οι χρήστες οφείλουν να αποφεύγουν λογισμικό που δεν γνωρίζουν, ειδικά αν αυτό απαιτεί πολλά δικαιώματα.



Εικόνα 8, Δικαιώματα που αιτείται η εφαρμογή Maps



Εικόνα 9, Δικαιώματα που αιτείται μία εφαρμογή διαχείρισης της μπαταρίας

Από την σκοπιά του προγραμματιστή τα δικαιώματα είναι strings που συνδέονται με την εφαρμογή και το UID της. Με την χρήση της μεθόδου checkPermission της κλάσης *Context* (δικαίωμα String, int pid[10], int uid) ελέγχεται προγραμματιστικά εάν μια διαδικασία (και το αντίστοιχο UID) έχουν ένα συγκεκριμένο δικαίωμα, όπως πχ. READ_CONTACTS. Αυτός είναι ένας από τους πολλούς τρόπους να ελέγχονται τα δικαιώματα κατά το χρόνο εκτέλεσης από τους προγραμματιστές. Η άποψη των χρηστών για τα δικαιώματα είναι απλή και συνεπής: Το ιδίωμα για την επιβολή από τους προγραμματιστές είναι ικανοποιητικά κατανοητό. Η Εικόνα 11 δείχνει ένα παράδειγμα για τον καθορισμό ενός νέου προσαρμοσμένου δικαιώματος. Η

10 Pid: ακρώνυμο του Processes Identifier

περιγραφή (description) και η ετικέτα (label) είναι οι πόροι που βοηθούν στον εντοπισμό των δικαιωμάτων από την εφαρμογή.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.novaapps.findevents" android:versionCode="1"
    android:versionName="1.0"
    android:installLocation="preferExternal">
    <uses-sdk android:minSdkVersion="4" />
    <supports-screens
        android:largeScreens="true"
        android:normalScreens="true"
        android:smallScreens="true"
        android:resizeable="true"
        android:anyDensity="true" />

<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_LOCATION_EXTRA_COMMANDS" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />

    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".FindEventsGADroidActivity"
            android:label="@string/app_name" android:configChanges="orientation|keyboardHidden">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <activity android:name="com.phonegap.DroidGap" android:label="@string/app_name"
            android:configChanges="orientation|keyboardHidden">
            <intent-filter>
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Εικόνα 10, Δήλωση δικαιωμάτων στο AndroidManifest.xml

```
file:///home/dimi/Ubuntu One/ΠΤΥΧΙΑΚΗ/Malware apps/tested 19 3/extracted
<permission android:name="com.dimi.android.A_CUSTOM_PERM"
android:description="@string/access_perm_desc" android:protectionLevel="normal"
android:label="@string/access_perm_label"> </permission>
<uses-permission android:name="android.permission.A_CUSTOM_PERM"> </uses-permission>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"> </uses-
permission>
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"> </uses-
permission>
<uses-sdk android:minSdkVersion="4" android:targetSdkVersion="8"> </uses-sdk>
</manifest>
```

Εικόνα 11, Δήλωση ενός προσαρμοσμένου δικαιώματος με όνομα CUSTOM_PERM

Τα Manifest δικαιώματα, όπως φαίνεται στην Εικόνα 10 έχουν μερικές βασικές ιδιότητες:

1) Απαιτούνται δύο περιγραφές κειμένου, μια μικρή ετικέτα κειμένου (label) και μια εκτενέστερη περιγραφή που χρησιμοποιείται για την εγκατάσταση (description).

2) Προαιρετικά ένα εικονίδιο για το δηλωθέν δικαίωμα.

Όλα τα δικαιώματα πρέπει επίσης να έχουν ένα όνομα το οποίο να είναι μοναδικό σε παγκόσμιο επίπεδο[11]. Το όνομα του δικαιώματος είναι το αναγνωριστικό που χρησιμοποιείται από τους προγραμματιστές και είναι η πρώτη παράμετρος για το Context.checkPermission. Τα

11 Παγκόσμιο επίπεδο νοείται όλη η εφαρμογή.

δικαιώματα έχουν επίσης ένα επίπεδο προστασίας που ονομάζεται protectionLevel όπως φαίνεται στον Πίνακα 1. Υπάρχουν έξι επίπεδα προστασίας για τα δικαιώματα [13]:

Επίπεδο	Περιγραφή
Normal	Δικαιώματα για τα χαρακτηριστικά της εφαρμογής των οποίων οι συνέπειες είναι μικρές, όπως για παράδειγμα το VIBRATE, το οποίο επιτρέπει στις εφαρμογές να ενεργοποιούν την δόνηση της συσκευής. Το επίπεδο αυτό είναι κατάλληλο για τη χορήγηση δικαιωμάτων όπου γενικά δεν είναι έντονο το ενδιαφέρον από τους χρήστες και δεν ενημερώνονται για την ύπαρξη αυτών των δικαιωμάτων.
Dangerous	Δικαιώματα όπως WRITE_SETTINGS ή SEND_SMS είναι επικίνδυνα, δεδομένου ότι θα μπορούν να χρησιμοποιηθούν για να επαναρυθμίσουν τη συσκευή ή να επιβαρύνουν τον λογαριασμό του χρήστη με περιττές χρεώσεις. Αυτό το επίπεδο χρησιμοποιείται για να σηματοδοτήσει τα κρίσιμα δικαιώματα της εφαρμογής που οι χρήστες πρέπει να ενδιαφέρονται. Το Android προειδοποιεί τους χρήστες σχετικά με αυτά τα δικαιώματα κατά την εγκατάσταση.
Signature	Ένα δικαίωμα το οποίο χορηγείται από το σύστημα μόνο εάν η αιτούσα εφαρμογή είναι υπογεγραμμένη με το ίδιο πιστοποιητικό με την εφαρμογή που δήλωσε το δικαίωμα. Αν τα πιστοποιητικά ταιριάζουν, τότε το σύστημα παρέχει αυτόματα το δικαίωμα χωρίς να ενημερώσει ή να ζητήσει τη ρητή έγκριση του χρήστη.
SignatureOrSystem	Παρόμοια με το επίπεδο Signature χωρίς να συμπεριλαμβάνονται τα προγράμματα του συστήματος. Επιτρέπει στα προγράμματα προσαρμοσμένου συστήματος Android να πάρουν ένα δικαίωμα. Το επίπεδο αυτό βοηθάει στην ενσωμάτωση νέον τμημάτων και αναβαθμίσεων που προέρχονται από 3ες ομάδες και δεν χρειάζεται συνήθως από τους προγραμματιστές.
System	Αυτό το επίπεδο δικαιωμάτων μπορεί να χορηγηθεί σε οσεσδήποτε εφαρμογές που είναι εγκατεστημένες στην αρχική έκδοση του συστήματος. Δεν συνιστάται η χρήση αυτής της επιλογής, καθώς το επίπεδο προστασίας **Signature** καλύπτει τις περισσότερες ανάγκες δικαιωμάτων ανεξάρτητα από το πού και το πότε ακριβώς είναι εγκατεστημένες εφαρμογές. Αυτό το επίπεδο δικαιωμάτων χρησιμοποιείται σε ορισμένες ειδικές περιπτώσεις, όπου κάποιοι προμηθευτές έχουν εφαρμογές που στηρίζονται σε μια έκδοση του λειτουργικού και πρέπει να μοιράζονται ρητά ειδικά χαρακτηριστικά επειδή είναι αλληλοεξαρτώμενες.
`Development`	Αυτό το επίπεδο δικαιωμάτων μπορεί (προαιρετικά) να χορηγηθεί κατά την διάρκεια της ανάπτυξης εφαρμογών.

Πίνακας 1, Επίπεδα προστασίας για τα δικαιώματα που ορίζονται στο Android manifest

Εκτός από την ανάγνωση και την εγγραφή δεδομένων, πολλά δικαιώματα επιτρέπουν στις εφαρμογές να εκτελούν Services ή Activities του συστήματος με κρίσιμα από άποψη

ασφάλειας αποτελέσματα. Για παράδειγμα, με το ανάλογο δικαίωμα ένα βιντεοπαιχνίδι μπορεί να πάρει τον πλήρη έλεγχο της οθόνης και να αποκρύπτει τη γραμμή κατάστασης, ενώ ένας dialer μπορεί να εκκινήσει μία ανεπιθύμητη κλίση σε έναν αριθμό με υψηλή χρέωση, χωρίς να το αντιλαμβάνεται ο χρήστης. Επίσης όταν κάποια εφαρμογή αιτείται ένα δικαίωμα που δεν της έχει χορηγηθεί εγείρεται ένα SecurityException.

2.1.2 Rooting

Το Rooting είναι η διαδικασία με την οποία ανακτάται η πλήρης διαχειριστική πρόσβαση σε μία συσκευή. Παρά το γεγονός ότι το Android είναι ένα λειτουργικό σύστημα ανοικτού κώδικα δεν επιτρέπεται στους χρήστες η πλήρης πρόσβαση ως διαχειριστές. Το 2007 όταν ξεκίνησε το iPhone, οι προγραμματιστές συνειδητοποίησαν γρήγορα τις πραγματικές αδυναμίες της συσκευής οι οποίες οδήγησαν στην εκμετάλλευση των αδυναμιών του iPhone από μη εξουσιοδοτημένους χρήστες (Jailbreaking). Αυτή η τεχνική γνωστή και ως privilage escalation ή αλλιώς Rooting υλοποιήθηκε και σε άλλες πλατφόρμες όπως και στο Android.

Σημείωση: Η πρόσβαση ως root είναι όπως η φωτιά: είναι τόσο χρήσιμο όσο και επικίνδυνο. Είναι προφανές ότι ορισμένες διαδικασίες, όπως το σύστημα Zygote[12] χρειάζεται δικαιώματα root για να κάνουν τη δουλειά τους. Είναι επίσης γνωστό ότι για στον εξομοιωτή[13] είναι εύκολο να αποκτήσει κανείς δικαιώματα διαχειριστή (su, ADB Shell, κ.τλ.), στις διανομές Android που πληρούν τους κανονισμούς και τους νόμους δεν επιτρέπεται αυτό. Κάποιες φορές όμως υπό προϋποθέσεις χορηγείται πρόσβαση root, όπως για παράδειγμα, για να δημιουργηθούν κάποια κρυπτογραφημένα και συμπιεσμένα αρχεία εικονικής μνήμης. Τέτοιες περιπτώσεις είναι εύκολα αντιμετωπίσιμες για προσαρμοσμένες εκδόσεις του Android, που λειτουργούν με emulators. Όπως για παράδειγμα στις διανομές Android όπου οι μηχανισμοί ασφάλειας είναι ενεργοποιημένοι. Αντιθέτως εάν η παράμετρος συστήματος (SystemProperty) ro.secure δεν έχει οριστεί σε 1, τότε η πλατφόρμα εφαρμόζει τα ελάχιστα μέτρα ασφάλειας και έχει το επίπεδο ασφάλειας του εξομοιωτή.

2.1.3 Κατάσταση επαναφοράς (DFU mode)

Ο Bootloader [6] υπάρχει στις συσκευές Android για να ξεκινήσει η διαδικασία φόρτωσης του λειτουργικού συστήματος στη μνήμη RAM. Διαφορετικές εκδόσεις του Android OS και διαφορετικά μοντέλα κινητών τηλεφώνων μπορούν να έχουν διαφορετικές εκδόσεις bootloader. Σε ορισμένες περιπτώσεις, οι χρήστες Android έχουν την δυνατότητα να εισέλθουν σε bootloader mode, είτε για να την αναβάθμιση του bootloader είτε για την αναβάθμιση της μνήμης flash ή ακόμα και την αναβάθμιση της έκδοσης του λειτουργικού. Υπάρχουν μερικές μέθοδοι για την είσοδο σε κατάσταση bootloader στις Android συσκευές, ανάλογα αν το τηλέφωνο διαθέτει πληκτρολόγιο QWERTY ενσωματωμένο ή όχι και αν το επιτρέπει ο κατασκευαστής της συσκευής.

Παραδείγματα ενεργοποίησης της λειτουργίας bootloader:

Σε μία κλειστή συσκευή:

12 http://vinnysoft.blogspot.gr/2009/12/zygote-system-process.html

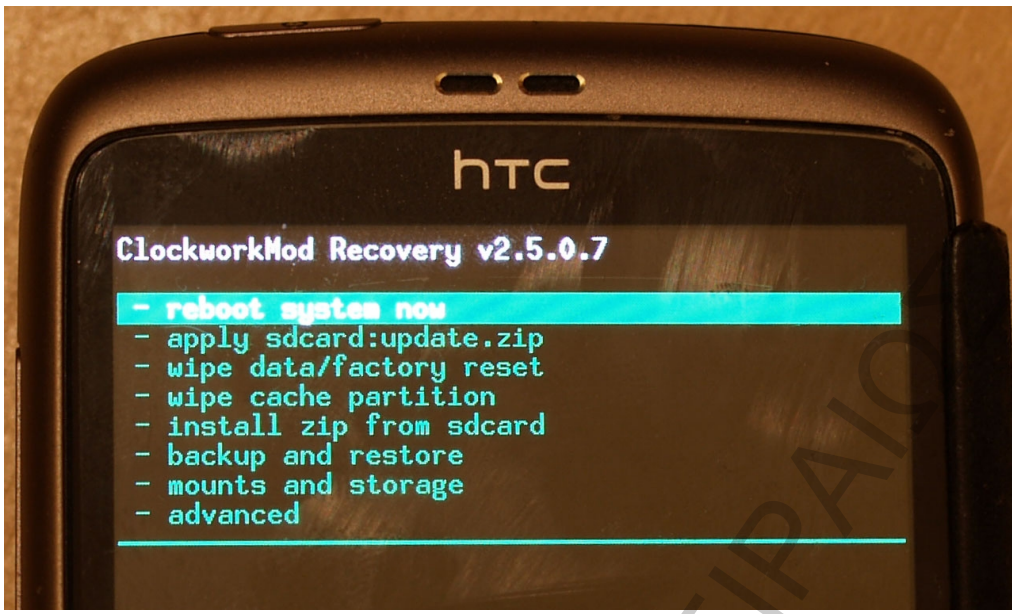
13 Εξομοιωτής νοείται μία εικονική συσκευή (Κεφάλαιο 6)

Μέθοδος 1. Κρατιέται πατημένο το βελάκι προς τα πάνω και πιέζεται το πλήκτρο για να ανοίξει η συσκευή μέχρι να ανάψει το φως της οθόνης.
Μέθοδος 2 Κρατιέται πατημένο το πλήκτρο που χαμηλώνει την ένταση του ήχου και το πλήκτρο της κάμερας και πιέζεται το πλήκτρο για να ανοίξει η συσκευή μέχρι να ανάψει το φως της οθόνης.
Μέθοδος 3 Κρατιέται πατημένο το πλήκτρο που χαμηλώνει την ένταση του ήχου και πιέζεται το πλήκτρο για να ανοίξει η συσκευή μέχρι να ανάψει το φως της οθόνης.

Κάποιες φορές οι χρήστες επιθυμούν να πάνε στο Σύστημα Επαναφοράς Λειτουργίας (System Recovery Mode) της συσκευής Android. Είτε για να καθαρίσουν κάποια cache partition, ή να διαγράψουν εργοστασιακά δεδομένα ή την επαναφορά του τηλεφώνου στην εργοστασιακή του κατάσταση ή να γίνει κάποια αναβάθμιση στο λειτουργικό ή την κάρτα SD. Παράδειγμα ενεργοποίησης της λειτουργίας επαναφοράς συστήματος στη συσκευή Android Motorola Droid / Milestone.

Σε ανοιχτή συσκευή:

Μέθοδος 1: Κρατιέται πατημένο το πλήκτρο X για το πληκτρολόγιο QWERTY του τηλεφώνου και πατήστε το κουμπί λειτουργίας μέχρι να ανάψει η οθόνη. Όταν εμφανιστεί το λογότυπο της Motorola φαίνεται, αφήνεται το κουμπί τροφοδοσίας, αλλά κρατιέται το πλήκτρο X μέχρι το ανάλογο εικονίδιο rooting να εμφανιστεί.
Μέθοδος 2: Κρατιέται πατημένο το πλήκτρο κάμερας στο πλάι του τηλεφώνου και μετά το κουμπί λειτουργίας μέχρι να ανάψει η οθόνη. Όταν εμφανιστεί το λογότυπο της Motorola φαίνεται, αφήνεται το κουμπί τροφοδοσίας, αλλά παραμένει πατημένο το πλήκτρο κάμερας μέχρι το ανάλογο εικονίδιο rooting να εμφανιστεί.



Εικόνα 12. Σύστημα Επαναφοράς Λειτουργίας

2.2 Δομικά στοιχεία του Android

Ακολουθεί μια παρουσίαση των δομικών στοιχείων του Android από την σκοπιά της ασφάλειας [12].

2.2.1 Intents

Τα *Intents* [14] είναι ένας ειδικός μηχανισμός στα Android για τη μεταφορά δεδομένων μεταξύ Android (λειτουργικού) και διαδικασιών και αποτελούν τον πυρήνα ενός μεγάλου μέρους των IPC μηχανισμών του Android. Τα Intents δεν επιβάλλουν την πολιτική ασφάλειας, αλλά είναι συνήθως ο αγγελιοφόρος που διασχίζει τα πραγματικά όρια ασφαλείας του συστήματος. Για να επιτραπεί ο επικοινωνιακός τους ρόλος τα Intents μπορούν να αποσταλούν μέσω διεπαφών Binder[14]. Σχεδόν όλα τα Android IPC είναι υλοποιημένα μέσω Binder, αν και τις περισσότερες φορές αυτό είναι κρυμμένο από τους χρήστες στα πλαίσια απόκρυψης της πολυπλοκότητας του συστήματος.

2.2.1.1 Επισκόπηση Intent

Τα Intents χρησιμοποιούνται για:

• Για την εκκίνηση ενός *Activity* και το συντονισμό με άλλα προγράμματα, όπως περιήγηση σε μια ιστοσελίδα με την χρήση της μεθόδου startActivity().
• Για την μετάδοση ενημερώσεων ή αλλαγών ή γεγονότων προς τις ανάλογες εφαρμογές με την χρήση των *Context* μεθόδων sendBroadcast(), sendStickyBroadcast() και sendOrderedBroadcast().

• Ως ένας τρόπος για να την εκκίνηση ή τον τερματισμό επικοινωνίας με υπηρεσίες που εκτελούνται στο παρασκήνιο με την χρήση των *Context* μεθόδων startService(), stopService() και bindService().

14 Κεφάλαιο 2.2.6 Binder Interfaces

• Για την χορήγηση πρόσβασης σε δεδομένα όπως οι επαφές του χρήστη μέσω των ContentProviders μεθόδων: Context’s getContentResolver() ή Activities managedQuery().

• Ή ως κλήσεις για τον χειρισμό γεγονότων, όπως η επιστροφή των αποτελεσμάτων ή λάθη συγχρονισμού (πχ με τα PendingIntents) που παρέχονται από τους clients μέσω διασυνδέσεων σε servers.

Τα *Intents* έχουν πάρα πολλές λεπτομέρειες υλοποίησης, η βασική ιδέα όμως είναι ότι αντιπροσωπεύουν μια άμορφη μάζα από συνέχειες δεδομένων που μπορούν να μεταφερθούν μεταξύ των προγραμμάτων για έναν σκοπό. Έχουν συνήθως μια ενέργεια (action), η οποία είναι μια συμβολοσειρά όπως πχ (android.intent.action.VIEW) που προσδιορίζει κάποιο συγκεκριμένο στόχο ή κάποια δεδομένα στοιχεία στην μορφή του Uri (υλοποίηση του android.net.Uri class).
Γενικά τα APIs που δέχονται Intents ως είσοδο μπορούν να περιοριστούν με δικαιώματα manifest. Αυτό επιτρέπει στους προγραμματιστές να δημιουργούν Activities, BroadcastReceivers, ContentProviders ή Services που είναι προσβάσιμα μόνο από την προκαθορισμένη εφαρμογή που ο προγραμματιστής έχει παραχωρήσει αυτό το δικαίωμα.

2.2.1.2 Φίλτρα Intent

Ανάλογα με τον τρόπο με τον οποίο αποστέλλονται τα Intents μπορούν να αποστέλλονται και από τον Android Activity Manager. Για παράδειγμα, ένα Intent μπορεί να χρησιμοποιηθεί για να εκκινήσει ένα Activity καλώντας την μέθοδο Context.startActivity (Intents intent). Το Activity εντοπίζεται από τον Android Activity Manager ταιριάζοντας το δοθέν Intent σε αντιπαράθεση με τα IntentFilters που έχουν καταχωρηθεί για όλα τα Activity στο λειτουργικό σύστημα προσπαθώντας να εντοπίσει τον καλύτερο δυνατό συνδυασμό. Παρ' όλα αυτά ένα Intent μπορεί να παρακάμψει τον συνδυασμό του IntentFilter που χρησιμοποιεί ο Activity Manager. Κάθε εξαγόμενο Activity μπορεί να εκκινήσει με οποιεσδήποτε Intent τιμές, είτε κάποια δράση (Action) ή κάποια δεδομένα ή extras, κλπ. Τα Intent recipients όπως και τα Activities, τα Services και τα BroadcastReceivers χρειάζονται για να τον χειρισμό των δυνητικά κακόβουλων κλήσεων στο σύστημα, σε αντίθεση με ένα IntentFilter που δεν φιλτράρει ένα κακόβουλο Intent. Τα IntentFilters βοηθούν το σύστημα να επιλέξει τον σωστό χειριστή για ένα συγκεκριμένο Intent, χωρίς όμως να αποτελούν ένα σύστημα φιλτραρίσματος εισόδου. Επειδή τα IntentFilters δεν είναι όρια ασφαλείας δεν μπορούν να συνδεθούν με δικαιώματα. Όπως φαίνεται και από τις ακόλουθες ενότητες, κανένας IPC μηχανισμός που χρησιμοποιεί IntentFilters δεν μπορεί να στηριχθεί πάνω τους για αξιολόγηση της ασφάλειας των εισροών.

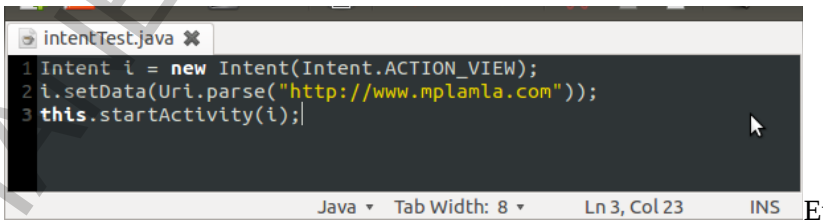
2.2.1.3 Ανάκλαση Intent

Ένα κοινό ιδίωμα κατά την επικοινωνία με το Android είναι η λήψη ενός callback μέσω *Intent*.
Ένα χαρακτηριστικό παράδειγμα αυτού του ιδιώματος σε χρήση αποτελεί ο Location Manager, ο οποίος είναι μια προαιρετική υπηρεσία. Ο Location Manager είναι ένα binder interface με τη μέθοδο LocationManager.addProximityAlert(). Αυτή η μέθοδος παίρνει ένα Pending Intent, το οποίο παρέχει τη δυνατότητα στους καλούντες της υπηρεσίας να προσδιορίσουν τον τρόπο ενημέρωσης τους. Τέτοια callbacks μπορούν να χρησιμοποιηθούν οποιαδήποτε στιγμή, αλλά συμβαίνουν συχνά ειδικά όταν εμπλέκονται σε IPC μέσω Activity, Service, BroadcastReceiver ή Binder interface χρησιμοποιώντας Intents.
Εάν μία εφαρμογή πρόκειται να στείλει ένα Intent όταν καλείται, θα πρέπει να μην αποφεύγεται η αποστολή Intent callback όσο αυτό είναι δυνατό. Και στην περίπτωση που καλείται κάποιο ανεπιθύμητο Intent callback, προλαμβάνεται η δράση του με την χρήση της *android.ap-*

p.PendingIntent class (Android SDK 0.9) Αν μια εφαρμογή εκθέτει μια διεπαφή/ interface που επιτρέπει στον καλώντα του interface να ενημερώνεται με τη λήψη ενός *Intent*, είναι προτιμότερο να οριστεί ένα PendingIntent αντί για *Intent*. Τα PendingIntents αποστέλλονται ως κομμάτι της διαδικασίας που τα δημιούργησε. Ο server επιβεβαιώνει ότι το callback θα να αντιμετωπιστεί ως προερχόμενο από τον καλούντα και θα λάβει τα αντίστοιχα δικαιώματα. Αυτό μετατοπίζει τον κίνδυνο από την υπηρεσία στον καλούντα. Την ίδια στιγμή ο καλών πρέπει να εμπιστεύεται την υπηρεσία και να της δίνει την δυνατότητα να στέλνει *Intent* όπως έχει και αυτός απέναντι της, απλή διαδικασία, δεδομένου ότι ο καλών ελέγχει τις ιδιότητες της *Intent* του. Η τεκμηρίωση του endingIntent συνιστά σύνεση, κλείδωμα του PendingIntent με το συγκεκριμένο component που έχει σχεδιαστεί για αποστέλλεται η ανάκληση με την μέθοδο setComponent(). Αυτό ελέγχει την αποστολή *Intent*.

2.2.2 Activities

Τα Activities [15] επιτρέπουν στις εφαρμογές να καλούν η μία την άλλη επανα- χρησιμοποιώντας έτσι η μία χαρακτηριστικά της άλλης επιτρέποντας έτσι την αντικατάσταση ή την βελτίωση των επιμέρους τμημάτων των εφαρμογών του συστήματος ανάλογα με τις επιθυμίες του χρήστη. Συχνά, τα Activities εκτελούνται ως αυτόνομες διαδικασίες, που τρέχουν με το δικό τους UID και έτσι δεν έχουν πρόσβαση στα δεδομένα της εφαρμογής που τα καλεί, με εξαίρεση τα δεδομένα που παρέχονται από το Intent (που χρησιμοποιείται για να καλέσει το εκάστοτε Activitiy). Ο ευκολότερος τρόπος για να είναι ασφαλή τα Activities είναι απλά να επιβεβαιώνονται οποιεσδήποτε αλλαγές ή ενέργειες ρητά από τον χρήστη. Στην περίπτωση που ξεκινάει ένα Activity από ένα Intent χωρίς να υπάρχει το ακριβές δικαίωμα εκκίνησης του συγκεκριμένου Activitiy μπορεί να προκληθεί κάποιο σφάλμα ή κάποια σύγχυση. Τα Activities δεν μπορούν να βασιστούν σε IntentFilters (<intent-filter> στο AndroidManifest.xml) ώστε να σταματήσουν την κλήση από επιβλαβή ρυθμισμένα Intents. Η παρανόηση αυτή είναι στην πραγματικότητα παρόμοια με τα κοινά σφάλματα στον κώδικα (λογικά σφάλματα). Τα Activities βασίζονται σε ελέγχους δικαιωμάτων ως μηχανισμό ασφαλείας. Η ρύθμιση του <android: *permission* attribute> (ένα τυχαίο δικαίωμα) σε μια δήλωση ενός Activity εμποδίζει στα προγράμματα που δεν έχουν το δικαίωμα αυτό από την παρακινδυνευμένη απ 'ευθείας έναρξη του Activity. Ο ακόλουθος κώδικας δείχνει την έναρξη ενός Activity με Intent. Στο συγκεκριμένο παράδειγμα, ο Activity Manager κατά πάσα πιθανότητα θα αποφασίσει να ξεκινήσει τον web browser για να το χειριστεί, επειδή το πρόγραμμα περιήγησης στο Web έχει καταχωρηθεί Activities με ένα ανάλογο intent-filter.



Ει

κόνα 13, Εκκίνηση ενός Activity από το IntentFilter του

2.2.3 Broadcasts

Τα μηνύματα Broadcasts [16] είναι ένα μέσο να επικοινωνούν οι εφαρμογές με τα στοιχεία του συστήματος αποτελεσματικά και με ασφάλεια. Τα μηνύματα αυτά αποστέλλονται ως Intents και το σύστημα διαχειρίζεται την αποστολή τους με βάση τους Intent receivers και την ανάλογη πολιτική δικαιωμάτων.

2.2.3.1 Μετάδοση Intents

Τα Intent *μηνύματα* μπορούν να μεταδοθούν (ως broadcast) προς τους BroadcastReceivers, επιτρέποντας έτσι την ανταλλαγή μηνυμάτων μεταξύ των εφαρμογών. Καταχωρώντας ένα BroadcastReceiver στο AndroidManifest.xml μιας εφαρμογής δηλώνεται το receiver class της εφαρμογής να ξεκινάει κατά την εκκίνηση και και να καλείται κάθε φορά που υπάρχει ένα εισερχόμενο broadcast. Ο Activity Manager χρησιμοποιεί τα IntentFilters της εφαρμογής για την καταχώρηση και την αναγνώριση της εφαρμογής που πρόκειται να χρησιμοποιήσει το συγκεκριμένο broadcast.
Όπως αναφέρθηκε και στο κεφάλαιο 2.2.1.2 τα IntentFilters δεν είναι ένας μηχανισμός ασφαλείας γι αυτό και δεν μπορεί να βασιστεί πάνω τους η ασφάλεια των Intent recipients. Ωστόσο, μπορούν να συντελέσουν στην ασφαλή μετάδοση των Intent όταν αυτά είναι ταξινομημένα σε κάποια κατηγορία, ο recipient μπορεί να ρυθμιστεί ώστε να δέχεται μόνο συγκεκριμένες κατηγορίες και να απορρίπτει οτιδήποτε δεν ανήκει στις προκαθορισμένες κατηγορίες. Όπως ένα *Activity*, ένας broadcast sender μπορεί να στείλει σε έναν receiver, ένα *Intent* που όμως δεν θα περάσει από το IntentFilter από την στιγμή που έχει καθορισμένο τον receiver προορισμό του. Οι receivers πρέπει να είναι προετοιμασμένοι για την λύψη απροσδόκητων Intent ή κατακερματισμένων δεδομένων. Όπως πάντα σε ασφαλή προγραμματισμό IPC, ένα προγράμμα πρέπει να επικυρώνει πάντα και προσεκτικά την είσοδο του. Οι BroadcastRecievers καταχωρούνται στο AndroidManifest.xml με την ετικέτα <receiver>. Δεν εξάγονται εξ αρχής, αλλά μπορούν να εξαχθούν εύκολα με την προσθήκη της ετικέτας <intent-filter> (έστω και κενή ετικέτα) ή θέτοντας την τιμή android*:exported="true"*. Μόλις εξαχθούν, οι receivers μπορούν να κληθούν από άλλα προγράμματα. Όπως τα *Activities*, τα *Intents* στα οποία οι BroadcastReceivers δέχονται μπορεί να μην ταιριάζουν με αντίστοιχο IntentFilter. Για τον περιορισμό των διαφόρων εφαρμογών- αποστολέων Intent σε μία εφαρμογή ορίζεται ένα manifest δικαίωμα στο *android:permission* attribute στην ετικέτα του receiver. Όταν ένα δικαίωμα καθορίζεται για έναν receiver, ο Activity Manager επικυρώνει ότι ο αποστολέας έχει το συγκεκριμένο δικαίωμα πριν από την παράδοση του *Intent*(manifest *permission)*. Τα δικαιώματα είναι ο μόνος σωστός τρόπος για να διασφαλιστούν οι receivers δέχονται μόνο *Intents* από τους κατάλληλους αποστολείς- senders, όμως τα δικαιώματα δεν επηρεάζουν τις ιδιότητες των εισερχόμενων *Intents*.

2.2.3.2 Ασφαλής αποστολή Broadcast Intent

Στην αποστολή ενός broadcast, οι προγραμματιστές περιλαμβάνουν κάποιες πληροφορίες, μερικές φορές ακόμα και ευαίσθητα δεδομένα, όπως ένα Binder. Εάν τα δεδομένα που αποστέλλονται είναι ευαίσθητα, πρέπει να υφίσταται ιδιαίτερη προσοχή όσο αφορά τον προορισμό του ποιος είναι και εάν είναι σίγουρα ο προβλεπόμενος. Ο πιο απλός τρόπος για να προστατεύσει αυτή η δυναμική διαδικασία αποστολής είναι η απαίτηση από τον αποδέκτη να έχει ένα συγκεκριμένο δικαίωμα. Περνώντας το όνομα του manifest δικαιώματος (receiverPermission είναι το όνομα της παραμέτρου) σε μία από τις μεθόδους Context's broadcastIntent() απαιτεί από τους recipients να έχουν αυτό το δικαίωμα για να μπορέσουν να δεχτούν το *Intent*. Έτσι οι προγραμματιστές ελέγχουν ποιες εφαρμογές μπορούν να λάβουν το *Intent.* Τα Broadcasts προτιμούνται μεταξύ των μηχανισμών IPC, όταν πρόκειται για αποστολή ευαίσθητων δεδομένων λόγω της ευκολίας της ρύθμισης των απαιτούμενων δικαιωμάτων προς τους Receivers.

2.2.4 Services

Τα Services [17] είναι διεργασίες που τρέχουν στο παρασκήνιο και παρέχονται από το Android για να επιτρέψει παρασκηνιακές εργασίες να εκτελούνται παράλληλα με αυτές που εκτελούνται στο προσκήνιο, όπως μουσική ή ένας server. Μπορούν να εκκινηθούν από ένα *Intent* και προαιρετικά μπορούν να επικοινωνήσουν με το προσκήνιο με ένα interface Binder καλώντας την μέθοδο του bindService(). Τα Services είναι παρόμοια με τους BroadcastReceivers και τα *Activities* υπό την έννοια ότι μπορούν να ξεκινήσουν ανεξάρτητα από τα IntentFilters τους, καθορίζοντας ένα Component (στην περίπτωση που εξάγονται). Τα Services μπορούν επίσης να διασφαλίζονται με την προσθήκη ελέγχου του αντίστοιχου δικαιώματος <service> στο AndroidManifest.xml. Οι μεγάλης διάρκειας συνδέσεις που παρέχονται από την bindService() δημιουργούν ένα γρήγορο δίαυλο IPC μεταξύ των Services και της εφαρμογής και βασίζεται στη διεπαφή Binder (κεφάλαιο 2.2.6 Binder Interfaces). Τα Binder interfaces μπορούν να ελέγχουν τα δικαιώματα του στοιχείου που τα καλεί, επιτρέποντάς τους να επιβάλουν περισσότερα από ένα δικαιώματα κάθε φορά ή διαφορετικά δικαιώματα σε διαφορετικές αιτήσεις. Τα Services παρέχουν ως εκ τούτου πολλούς τρόπους για την διασφάλιση ότι το καλών στοιχείο είναι αξιόπιστο παρόμοια με *Activities*, BroadcastReceivers και interfaces Binders. Η κλήση ενός Service είναι λίγο πιο περίπλοκη. Αυτό δεν έχει σημασία για ασήμαντες διεργασίες όπως ο χρονοπρογραμματισμός μίας λίστας MP3, αλλά σε περιπτώσεις κλήσεων Service με ευαίσθητα δεδομένα ή όπως η αποθήκευση των κωδικών πρόσβασης ή προσωπικών μηνυμάτων, θα πρέπει να επιβεβαιωθεί ότι το Service που γίνεται η σύνδεση είναι το σωστό και όχι κάποιο κακόβουλο που δεν πρέπει να έχει πρόσβαση στις πληροφορίες που παρέχονται. Εάν το επιθυμητό για σύνδεση στοιχείο είναι γνωστό εκ των προτέρων, τότε μπορεί να καθοριστεί ρητά στο αντίστοιχο Intent που χρησιμοποιείται για την σύνδεση αυτή. Εναλλακτικά, μπορεί να το επαληθευτεί από το όνομα που παρέχεται στη μέθοδο onServiceConnected(ComponentName όνομα, IBinder Service) της υλοποίησης της SeviceConnection. Όμως αυτό δεν είναι δυναμικό και δεν επιτρέπει στους χρήστες να αντικαταστήσουν το παρεχόμενο Service. Για να επιτρέπεται δυναμικά στους χρήστες να αντικαθιστούν Services και στη συνέχεια να τα εγκρίνουν μέσω του ελέγχου των δικαιωμάτων, τα οποία έχουν δηλωθεί εκ των προτέρων και έχουν λάβει εξουσιοδότηση από τον χρήστη. Το όνομα του package είναι ένας τρόπος για να επικυρωθεί ένα δικαίωμα. Το όνομα του υλοποιημένου στοιχείου λαμβάνεται με το αποτέλεσμα της κλήσης της μεθόδου onServiceConnected() και αυτό το όνομα συνδέεται με τα δικαιώματα της εφαρμογής.

Παράδειγμα, έλεγχος εάν ένα πακέτο έχει ένα δικαίωμα:

```
int res = getPackageManager().checkPermission(permToCheck,
name.getPackageName());
```

Το αποτέλεσμα της checkPermission() είναι είτε PackageManager.PERMISSION_GRANTED είτε PackageManager.PERMISSION_DENIED με integer τιμές 1 και 0 αντίστοιχα.

2.2.5 ContentProviders

Το Android έχει το μηχανισμό ContentProvider [18] για να επιτρέπει στις εφαρμογές να μοιράζονται τα ανεπεξέργαστα δεδομένα. Αυτό εφαρμόζεται για τον διαμοιρασμό SQL δεδομένων,για εικόνες, για ήχους κ.α. Αυτή η διασύνδεση έχει σχεδιαστεί προφανώς για να χρησιμοποιείται σε SQL backend και αυτός είναι ο κύριος σκοπός του. Οι ContentProviders υλοποιούνται από τις εφαρμογές για να κάνουν προσβάσιμα τα δεδομένα τους στο υπόλοιπο συστήματος, η ετικέτα <provider> στο AndroidManifest.xml της εφαρμογής κατοχυρώνει έναν ContentProvider για διάθεση και καθορίζει τα δικαιώματα πρόσβασης. Η τεκμηρίωση ασφάλειας του Android αναφέρει ότι μπορούν να υπάρξουν ξεχωριστά δικαιώματα ανάγνωσης και εγγραφής σε έναν ContentProvider. Κρατώντας όμως μόνο το δικαίωμα εγγραφής δεν σημαίνει ότι μπορεί να είναι λειτουργική και η ανάγνωση σε έναν ContentProvider, λόγω του ότι δεν είναι δυνατόν να έχουμε μόνο write (εγγραφή) . Για παράδειγμα, σε SQL ερωτήματα, τα

αποτελέσματα κλήσης της updateQuery() ή της deleteQuery() παράγουν ένα SQL statement στο οποίο υπάρχει η ρήτρα *"where"* που καθορίζεται από τον καλούντα. (Αυτό ισχύει ακόμα και αν ο καλών έχει μόνο το δικαίωμα εγγραφής.)
Ελέγχοντας μια ρήτρα *"where"* που δεν επιστρέφει άμεσα δεδομένα, αλλά η ικανότητα να αλλάξει τη συμπεριφορά του SQL statement που βασίζεται στην τιμή των δεδομένων που επιστρέφονται. Παρακολουθώντας αυτές τις παρενέργειες μιας σειράς από έξυπνες κλήσεις με *"where"* ρήτρες, οι κακόβουλοι καλούντες μπορούν να ανακατασκευάσουν σιγά-σιγά όλα τα δεδομένα που περιέχει μία βάση. Η δημιουργία ενός ασφαλούς ContentProvider βασισμένο σε μόνο ένα τύπο δικαιωμάτων, σαφώς και είναι λειτουργική σε περιπτώσεις για απλές εφαρμογές, που χειρίζονται μόνο αρχεία ή μνήμη, αλλά αυτό δεν είναι δυνατό σε σύνθετα SQL ερωτήματα. Η δήλωση των επιθυμητών δικαιωμάτων ανάγνωσης και εγγραφής επιβάλλονται από το σύστημα απευθείας μέσω της ετικέτας <provider> στο AndroidMainfext.xml. Οι ετικέτες είναι android: readPermission και android: writePermission. Αυτά τα δικαιώματα ισχύουν κατά τον χρόνο πρόσβασης με την επιφύλαξη των πιθανών περιορισμών που μπορεί να έχει μία εφαρμογή. Διότι μπορεί να υπάρχει μία ετικέτα που να καθορίζει ένα γενικό δικαίωμα που θα απαιτείται για κάθε πρόσβαση.

2.2.6 Binder Interfaces

Το Binder [19] είναι ένα πρόγραμμα οδήγησης συσκευής πυρήνα που χρησιμοποιεί κοινόχρηστη μνήμη στο Linux, χαρακτηριστικό απαραίτητο για να επιτευχθεί αποτελεσματικό και ασφαλή IPC. Οι υπηρεσίες του συστήματος που δημοσιεύονται ως Binder Interfaces και η **AIDL** (Android Interface Definition Language) χρησιμοποιούνται εκτός από τον καθορισμό των Interfaces του συστήματος και για να επιτρέψουν στους προγραμματιστές να δημιουργήσουν τα δικά τους client- Binders και τους servers. Η ορολογία μπορεί να προκαλέσει σύγχυση, αλλά γενικά οι servers κληρονομούν την android.os.Binder και υλοποιούν την μέθοδο onTransact() ενώ οι clients λαμβάνουν ένα binder interface ως αναφορά του android.os.IBinder και να καλούν την μέθοδο transact(). Τόσο η transact() αλλά και η onTransact() χρησιμοποιούν στιγμιότυπα του android.os.Parcel για να ανταλλάσσουν δεδομένα αποτελεσματικά. Το Android υποστηρίζει τα Binder περιλαμβάνοντας το Parcelable interface. Τα Parcelable αντικείμενα μπορούν να μετακινηθούν μεταξύ των διεργασιών μέσω ενός Binder. Στην ουσία μια αναφορά σε ένα Binder είναι ένας περιγραφέας που διατηρείται από τη διάταξη Binder (η οποία είναι ένας driver συσκευής του πυρήνα). Το Binder IPC μπορεί να χρησιμοποιηθεί για να πάρει και να επιστρέψει πρωτογενής τύπους Parcelable αντικείμενα, περιγραφείς αρχείων (descriptors, που επιτρέπουν memory maps) και *Binders*. Έχοντας μια αναφορά σε ένα Binder interface επιτρέπονται κλήσεις προς τη διεπαφή του (δηλαδή κλήσεις προς transact() αντιστοιχίζεται με μία κλήση onTransact () στην πλευρά του server) - αλλά δεν εγγυάται ότι η υπηρεσία που εκθέτει αυτό το interface θα κάνει οτιδήποτε αιτείται ο client. Για παράδειγμα, κάθε πρόγραμμα έχει την δυνατότητα να πάρει μια αναφορά στον Binder της *Zygote υπηρεσίας* συστήματος και έτσι να καλέσει τη μέθοδο για να εκκινήσει μια εφαρμογή ως κάποιος άλλος χρήστης, αλλά το *Zygote* αγνοεί τα αιτήματα από μη εξουσιοδοτημένες διαδικασίες. Η ασφάλεια του Binder έχει δύο βασικούς τρόπους με τους οποίους μπορεί να επιβάλει την ασφάλεια ελέγχοντας :

- Την ταυτότητα του καλούντος
- Την ασφάλεια της αναφοράς του Binder

2.3 Δικαιώματα αρχείων

Τα UNIX-style δικαιώματα αρχείων υπάρχουν και στο Android [20] για τα συστήματα αρχείων που έχουν διαμορφωθεί για τη υποστήριξή του λειτουργικού, όπως και στο root σύστημα αρχείων. Κάθε εφαρμογή έχει τη δική της περιοχή στο σύστημα αρχείων που της ανήκει όπως τα προγράμματα έχουν έναν κατάλογο στο Home για να πάει μαζί με τις ταυτότητες των χρηστών τους. Ένα Activity ή ένα Service μπορεί να δώσει πρόσβαση σε αυτόν τον κατάλογο με τις μεθόδους getFilesDir(), getDir(), openFileOutput(), openFileInput(), getFileStreamPath(), αλλά τα αρχεία και τα μονοπάτια που επιστρέφονται δεν είναι κάτι διαφορετικό από ότι συνήθως και έτσι μπορούν να χρησιμοποιηθούν με άλλα αντικείμενα διαχείρισης αρχείων, όπως πχ. η FileInputStream. Η παράμετρος mode χρησιμοποιείται για να δημιουργήσει ένα αρχείο με ένα συγκεκριμένο σύνολο δικαιωμάτων (που αντιστοιχούν στα δικαιώματα αρχείων UNIX). Τα οποία μπορούν να συνδυαστούν με την συνθήκη OR. Για παράδειγμα, ένας τρόπος MODE_WORLD_WRITABLE | MODE_WORLD_READABLE κάνει ένα αρχείο αναγνώσιμο και εγγράψιμο από όλους. Η τιμή MODE_PRIVATE δεν μπορεί να συνδυαστεί με αυτόν τον τρόπο, δεδομένου ότι είναι σαν τιμή είναι μηδέν. Παραδόξως η παράμετρος mode υποδεικνύει εάν το προκύπτον αρχείο δικαιωμάτων έχει περικοπεί ή να αλλαχθεί για πρόσθεση δικαιωμάτων-με την παράμετρο MODE_APPEND.
Δημιουργία ενός αρχείου που είναι εγγράψιμο από όλους:

```
fos = openFileOutput("PublicKey", Context.MODE_WORLD_READABLE);
```

Η προκύπτουσα FileOutputStream (που ονομάζεται *fos* παραπάνω) μπορεί να γραφτεί μόνο από αυτή τη διαδικασία, αλλά να μπορεί να διαβαστεί από οποιοδήποτε πρόγραμμα στο σύστημα.
Ο μηχανισμός που περνάει τα διακαιώματα world-readable ή world-writable των αρχείων σαν flags είναι απλούστερος από από τον αντίστοιχο που υποστηρίζεται τα δικαιώματα αρχείων Linux, αλλά παράλληλα και επαρκής για τις περισσότερες εφαρμογές Android. Γενικά, για κάθε κώδικα που δημιουργεί δεδομένα που είναι προσβάσιμα από παντού πρέπει προσεκτικά να εξετάζεται:

- Εάν υπάρχουν στο αρχείο ευαίσθητα δεδομένα
- Εάν κάποια αλλαγή στα δεδομένα μπορεί να προκαλέσει κάτι απροσδόκητο ή ανεπιθύμητο.
- Σε περίπτωση που τα δεδομένα είναι σε μία σύνθετη μορφή parser (μετασχηματιστής-αναγνώστης) μπορεί να έχει εκμεταλλεύσιμα τρωτά σημεία[15].
- Στην περίπτωση που το εξαγόμενο αρχείο είναι εγγράψιμο από όλους (world-writeable), υπάρχει η πιθανότητα να τροποποιηθεί κακόβουλα από μία 3η εφαρμογή και να χρεωθεί την ευθύνη η αρχική εφαρμογή[16].

2.4 SQL injection

Για την αποφυγή αιτημάτων SQL injection πρέπει να οριοθετηθούν με σαφήνεια οι SQL δηλώσεις όπως και τα στοιχεία που περιλαμβάνονται σε αυτές. Εάν τα δεδομένα έχουν παρερμηνευθεί ώστε να είναι μέρος της πρότασης SQL, το αποτέλεσμα SQL injection μπορεί να προκαλέσει δυσάρεστες συνέπειες, από αβλαβή μικρά σφάλματα που ενοχλούν απλά τους χρήστες έως δημιουργία σοβαρών κενών ασφαλείας που εκθέτουν δεδομένα του χρήστη. Το

15 Ιστορικά, σύνθετες δομές αρχείων που έχουν γραφτεί με C ή C++ έχουν πολλές εκμεταλλεύσιμες ευπάθειες.

16 Αυτό συμβαίνει διότι το αρχείο αυτό αποθηκεύεται στο home φάκελο της εφαρμογής.

SQL injection εύκολα αποφεύγεται σε σύγχρονες πλατφόρμες όπως το Android, χρησιμοποιώντας παραμετροποιημένα queries που διακρίνουν τα δεδομένα από το ερώτημα. Όλες οι μέθοδοι: query(), update(), delete() των *ContentProivder's* και η managedQuery() των *Activities'* υποστηρίζουν παραμετροποίηση, παίρνουν σαν όρισμα ένα πίνακα συμβλοσειρών (String[]) με παραμέτρους που θα αντικαταστήσουν τους χαρακτήρες '?' που βρίσκονται μέσα στο δοθέν query, με την σειρά που αυτοί εμφανίζονται. Αυτό παρέχει σαφή διαχωρισμό μεταξύ SQL statement και των δεδομένων που περιλαμβάνονται. Ακόμα και στην περίπτωση που τα δεδομένα περιλαμβάνουν χαρακτήρες που μπορούν να αλλοιώσουν το νόημα του statement, η βάση δεν θα παρερμηνεύσει την είσοδο αυτή. Σφάλματα SQL injection μπορούν να συμβούν κατά την είσοδο δεδομένων από τον χρήστη όταν εισάγονται χαρακτήρες όπως (') και (''). Ένα SQL injection μπορεί να συμβεί οποτεδήποτε λαμβάνονται δεδομένα και μετά χρησιμοποιούνται σε ένα SQL query, αυτό σημαίνει ότι δεδομένα από *Binder* interfaces, ή *Intents* που έχουν ληφθεί από ένα broadcast, ή από κάποια κλήση *Service* ή *Activity* και φυσικά αυτά αποτελούν στόχο προς εκμετάλλευση από ένα κακόβουλο λογισμικό. Επίσης σημαντικός είναι και ο έλεγχος για SQL injection όταν τα δεδομένα έρχονται από απομακρυσμένες πηγές (όπως RSS feeds, ιστοσελίδες, κ.τ.λ). Με την χρήση παραμετροποιημένων τύπων για όλες τις τιμές που αναφέρονται και την αποφυγή ενώσεων των συμβολοσειρών κατά τη δημιουργία SQL statements, οι επιθέσεις SQL injection, μπορούν να αποφευχθούν τελείως.

2.5 Μαζική αποθήκευση Mass Storage

Οι Android συσκευές είναι πιθανό να έχουν περιορισμένη μνήμη στο εσωτερικό του συστήματος αρχείων. Ορισμένες συσκευές μπορούν να υποστηρίξουν προσθήκη μνήμης για τα συστήματα αρχείων η οποία, ωστόσο προσαρτάται στις κάρτες μνήμης. Η αποθήκευση των δεδομένων σε αυτά τα συστήματα αρχείων φαντάζει λίγο περιπλοκή. Για να είναι εύκολο για τους χρήστες να μετακινούν τα δεδομένα από και προς σε φωτογραφικές μηχανές, υπολογιστές και Android, το σύστημα αρχείων των SD καρτών είναι VFAT. Το οποίο είναι ένα παλιό πρότυπο που δεν υποστηρίζει τους ελέγχους πρόσβασης του Linux, έτσι τα δεδομένα που αποθηκεύονται στις sd- cards είναι απροστάτευτα. Οι χρήστες θα πρέπει να ενημερώνονται ότι η μαζική αποθήκευση γίνεται από κοινού με όλα τα προγράμματα της συσκευής τους, ώστε να τους αποτρέψει να αποθηκεύουν ευαίσθητα δεδομένα εκεί. Στην περίπτωση που χρειαστεί να αποθηκευθούν εμπιστευτικά δεδομένα στην sd-card μπορούν να κρυπτογραφηθούν και να αποθηκεύσουν το κλειδί στην αποθηκευτική περιοχή της εφαρμογής, ενώ το μεγάλο ciphertext (κρυπτογράφημα) στην κάρτα μνήμης. Εφ 'όσον ο χρήστης δεν επιθυμεί να χρησιμοποιήσει την κάρτα αποθήκευσης για να μετακινεί τα δεδομένα σε ένα άλλο σύστημα η λύση της κρυπτογραφίας είναι λειτουργική. Επιπροσθέτως χρειάζεται κάποιος μηχανισμός για να αποκρυπτογραφεί τα δεδομένα και να χειρίζεται το κλειδί για το χρήστη όπως επίσης και την μεταφορά εμπιστευτικών δεδομένων μεταξύ συστημάτων.

2.6 Ασφάλεια με άδεια καλούντος - Έλεγχος ταυτότητας

Όταν μια διεπαφή Binder καλείται, η ταυτότητα του καλούντος παρέχεται ασφαλώς από τον πυρήνα. Το Android συσχετίζει την ταυτότητα της καλούντος εφαρμογής με το νήμα στο οποίο υποβάλλεται η αίτηση[17]. Αυτό επιτρέπει στον παραλήπτη να χρησιμοποιήσει τις μεθόδους του

17 Αναφέρεται στο UID της εφαρμογής και στο PID της διαδικασίας.

Context, checkCallingPermission(String permission) ή checkCallingPermissionOrSelf (String permission) για την επικύρωση των δικαιωμάτων του καλούντος. Κάποιες φορές οι εφαρμογές θέλουν να επιβάλουν τα δικαιώματα που δεν έχουν στους καλούντες και αυτό γίνεται με την checkCallingPermissionOrSelf (permission String) που επιτρέπει στην εφαρμογή να αυτο-καλείται ακόμα, ακόμη και αν δεν διαθέτει το δικαίωμα που απαιτείται κανονικά. Οι Binder υπηρεσίες έχουν την δυνατότητα να κάνουν άλλες Binder κλήσεις, αλλά αυτό επιτρέπεται μόνο για Services με ίδια ταυτότητα (UID και PID). Οι Binder υπηρεσίες έχουν επίσης πρόσβαση στην ταυτότητα των καλούντων με τις στατικές μεθόδους getCallingUid() και getCallingPid() της κλάσης Binder. Αυτές οι μέθοδοι επιστρέφουν το UID και το αναγνωριστικό διαδικασίας (PID) της διαδικασίας που έκανε το Binder call. Οι πληροφορία ταυτότητας κοινοποιείται με ασφάλεια στον εκτελεστή ενός Binder interface από το kernel.

Συμπεράσματα 2ου κεφαλαίου

Κάθε Android εφαρμογή έχει τη δική της ταυτότητα όπως επιβάλλεται από το σύστημα. Οι εφαρμογές μπορούν να επικοινωνούν μεταξύ τους χρησιμοποιώντας τους μηχανισμούς που παρέχονται από το σύστημα όπως τα αρχεία, τα *Activities*, *τα Services*, *τα BroadcastReceivers και τους ContentProviders*. Εάν μία εφαρμογή χρησιμοποιεί μόνο έναν από τους μηχανισμούς αυτούς θα πρέπει να είναι βέβαιο ότι επικοινωνεί με το σωστό μέλος, αυτό επικυρώνεται συνήθως γνωρίζοντας το δικαίωμα που συνδέεται με το δικαίωμα άσκησης. Αν μία εφαρμογή εκτίθεται σε πρόσβαση μέσω προγραμματισμού από άλλους, πρέπει να είναι βέβαιο ότι έχουν εφαρμόσει τα ανάλογα δικαιώματα, έτσι ώστε μη εξουσιοδοτημένες εφαρμογές να μην μπορούν να πάρουν προσωπικά δεδομένα του χρήστη ή να εκμεταλλευτούν την εφαρμογή. Η αρχιτεκτονική ασφάλειας μίας εφαρμογής οφείλει να είναι όσο το δυνατόν πιο απλή και σαφής. Κατά την επικοινωνία με άλλα προγράμματα, πρέπει να είναι ξεκάθαρη η πολιτική ελέγχου των εισόδων και το πως το πως επικυρώνεται η ταυτότητα των υπηρεσιών την καλούν. Πριν από το πακετάρισμα, κάθε προγραμματιστής οφείλει να έχει αναπτύξει σχέδιο επιδιόρθωσης πιθανών προβλημάτων.

Κεφάλαιο 3 Mobile και Android malware

3.1 Τύποι Απειλών

Το μοντέλο απειλών στις κινητές συσκευές περιλαμβάνει τρεις τύπους απειλών: malware, grayware και το spyware [21]. Η διάκριση μεταξύ των τριών γίνεται με βάση τη μέθοδο διανομής, τη νομιμότητά τους και το κατά πόσο ενημερώνουν το χρήστη για την δράση τους. Επίσης οι τρεις αυτές κατηγορίες απειλών έχουν διαφορετικούς φορείς της επίθεσης έχουν διαφορετικά κίνητρα και απαιτούν διαφορετικούς μηχανισμούς άμυνας.

3.1.1 Κακόβουλο Λογισμικό (Malware)

Το ιομορφικό λογισμικό ή malware αποκτά πρόσβαση σε μια συσκευή με σκοπό την κλοπή δεδομένων, την βλάβη της συσκευής ή την ενόχληση του χρήστη. Ο επιτιθέμενος εξαπατά το χρήστη και εγκαθιστά την κακόβουλη εφαρμογή ή αποκτά μη εξουσιοδοτημένη απομακρυσμένη πρόσβαση εκμεταλλευόμενος κάποια ευπάθεια του λογισμικού ή της συσκευής. Στο malware δεν παρέχεται καμία νομική ανακοίνωση στο χρήστη - θύμα. Σε αυτή την κατηγορία απειλών περιλαμβάνονται τα Trojans, τα worms, τα botnets και οι ιοί. Το malware διώκεται ποινικά σε πολλές χώρες και η διανομή του τιμωρείται με φυλάκιση. Στο Android η πιο συχνή μορφή malware είναι οι πλαστές εφαρμογές όπου ο δημιουργός του ιού εισάγει τον κακόβουλο κώδικα σε κάποια σημεία της αυθεντικής εφαρμογής και την αναδημοσιεύει ως μία νέα ή ως μία διαφορετική δωρεάν καινούρια έκδοση της αυθεντικής σε εναλλακτικά market. Ένα χαρακτηριστικό γνώρισμα του ιομορφικού αντίγραφου είναι ότι συνήθως ο πραγματικός δημιουργός μίας εφαρμογής δεν την αναδημοσιεύει σε εναλλακτικά market. Και ο προσδιορισμός των πλαστών ή παραποιημένων εφαρμογών γίνεται με την συσχέτιση των εφαρμογών που βρίσκονται στο επίσημο market με τα δείγματα που βρίσκονται στα εναλλακτικά, μη επίσημα, market.

3.1.2 Λογισμικά υποκλοπής προσωπικών δεδομένων (Personal Spyware)

Το Personal Spyware συλλέγει προσωπικές πληροφορίες, όπως η τοποθεσία ή το ιστορικό των γραπτών μηνυμάτων κατά τη διάρκεια μιας συγκεκριμένης χρονικής περιόδου. Με το Personal Spyware, ο επιτιθέμενος έχει φυσική πρόσβαση στη συσκευή και εγκαθιστά το λογισμικό χωρίς τη γνώση του χρήστη. Το Personal spyware έχει την δυνατότητα να στέλνει πληροφορίες του θύματος στο 3ο πρόσωπο που εγκατέστησε την εφαρμογή στη συσκευή του θύματος και όχι στον δημιουργό της εφαρμογής. Για παράδειγμα, ένα άτομο μπορεί να εγκαταστήσει ένα Personal spyware στο τηλέφωνό του συζύγου του/της. Σε κάποιες χώρες είναι νόμιμο να πωλείται Personal spyware, υπό την έννοια ότι δεν εξαπατάται ο αγοραστής (δηλαδή, ο κακόβουλος). Το Personal spyware είναι ειλικρινής σχετικά με το σκοπό του απέναντι στο πρόσωπο που το αγοράζει και εγκαθιστά την εφαρμογή. Ωστόσο, είναι παράνομη η εγκατάσταση του Personal spyware σε smartphone ενός ατόμου χωρίς την άδεια του.

3.1.3 Grayware

Κάποιες νόμιμες εφαρμογές συλλέγουν δεδομένα των χρηστών τους για τους σκοπούς μάρκετινγκ ή ως μέσο για στοχευμένη διαφήμιση στον χρήστη. Κατά κάποιο τρόπο μπορεί να ερμηνευτεί σαν κατασκοπία στους χρήστες αλλά οι εταιρείες που διανέμουν grayware δεν αποσκοπούν στο να βλάψουν τους χρήστες. Τμήματα του grayware προσφέρουν πραγματική λειτουργικότητα στους χρήστες. Οι εταιρείες που διανέμουν grayware μπορούν να αποκαλύπτουν τις συνήθειες των εφαρμογών τους για συλλογή δεδομένων τους στην πολιτική απορρήτου που συνοδεύει την αντίστοιχη εφαρμογή. Φυσικά αυτή η ενημέρωση συχνά τοποθετείται σε ένα αχανές κείμενο όρων χρήσης (Terms of use) και με ποικίλους βαθμούς σαφήνειας. Το Grayware κυριολεκτικά βρίσκεται στο μεταίχμιο μεταξύ της νομιμότητας και της παρανομίας, η συμπεριφορά του μπορεί να είναι νόμιμη ή παράνομη, ανάλογα με την αιτιολογία της καταγγελίας και το κείμενο της πολιτικής απορρήτου. Σε αντίθεση με το Malware ή το Spyware, το παράνομο grayware τιμωρείται με εταιρικές και όχι προσωπικές ποινές. Ακόμα και όταν η δραστηριότητα του grayware είναι νόμιμη, οι χρήστες έχουν την δυνατότητα να αντιταχθούν στη συλλογή των δεδομένων τους, αν το ανακαλύψουν. Τα Market εφαρμογών έχουν την δυνατότητα, σε κάθε περίπτωση ξεχωριστά, να επιλέξουν να αφαιρέσουν ή να κρατήσουν ένα grayware όταν αυτό ανιχνευτεί.

3.2 Χαρακτηρισμός/κατηγοριοποίηση Malware

Σε αυτή την ενότητα, παρουσιάζεται ένας συστηματικός χαρακτηρισμός των υφιστάμενων Android Malware, που ποικίλουν ανάλογα με την εγκατάσταση τους, την ενεργοποίηση τους, ακόμα και με τα κακόβουλα φορτία που μπορεί να περιέχουν.

3.2.1 Εγκατάσταση Malware

Τα υπάρχοντα Android malware μπορούν να κατηγοριοποιηθούν ανάλογα με τους υφιστάμενους τρόπους που χρησιμοποιούν για την εγκατάσταση τους σε κινητά τηλέφωνα των χρηστών και γενικεύονται σε τρεις κύριες κατηγορίες βασισμένες στην κοινωνική μηχανική δηλαδή, ανασυσκευασία (repackaging), επίθεση με ενημέρωση (update attack) και drive-by download. Οι τεχνικές αυτές δεν αλληλοαποκλείονται, ως διαφορετικές παραλλαγές του ίδιου τύπου μπορούν να χρησιμοποιούν διαφορετικές τεχνικές παράλληλα για να προσελκύσουν τους χρήστες για τη λήψη της malware εφαρμογής.

3.2.1.1 Ανασυσκευασία - (Repackaging)

Η ανασυσκευασία είναι μία από τις πιο κοινές τεχνικές που χρησιμοποιούν οι συγγραφείς malware με σκοπό να κρύψουν το κακόβουλα ωφέλιμο φορτίο σε διάφορες νόμιμες και συνήθως δημοφιλείς εφαρμογές. Στην ουσία, οι συγγραφείς malware εντοπίζουν και να αποκτούν μία δημοφιλή εφαρμογή, την αποσυναρμολογούν (disassemble), προσθέτουν τα κακόβουλα φορτία και στη συνέχεια την συναρμολογούν εκ νέου (re-assemble) και να υποβάλουν την νέα εφαρμογή στο επίσημο ή/και σε εναλλακτικά Android Market.
Οι χρήστες εκθέτονται σε κίνδυνο όταν δελεάζονται, κατεβάζουν και εγκαθιστούν αυτές τις μολυσμένες εφαρμογές. Μία προσέγγιση στην ανάλυση για Repackaging είναι όταν ένα δείγμα

μοιράζεται το ίδιο όνομα πακέτου με μια εφαρμογή στο επίσημο Android Market, τότε το δείγμα συγκρίνεται με την επίσημη, αρχική εφαρμογή και η διαφορά τους συνήθως περιέχει το κακόβουλο ωφέλιμο φορτίο που έχει προστεθεί από τον συγγραφέα του malware. Στην περίπτωση που η αρχική εφαρμογή δεν είναι διαθέσιμη, τότε ο αναλυτής πρέπει να αποσυναρμολογήσει χειροκίνητα το δείγμα και να καθορίσει αν το κακόβουλο ωφέλιμο φορτίο είναι ένα φυσικό μέρος της κύριας λειτουργικότητας της εφαρμογή ή εάν αποτελεί κομμάτι κάποιας ανασκευασμένης εφαρμογής.

	Installation				Activation								
	Repackaging	Update	Drive-by Download	Standalone	BOOT	SMS	NET	CALL	USB	PKG	BATT	SYS	MAIN
ADRD	√				√		√	√					
AnserverBot	√	√			√	√	√		√		√	√	
Asroot				√									
BaseBridge	√	√			√	√	√				√	√	
BeanBot	√					√		√					
BgServ	√				√	√							√
CoinPirate	√				√	√							
Crusewin				√	√	√							
DogWars	√												
DroidCoupon	√				√		√	√		√			
DroidDeluxe				√									
DroidDream	√												√
DroidDreamLight	√				√			√					
DroidKungFu1	√				√						√	√	
DroidKungFu2	√				√						√	√	
DroidKungFu3	√				√						√	√	
DroidKungFu4	√				√						√	√	
DroidKungFuSapp	√				√						√	√	
DroidKungFuUpdate	√	√											
Endofday	√				√	√							
FakeNetflix				√									
FakePlayer				√									
GamblerSMS				√	√								
Geinimi	√				√	√							
GGTracker			√	√	√	√					√		
GingerMaster	√				√								
GoldDream	√			√	√	√		√					
Gone60				√									
GPSSMSSpy				√		√							
HippoSMS	√				√	√							√
Jifake	√		√										
jSMSHider	√									√			√
KMin				√	√								
Lovetrap				√	√	√							
NickyBot				√	√	√							
Nickyspy				√	√								
Pjapps	√				√	√						√	
Plankton		√		√									
RogueLemon				√		√							
RogueSPPush				√		√							
SMSReplicator				√		√							
SndApps				√	√								
Spitmo			√	√		√		√					
TapSnake				√	√								
Walkinwat				√									
YZHC				√	√								
zHash				√	√								
Zitmo			√	√		√							
Zsone	√					√							√
number of families	*25*	*4*	*4*	*25*	*29*	*21*	*4*	*6*	*1*	*2*	*8*	*8*	*5*
number of samples	*1083*	*85*	*4*	*177*	*1050*	*398*	*288*	*112*	*187*	*17*	*725*	*782*	*56*

Πίνακας 2, Οικογένειες Android malware (2011-2012) εγκατάσταση και ενεργοποίηση

	Privilege Escalation					Remote Control		Financial Charges			Personal Information Stealing		
	Exploid	RATC/ Zimperlich	Ginger Break	Asroot	Encrypted	NET	SMS	Phone Call	SMS	Block SMS	SMS	Phone Number	User Account
ADRD						√							
AnserverBot						√			√ †				
Asroot				√									
BaseBridge		√				√		√	√ †	√			
BeanBot						√		√	√ †	√		√	
BgServ						√			√ †	√		√	
CoinPirate						√			√ †	√	√		
Crusewin						√			√	√	√		
DogWars									√				
DroidCoupon		√				√							
DroidDeluxe		√											
DroidDream	√	√				√							
DroidDreamLight						√							√
DroidKungFu1	√	√			√	√						√	
DroidKungFu2	√	√			√	√						√	
DroidKungFu3	√	√			√	√						√	
DroidKungFu4						√							
DroidKungFu5	√	√			√	√						√	
DroidKungFuUpdate													
Endofday						√			√			√	
FakeNetflix													√
FakePlayer									√ ‡				
GamblerSMS											√		
Geinimi						√		√	√ †	√	√	√	
GGTracker									√ ‡	√	√	√	
GingerMaster			√			√						√	
GoldDream						√		√	√ †		√	√	
Gone60											√		
GPSSMSSpy									√				
HippoSMS									√ ‡	√			
Jifake									√ ‡				
jSMSHider						√			√ †	√		√	
KMin						√			√ †	√			
Lovetrap									√ †	√			
NickyBot							√		√		√		
Nickyspy						√			√		√		
Pjapps						√			√ †	√		√	
Plankton						√							
RogueLemon						√			√ †	√	√		
RogueSPPush									√ ‡	√			
SMSReplicator									√		√		
SndApps													√
Spitmo						√			√ †	√	√	√	
TapSnake													
Walkinwat									√				
YZHC						√			√ ‡	√		√	
zHash	√												
Zitmo											√		
Zsone									√ ‡	√			
number of families	*6*	*8*	*1*	*1*	*4*	*27*	*1*	*4*	*28*	*17*	*13*	*15*	*3*
number of samples	*389*	*440*	*4*	*8*	*363*	*1171*	*1*	*246*	*571*	*315*	*138*	*563*	*43*

Πίνακας 3, Οικογένειες Android malware (2011-2012) στόχοι επιθέσεων

Στην έρευνα που έγινε από τους Yajin Zhou και Xuxian Jiang, Dissecting Android Malware: Characterization and Evolution [22] παρατηρήθηκε ότι συνολικά από τα 1260 δείγματα malware εκ των οποίων τα 1083 (ή 86,0%) ήταν προϊόντα ανασυσκευασίας. Επίσης με την περαιτέρω κατάταξη ανά οικογένεια malware (Πίνακας 2) διαπιστώνεται ότι σε ένα σύνολο 49 οικογενειών που μολύνουν τους χρήστες, 25 εξ αυτών είναι προϊόντα ανασυσκευασίας ενώ 25 από αυτές είναι αυτόνομες εφαρμογές,εξ αυτών, οι περισσότερες είναι spyware. Η οικογένεια malware η GoldDream ανήκει και στις δύο κατηγορίες. Μεταξύ των 1083 ανασκευασμένων εφαρμογών, οι συγγραφείς malware έχουν επιλέξει μια ποικιλία από τις κανονικές εφαρμογές προς ανασυσκευασία που περιλαμβάνει αμειβόμενες, δημοφιλείς εφαρμογές αλλά και παιχνίδια ή ισχυρές εφαρμογές συστήματος (συμπεριλαμβανομένων και των ενημερώσεων ασφαλείας), καθώς και πορνογραφικές εφαρμογές. Για παράδειγμα, το δείγμα malware το AnserverBot[18] ήταν η ανασκευασμένη εφαρμογή com.camelgames.mxmotor που ήταν διαθέσιμη στο επίσημο Android Market. Ενώ το δείγμα BgServ[19] [23] malware αποτελούσε ανασυσκευασία του

18 SHA1: ef140ab1ad04bd9e52c8c5f2fb6440f3a9ebe8ea
19 SHA1: bc2dedad0507a916604f86167a9fa306939e2080

εργαλείου ασφαλείας που είχε διανέμει η Google για την αφαίρεση του DroidDream από μολυσμένες συσκευές. Επίσης, στην προσπάθεια τους να κρύψουν το κακόβουλα ωφέλιμο φορτίο, οι συγγραφείς malware έχουν την τάση να χρησιμοποιούν class-file ονόματα που φαίνονται νόμιμα και καλοήθεις. Για παράδειγμα, το malware AnserverBot χρησιμοποιεί το com.sec.android.provider.drm για όνομα του πακέτου για το ωφέλιμο φορτίο του, το οποίο μοιάζει με ένα module που παρέχει νόμιμη λειτουργία DRM. Η πρώτη έκδοση του DroidKungFu χρησιμοποιεί com.google.ssearch ώστε να παρίσταται το Google search module. Έτσι στις ακόλουθες εκδόσεις του που χρησιμοποιούν το com.google.update, ώστε να προσποιείται ότι είναι μια επίσημη ενημέρωση της Google. Έχει ενδιαφέρον να σημειωθεί ότι μία οικογένεια malware – η jSMSHider χρησιμοποιεί το διαθέσιμο στο κοινό ιδιωτικό κλειδί (με αύξοντα αριθμό: b3998086d056cffa) που διανέμεται στο Android έργο ανοικτού πηγαίου κώδικα (AOSP). Το τρέχον μοντέλο ασφαλείας του Android επιτρέπει στις εφαρμογές που έχουν υπογράφει με το ίδιο κλειδί πλατφόρμας λογισμικού του τηλεφώνου (phone firmware) να μπορούν να αιτούνται δικαιώματα, τα οποία δεν είναι διαθέσιμα με άλλο τρόπο σε εφαρμογές τρίτων. Ένα τέτοιο δικαίωμα περιλαμβάνει την δυνατότητα εγκατάστασης επιπρόσθετων εφαρμογών χωρίς την παρέμβαση του χρήστη. Δυστυχώς, στο παρελθόν μερικά παραμετροποιημένα firmware images υπογράφηκαν από το προεπιλεγμένο κλειδί που διανέμεται σε AOSP. Ως αποτέλεσμα, οι εφαρμογές jSMSHiderinfected μπορούν να αποκτήσουν προνομιακά δικαιώματα για την εκτέλεση επικίνδυνων εργασιών χωρίς την ευαισθητοποίηση και ενημέρωση του χρήστη.

3.2.1.2 Επίθεση μέσω ενημέρωσης (update attack)

Στην πρώτη τεχνική update attack συνήθως κρύβεται ολόκληρο το κακόβουλο ωφέλιμο φορτίο σε απομακρυσμένες (host) εφαρμογές, οι οποίες δυνητικά μπορούν να εκθέσουν την παρουσία τους. Η δεύτερη τεχνική, επίσης στοχεύει στο να ανασυσκευάζονται δημοφιλείς εφαρμογές όμως είναι αρκετά πιο δύσκολο να ανιχνευτεί διότι αντί να περικλείει το κακόβουλο φορτίο στο σύνολό του, περιλαμβάνει μόνο ένα τμήμα κώδικα που όταν εκτελεστεί θα λάβει τα κακόβουλα ωφέλιμα φορτία σε κάποιο δεύτερο χρόνο όπως πχ. κατά το χρόνο εκτέλεσης. Ως εκ τούτου, μια στατική σάρωση των host εφαρμογών μπορεί να αποτύχει στο να εντοπίσει τα κακόβουλα ωφέλιμα φορτία. Στο σύνολο των δεδομένων της έρευνας [22] οι οικογένειες κακόβουλου λογισμικού BaseBridge, DroidKungFuUpdate, AnserverBot και Plankton υιοθετούν αυτή την επίθεση (Πίνακας 2). Το malware BaseBridge διαθέτει μια σειρά από παραλλαγές αρκετές από τις οποίες εκμεταλλεύονται root exploits, που επιτρέπουν την αθόρυβη εγκατάσταση πρόσθετων εφαρμογών χωρίς την παρέμβαση του χρήστη και άλλες παραλλαγές χρησιμοποιούν update attack χωρίς να εκμεταλλεύονται άλλες αδυναμίες. Πιο συγκεκριμένα, όταν μία εφαρμογή μολυσμένη με BaseBridge εκτελεστεί, θα ελέγξει εάν μπορεί να εμφανίσει ένα διάλογο ενημέρωσης. Εάν ναι, τότε ενημερώνει τον χρήστη ότι μια νέα έκδοση είναι διαθέσιμη, ο χρήστης συχνά προσφέρεται να εγκαταστήσει την ενημερωμένη έκδοση. (Η νέα έκδοση είναι αποθηκευμένη σε κάποιο host ή σε κάποιο αρχείο.) Όταν ο χρήστης αποδέχεται την εγκατάσταση, μια "ενημερωμένη" έκδοση με το κακόβουλο ωφέλιμο φορτίο θα εγκατασταθεί. Επειδή το κακόβουλο ωφέλιμο φορτίο είναι στην "ενημερωμένη" εφαρμογή και όχι στην αρχική εφαρμογή είναι πιο κρυφό από την πρώτη τεχνική που περιλαμβάνει άμεσα ολόκληρο το κακόβουλο ωφέλιμο φορτίο εξ αρχής. Το malware DroidKungFuUpdate είναι παρόμοιο με το BaseBridge. Όμως, αντί να μεταφέρει ή να περικλείει την «ενημερωμένη» έκδοση στο εσωτερικό της αρχικής εφαρμογής, επιλέγει να κατεβάζει εξ αποστάσεως μια νέα έκδοση από το δίκτυο. Επιπλέον, παραμένει κρυφή σαν εφαρμογή χρησιμοποιώντας για

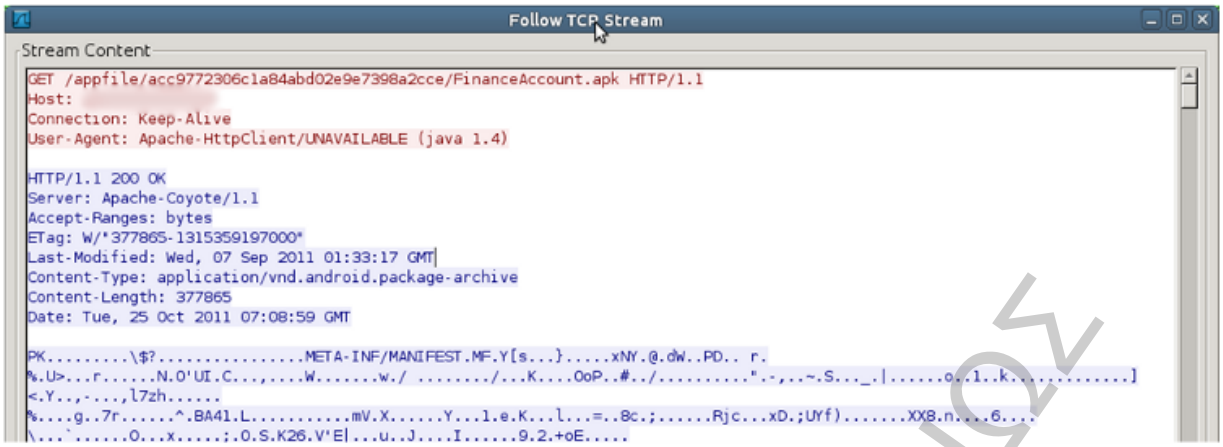
κοινοποίηση προς τους χρήστες μια βιβλιοθήκη[20] η οποία παρέχει τη (νόμιμη) λειτουργικότητα της κοινοποίησης.

Η εικόνα 14 δείχνει την κίνηση του δικτύου όπως έχει συλληφθεί ξεκινώντας από την αρχική εφαρμογή να ενημερώνεται από τον host. Μετά τη λήψη, η "ενημερωμένη" εκδοσή αποδεικνύεται ότι είναι το DroidKungFu3 malware. Το DroidKungFuUpdate ήταν διαθέσιμο και στο επίσημο Android Market όπως και στα εναλλακτικά. Οι δύο προηγούμενες update attacks απαιτούν την έγκριση του χρήστη για να γίνει λήψη και εγκατάσταση της νέας έκδοσης. Τα επόμενα δύο malware, δηλαδή, το AnserverBot και το Plankton, βελτιώνουν την update attack, αναβαθμίζοντας και ορισμένα στοιχεία της εφαρμογής παράλληλα με το κακόβουλο πακέτο. Ως αποτέλεσμα, δεν απαιτείται η έγκριση του χρήστη. Ειδικότερα, το Plankton έκανε άμεση λήψη και εκτελούσε ένα αρχείο jar από έναν απομακρυσμένο server, ενώ το AnserverBot ανακτά μία δημόσια, κρυπτογραφημένη είσοδο σε host όπου περιέχεται το κακόβουλα ωφέλιμο φορτίο της ενημέρωσης. Προφανώς, ο λαθραίος χαρακτήρας αυτών των επιθέσεων ενημέρωσης θέτει σημαντικές προκλήσεις για τον εντοπισμό τους (Πίνακας 4).

20 waps.cn

	#	AVG		Lookout		Norton		Trend Micro	
		#	%	#	%	#	%	#	%
ADRD	22	22	100.0	13	59.0	5	22.7	11	50.0
AnserverBot	187	165	88.2	89	47.5	2	1.0	57	30.4
Asroot	8	3	37.5	0	0.0	0	0.0	6	75.0
BaseBridge	122	110	90.1	112	91.8	40	32.7	119	97.5
BeanBot	8	0	0.0	0	0.0	0	0.0	0	0.0
Bgserv	9	9	100.0	1	11.1	2	22.2	9	100.0
CoinPirate	1	0	0.0	0	0.0	0	0.0	0	0.0
CruseWin	2	0	0.0	2	100.0	2	100.0	2	100.0
DogWars	1	1	100.0	1	100.0	1	100.0	1	100.0
DroidCoupon	1	0	0.0	0	0.0	0	0.0	0	0.0
DroidDeluxe	1	1	100.0	1	100.0	0	0.0	1	100.0
DroidDream	16	11	68.7	16	100.0	9	56.2	16	100.0
DroidDreamLight	46	14	30.4	45	97.8	11	23.9	46	100.0
DroidKungFu1	34	34	100.0	34	100.0	2	5.8	33	97.0
DroidKungFu2	30	30	100.0	30	100.0	1	3.3	30	100.0
DroidKungFu3	309	0	0.0	307	99.3	1	0.3	305	98.7
DroidKungFu4	96	4	4.1	96	100.0	2	2.0	12	12.5
DroidKungFuSapp	3	0	0.0	0	0.0	0	0.0	0	0.0
DroidKungFuUpdate	1	0	0.0	1	100.0	0	0.0	0	0.0
Endofday	1	1	100.0	1	100.0	1	100.0	1	100.0
FakeNetflix	1	0	0.0	1	100.0	1	100.0	1	100.0
FakePlayer	6	6	100.0	6	100.0	6	100.0	6	100.0
GamblerSMS	1	0	0.0	0	0.0	0	0.0	1	100.0
Geinimi	69	69	100.0	69	100.0	38	55.0	67	97.1
GGTracker	1	1	100.0	1	100.0	1	100.0	1	100.0
GingerMaster	4	4	100.0	4	100.0	4	100.0	4	100.0
GoldDream	47	29	61.7	40	85.1	19	40.4	47	100.0
Gone60	9	9	100.0	9	100.0	4	44.4	7	77.7
GPSSMSSpy	6	0	0.0	6	100.0	2	33.3	3	50.0
HippoSMS	4	0	0.0	2	50.0	2	50.0	2	50.0
Jifake	1	0	0.0	1	100.0	0	0.0	1	100.0
jSMSHider	16	11	68.7	16	100.0	13	81.2	16	100.0
KMin	52	52	100.0	0	0.0	40	76.9	52	100.0
LoveTrap	1	0	0.0	1	100.0	1	100.0	1	100.0
NickyBot	1	0	0.0	0	0.0	0	0.0	0	0.0
NickySpy	2	2	100.0	2	100.0	2	100.0	2	100.0
Pjapps	58	44	75.8	57	98.2	26	44.8	50	86.2
Plankton	11	11	100.0	0	0.0	1	9.0	6	54.5
RogueLemon	2	0	0.0	0	0.0	0	0.0	0	0.0
RogueSPPush	9	9	100.0	3	33.3	0	0.0	8	88.8
SMSReplicator	1	1	100.0	1	100.0	1	100.0	1	100.0
SndApps	10	10	100.0	6	60.0	0	0.0	4	40.0
Spitmo	1	1	100.0	1	100.0	1	100.0	1	100.0
Tapsnake	2	0	0.0	2	100.0	1	50.0	1	50.0
Walkinwat	1	0	0.0	1	100.0	1	100.0	1	100.0
YZHC	22	1	4.5	1	4.5	3	13.6	10	45.4
zHash	11	11	100.0	11	100.0	2	18.1	11	100.0
Zitmo	1	1	100.0	1	100.0	1	100.0	1	100.0
Zsone	12	12	100.0	12	100.0	5	41.6	12	100.0
Detected Samples (out of 1260)		*689* (54.7%)		*1003* (79.6%)		*254* (20.2%)		*966* (76.7%)	

Πίνακας 4, Αποτελέσματα εντοπισμού ανά οικογένεια malware



Εικόνα 14, DroidKungFuUpdate, κίνηση του δικτύου όπως έχει συλληφθεί από το Wireshark

3.2.1.3 Drive-by download

Η τρίτη τεχνική εφαρμόζει την παραδοσιακή drive-by download επίθεση σε κινητό περιβάλλον. Δεν εκμεταλλεύεται απευθείας τα τρωτά σημεία του browser, αλλά αυτό που κάνει ουσιαστικά είναι να προσελκύει τους χρήστες να κατεβάζουν "ενδιαφέρουσες" ή "πλούσιες σε χαρακτηριστικά" εφαρμογές. Στην έρευνα των Yajin Zhou και Xuxian Jiang [22], εντοπίζονται τέσσερις τέτοιες οικογένειες κακόβουλου λογισμικού, δηλαδή GGTracker [24], Jifake [25], Spitmo [26] και ZitMo [27]. Τα δύο τελευταία έχουν σχεδιαστεί για να υποκλέπτουν ευαίσθητες τραπεζικές πληροφορίες. Το malware GGTracker εμφανίζεται ως μία επιπρόσθετη εφαρμογή που προβάλει διαφημίσεις, αλλά με την διαφορά ότι ο σύνδεσμος προς το προϊόν ανακατευθύνει σε κάποιο κακόβολο ιστότοπο. Συγκεκριμένα ισχυρίζεται ότι αναλύει την χρήση της μπαταρίας και που γίνεται στην συσκευή και θα ανακατευθύνει το χρήστη σε ένα ψεύτικο το Android Market για να κατεβάσει μια εφαρμογή που ισχυρίζεται ότι βελτιώνει την απόδοσης της μπαταρίας. Δυστυχώς, η λήψη της νέας αυτής πολλά υποσχόμενης εφαρμογής δεν είναι αυτή που εστιάζει στη βελτίωση της αποτελεσματικότητας της μπαταρίας, αλλά ένα κακόβουλο λογισμικό που εγγραφεί τον χρήστη σε μια υπηρεσία υψηλής χρέωσης εν άγνοια του. Παρομοίως, το Jifake εισέρχεται στην συσκευή όταν οι χρήστες ανακατευθύνονται στον κακόβουλο ιστοχώρο. Η διαφορά του σε σχέση με το GGTracker είναι ότι δεν εμφανίζεται ως πλατφόρμα διαφήμισης για να προσελκύει και να ανακατευθύνει αντί αυτού, χρησιμοποιεί ένα κακόβουλο QR code[21], το οποίο όταν ανιχνευτεί θα ανακατευθύνει το χρήστη σε μία διεύθυνση URL που περιέχει το trojan Jifake. Αυτό το κακόβουλο λογισμικό είναι το προϊόν ανακατασκευής του mobile ICQ client και είναι ρυθμισμένο να στέλνει πολλά SMS μηνύματα σε έναν αριθμό υψηλής χρέωσης. Ενώ διάδοση malware που βασίζεται σε QR code είναι γνωστή και από το παρελθόν [28], η περίπτωση του Jifake είναι η πρώτη φορά ότι αυτή η επίθεση ήταν επιτυχής. Τα δύο τελευταία ZitMo και Spitmo είναι εκδόσεις για το Android των ιών Zeus και SpyEye αντίστοιχα. Λειτουργούν με παρόμοια με τις προηγούμενες οικογένειες drive-by download attack. Όταν ένας χρήστης κάνει online τραπεζικές συναλλαγές από μία παραβιασμένη συσκευή, ανακατευθύνεται για να κάνει λήψη μίας συγκεκριμένης εφαρμογής που ισχυρίζεται ότι παρέχει καλύτερη προστασία για ηλεκτρονικές συναλλαγές. Η εφαρμογή αντ' αυτού είναι malware, το οποίο μπορεί να συλλέξει και να στείλει mTANs[22] ή SMS μηνύματα σε έναν απομακρυσμένο server. Αυτές οι δύο οικογένειες drive-by download malware στηρίζονται σε παραβιασμένους browser για να ξεκινήσει η επίθεση. Αν και φαινομενικά είναι δύσκολο να μολύνουν πραγματικά τους χρήστες, το γεγονός ότι μπορούν να υποκλέψουν τραπεζικές ευαίσθητες πληροφορίες εγείρει σοβαρές ανησυχίες για τους χρήστες.

21 Ο **κώδικας QR** είναι ένας γραμμωτός κώδικας (barcode) δύο διαστάσεων

22 Ακρώνυμο του moblie Transaction authentication number, που χρησιμοποιείται για mobile banking.

3.2.1.4 Άλλες μορφές

Μέχρι στιγμής έχουν παρουσιαστεί τρεις τεχνικές κυρίως βασισμένες στην κοινωνική μηχανική. Υπάρχουν όμως περιπτώσεις που δεν ανήκουν σε μία από αυτές τις κατηγορίες. (Το σύνολο δειγμάτων της ερευνάς των Yajin Zhou και Xuxian Jiang [22] περιέχει 177 αυτόνομες εφαρμογές). Οι λοιπές αυτές κακόβουλες εφαρμογές μπορούν να κατηγοριοποιηθούν εκ νέου σε 4 ομάδες εφαρμογών.

Η πρώτη ομάδα θεωρείται spyware, διότι οι εφαρμογές που ανήκουν σε αυτήν προτίθενται να εγκατασταθούν στο τηλέφωνο του θύματος επίτηδες, ως επιθυμία του χρήστη. Αυτό εξηγεί πιθανώς γιατί οι επιτιθέμενοι δεν έχουν κίνητρα ή ανάγκη να δελεάσουν θύμα για την εγκατάσταση. Παράδειγμα αποτελεί το GPSSMSSpy που εκτελεί εντολές μέσω SMS για την καταγραφή και να αποστολή της τρέχουσας θέσης του θύματος.

Η δεύτερη ομάδα περιλαμβάνει τις πλαστές εφαρμογές που μεταμφιέζονται όπως οι νόμιμες αλλά εκτελούν κρυφά κακόβουλες ενέργειες, όπως η κλοπή των διαπιστευτηρίων των χρηστών ή η αποστολή Μηνύματα SMS από το παρασκήνιο. Το FakeNetflix είναι ένα παράδειγμα αυτής της ομάδας που κλέβει τον λογαριασμό και τον κωδικό πρόσβασης ενός χρήστη του Netflix. Δεν είναι μια ανακατασκευασμένη εκδοχή της εφαρμογής Netflix, αλλά αντ 'αυτού μεταμφιέζεται ώστε να είναι η εφαρμογή Netflix με την ίδια διεπαφή χρήστη. Το FakePlayer αποτελεί ένα άλλο παράδειγμα που μεταμφιέζεται ως ένα movie player, χωρίς τις ενοχλητικές διαφημίσεις. Το μόνο που κάνει είναι να στέλνει μηνύματα SMS υψηλής χρέωσης χωρίς την γνώση των χρηστών.

Η τρίτη ομάδα περιλαμβάνει εφαρμογές που επίσης σκοπίμως περιλαμβάνουν κακόβουλες λειτουργίες (π.χ. αποστολή μη εξουσιοδοτημένων SMS μηνυμάτων ή την αυτόματη εγγραφή σε κάποια υπηρεσία προστιθέμενης αξίας). Αλλά η διαφοροποίηση από τη δεύτερη ομάδα είναι ότι οι εφαρμογές δεν είναι ψεύτικες. Αντ 'αυτού, μπορούν να παρέχουν τη λειτουργικότητα που ισχυρίζονται. Ωστόσο, χωρίς να το γνωρίζουν οι χρήστες επιπροσθέτως περιλαμβάνουν ορισμένες κακόβουλες λειτουργικότητες. Για παράδειγμα, Το RogueSPPush είναι μια εφαρμογή για αστρολογία. Αλλά αυτομάτως εγγραφεί τους χρήστες του σε υπηρεσίες υψηλού κόστους αποκρύπτοντας τα SMS μηνύματα επιβεβαίωσης. Η τελευταία ομάδα περιλαμβάνει τις εφαρμογές που βασίζονται στο root privilege για να λειτουργήσουν. Ωστόσο, χωρίς να επιβεβαιώσει ο χρήστης την χορήγηση του root privilege σε αυτές τις εφαρμογές αξιοποιώντας κάποιο γνωστό root exploit παρακάμπτουν την ασφάλεια του λειτουργικού. Αν και αυτές οι εφαρμογές ενδέχεται να μην καταδεικνύουν σαφώς κακόβουλες προθέσεις, το γεγονός όμως του ότι εκτελούν Privilege Escalation και ανακτούν δικαιώματα διαχειριστή χωρίς την άδεια του χρήστη δεν τους επιτρέπει να χαρακτηριστούν και νόμιμα. Παραδείγματα σε αυτή την ομάδα αποτελούν οι Asroot και DroidDeluxe.

Abbreviation	Events	Abbreviation	Events	Abbreviation	Events
BOOT (Boot Completed)	BOOT_COMPLETED	SMS (SMS/MMS)	SMS_RECEIVED WAP_PUSH_RECEIVED	NET (Network)	CONNECTIVITY_CHANGE PICK_WIFI_WORK
CALL (Phone Events)	PHONE_STATE NEW_OUTGOING_CALL	USB (USB Storage)	UMS_CONNECTED UMS_DISCONNECTED	MAIN (Main Activity)	ACTION_MAIN
PKG (Package)	PACKAGE_ADDED PACKAGE_REMOVED PACKAGE_CHANGED PACKAGE_REPLACED PACKAGE_RESTARTED PACKAGE_INSTALL	BATT (Power/Battery)	ACTION_POWER_CONNECTED ACTION_POWER_DISCONNECTED BATTERY_LOW BATTERY_OKAY BATTERY_CHANGED_ACTION	SYS (System Events)	USER_PRESENT INPUT_METHOD_CHANGED SIG_STR SIM_FULL

Πίνακας 5, Συσχέτιση Android γεγονότων με κακόβουλη λειτουργία

3.2.2 Ενεργοποίηση malware

Στη συνέχεια, θα εξεταστούν τα system-wide Android events του Android που χρησιμοποιούν τα malware. Με την εγγραφή στα σχετικά system-wide γεγονότα, μία εφαρμογή μπορεί να βασιστεί στην ενσωματωμένη υποστήριξη των αυτόματων ενημερώσεων από το σύστημα όταν εγείρεται ένα γεγονός το οποίο της έχει καταχωρηθεί. Έτσι και τα malware αναμένουν κάποιο συγκεκριμένο notification ή callback ώστε να εκκινήσουν την εκτέλεση του κακόβουλα ωφέλιμου φορτίου τους. Τα πιο συχνά Android γεγονότα παρουσιάζονται στον Πίνακα 5 και κάθε οικογένεια malware που αναφέρεται, επίσης σχετίζεται με τα γεγονότα στον Πίνακα 2. Μεταξύ όλων των διαθέσιμων γεγονότων του συστήματος, το BOOT_COMPLETED αποτελεί το πλέον ενδιαφέρον στα υπάρχοντα Android malware. Αυτό το γεγονός δεν προκαλεί έκπληξη καθώς αυτό το συγκεκριμένο γεγονός ενεργοποιείται όταν το σύστημα ολοκληρώνει την διαδικασία της εκκίνησης του λειτουργικού - ένα τέλειο χρονικά σημείο για ένα malware να δώσει το έναυσμα για τις υπηρεσίες παρασκηνίου (πχ ένα Service). Στην έρευνα των Yajin Zhou και Xuxian Jiang [22], 29 οικογένειες malware (το 83,3% των δειγμάτων) αναμένουν αυτό το γεγονός. Για παράδειγμα το Geinimi[23] ακούει σε αυτό το γεγονός για να εκκινήσει την υπηρεσία φόντου -com.geinimi.AdService. Το γεγονός SMS_RECEIVED έρχεται δεύτερο με 21 οικογένειες malware να ενδιαφέρονται για αυτό. Αυτό είναι επίσης λογικό, διότι πολλά malware είναι πρόθυμα για παρεμπόδιση ή διαχείριση εισερχόμενων SMS μηνυμάτων. Ένα παράδειγμα αποτελεί, το zSone, που ακούει το SMS_RECEIVED και παρακολουθεί ή αφαιρεί όλα τα μηνύματα SMS από συγκεκριμένα νούμερα προέλευσης όπως "10086" και "10010". Επίσης έχει παρατηρηθεί ότι ορισμένα malware κατοχυρώνουν μια σειρά από γεγονότα συστήματος. Για παράδειγμα, το AnserverBot κατοχυρώνει callbacks από 10 διαφορετικά γεγονότα, ενώ το BaseBridge ενδιαφέρεται για 9 διαφορετικά γεγονότα. Η εγγραφή σε ένα μεγάλο αριθμό γεγονότων αναμένεται να επιτρέψει στο malware να ξεκινάει αξιόπιστα και γρήγορα τα κακόβουλα φορτία του. Επιπλέον, έχει παρατηρηθεί επίσης ότι μερικά δείγματα malware δεσμεύουν άμεσα κάποιο Activity από τις εφαρμογές του λειτουργικού, το οποίο ενεργοποιείται όταν ο χρήστης κάνει κλικ στο εικονίδιο της εφαρμογής στην αρχική οθόνη είτε όταν κάποιο Intent με ενέργεια ACTION_MAIN παραλαμβάνεται από την εφαρμογή. Η κακόβουλη αυτή δέσμευση του entry Activity επιτρέπει στο malware να αντικαταστήσει μία βασική υπηρεσία πριν από την έναρξη της κύριας δραστηριότητας της εφαρμογής. Για παράδειγμα, το DroidDream[24] αντικαθιστά την αρχική δραστηριότητα (entry Activity) με τη δικιά του com.android.root.main έτσι ώστε να μπορεί να αποκτήσει τον έλεγχο, ακόμη και πριν

23 SHA1: 179e1c69ceaf2a98fdca1817a3f3f1fa28236b13

24 SHA1:fdf6509b4911485b3f4783a72fde5c27aa9548c7

ξεκινήσει η αρχική δραστηριότητα com.codingcaveman.SoloTrial.SplashActivity. Άλλα malware παρεμβαίνουν σε ορισμένα γεγονότα αλληλεπίδρασης UI (π.χ. κάνοντας κλικ στο κουμπί). Ένα παράδειγμα είναι το zSone[25] malware που επικαλείται τον δικό του κωδικά αποστολής SMS μέσα στη μέθοδο onClick() της εφαρμογής.

3.2.3 Κακόβουλα ωφέλιμα φορτία

Τα υπάρχοντα Android malware μπορούν να χαρακτηριστούν σε μεγάλο βαθμό με βάση ωφέλιμο φορτίο που φέρουν. Η κατηγοριοποίηση αυτή γίνεται με βάση την λειτουργικότητα του ωφέλιμου φορτίου σε τέσσερις διαφορετικές κατηγορίες: κλιμάκωση προνομίων (privilege escalation), απομακρυσμένη πρόσβαση (remote control), οικονομικές χρεώσεις (financial charges), καθώς και κλοπή προσωπικών πληροφοριών (personal information stealing).

3.2.3.1 Κλιμάκωση προνομίων (privilege escalation)

Η πλατφόρμα Android είναι ένα περίπλοκο σύστημα που αποτελείται όχι μόνο από τον Linux πυρήνα, αλλά και ολόκληρο το πλαίσιο Android με περισσότερες από 90 open-source βιβλιοθήκες να συμπεριλαμβάνονται, όπως η WebKit, η SQLite και το OpenSSL.

Η πολυπλοκότητα αυτή φυσικά εισάγει τρωτά σημεία λογισμικού που μπορούν δυνητικά να αξιοποιηθούν για κλιμάκωση προνομίων. Στον δικτυακό τόπο **cvedetails**[26] υπάρχει μία εκτενής λίστα με τα γνωστά τρωτά σημεία επιπέδου πλατφόρμας στο Android που μπορούν να αξιοποιηθούν κακόβουλα, επίσης δείχνει τη λίστα των malware που εκμεταλλεύονται αυτά τα τρωτά σημεία για τη διευκόλυνση της εκτέλεσης των ωφέλιμων φορτίων τους. Συνολικά, υπάρχει ένας πολύ μικρός αριθμός ευπαθειών σε επίπεδο πλατφόρμας που είναι ενεργά εκμεταλλεύσιμες αυτή τη στιγμή. Οι τρεις κυριότερες ευπάθειες είναι η exploid, η RATC[27] (ή RageAgainstTheCage), asroot και η Zimperlich. Στην ανάλυσή των Yajin Zhou και Xuxian Jiang [22], ένα ανησυχητικό αποτέλεσμα είναι ότι μεταξύ 1260 δειγμάτων στο σύνολο δεδομένων, 463 από αυτούς (36,7%), τουλάχιστον ενσωματώνουν ένα root exploit (Πίνακας 3). Όσον αφορά την δημοτικότητα του καθενός exploit, υπάρχουν 389, 440, 4 και 8 δείγματα που περιέχουν exploid, RATC, GingerBreak και asroot, αντίστοιχα. Επίσης, δεν είναι ασυνήθιστο για ένα malware να έχει δύο ή περισσότερα root exploits για να μεγιστοποιήσει τις πιθανότητές του για την επιτυχή εκμετάλλευση σε πολλαπλές εκδόσεις της πλατφόρμας. (στην ίδια έρευνα, Υπάρχουν 378 δείγματα με περισσότερα από ένα root exploits.) Μια περαιτέρω ανάλυση σχετικά με το πώς αυτά τα exploits χρησιμοποιούνται στην πραγματικότητα δείχνει ότι πολλά από τα παλαιότερα malware απλά να αντιγραφούν κατά γράμμα τα root exploits που είναι διαθέσιμα στο Διαδίκτυο χωρίς καμία τροποποίηση, ακόμη και χωρίς την αφαίρεση των αρχικών μηνυμάτων εξόδου εντοπισμού σφαλμάτων ή κάποια αλλαγή των ονομάτων των αρχείων που συνδέονται με το εκάστοτε root exploit. Για παράδειγμα, το DroidDream περιέχει το όνομα του αρχείου exploid ακριβώς το ίδιο με το δημοσιευμένο exploit. Γεγονός που βοηθούσε στον εντοπισμό τους κατά την ανάλυση. Ωστόσο, τα πράγματα έχουν έχουν αλλάξει πρόσφατα. Το DroidKungFu δεν ενσωματώνει αυτούσια τα root exploits που χρησιμοποιεί. Αντ

25 SHA1:00d6e661f90663eeffc10f64441b17079ea6f819

26 http://www.cvedetails.com/vulnerability-list/vendor_id-1224/product_id-19997/Google-Android.html

27 RageAgainstTheCage, Κεφάλαιο 3.3.3

'αυτού, πρώτα κρυπτογραφεί αυτά τα root exploits και στη συνέχεια τα αποθηκεύει ως πόρο ή σαν αρχείο ιδιοτήτων. Κατά τον χρόνο εκτέλεσης, αποκαλύπτει αυτά τα δυναμικά κρυπτογραφημένα root exploits και στη συνέχεια τα εκτελεί ανάλογα, γεγονός που καθιστά ανίχνευση τους πολύ δύσκολη. Στην πραγματικότητα, όταν η πρώτη έκδοση του DroidKungFu ανακαλύφθηκε, είχε αναφερθεί ότι εκείνη τη στιγμή κανένα υπάρχον anti-virus σε κινητό δεν ήταν σε θέση να ανιχνεύσει, γεγονός που κατέδειξε την αποτελεσματικότητα αυτής της προσέγγισης. Επιπλέον, άλλα πρόσφατα malware, όπως DroidCoupon και GingerMaster εφαρμόζουν obfuscate τεχνικές στα ονόματα των αρχείων που συνδέονται με τα root exploits (όπως π.χ. την προσποίηση πως είναι αρχεία εικόνας με κατάληξη png). Γεγονός που ανακλά την εξελισσόμενη φύση του malware.

3.2.3.2 Απομακρυσμένη πρόσβαση (remote control)

Ένα πολύ μεγάλο ποσοστό malware μετατρέπει τις μολυσμένες συσκευές σε bots για remote control. Συγκεκριμένα στην έρευνα των Yajin Zhou και Xuxian Jiang [22], υπάρχουν 1.171 (93,0%), δείγματα που χρησιμοποιούν το HTTP πρωτόκολλο για να λαμβάνουν bot εντολές από τους servers C&C[28] τους. Επίσης παρατηρείται ότι κάποιες οικογένειες malware επιχειρούν να κρύβουν τους απομακρυσμένους server τους κρυπτογραφώντας τις διευθύνσεις URL καθώς και την επικοινωνία μαζί τους, όπως για παράδειγμα, η οικογένεια malware Pjapps[29] που χρησιμοποιεί το δικό της σύστημα κωδικοποίησης για να κρυπτογραφεί τις διευθύνσεις του C&C server της. Ένα από τα δείγματα της οικογενείας Pjapps κωδικοποιεί τον C&C server από mobilemeego91.com σε 2maodb3ialke8mdeme3gkos9g1icaofm. Ενώ το DroidKungFu3 χρησιμοποιεί AES encryption (με κλειδί το “Fuck_sExy-Key ALL! Pw”) για να κρύψει τους C&C servers του. Παρομοίως το Geinimi εφαρμόζει DES κρυπτογράφηση (με κλειδί 0x0102030405060708) κατά την επικοινωνία του με τους απομακρυσμένους C&C servers. Όταν αναλύθηκαν αυτές οι οικογένειες malware, αποδείχτηκε επίσης ότι οι περισσότεροι από αυτούς τους C&C Servers είναι εγγεγραμμένοι σε domains που ελέγχονται από τους ίδιους τους επιτιθέμενους. Ωστόσο, εντοπίστηκαν και περιπτώσεις όπου οι C&C servers βρίσκονταν σε δημόσια cloud. Για παράδειγμα, το Plankton spyware κατεβάζει δυναμικά και τρέχει το ωφέλιμο φορτίο του από ένα server που φιλοξενείται στο Amazon cloud. Πιο πρόσφατα, οι εισβολείς στρέφονται στα δημόσιους blog servers ως C&C servers. Το AnserverBot[30] είναι ένα παράδειγμα που χρησιμοποιεί τους Sina και Baidu, δύο δημοφιλείς κινέζικους δημόσιους blog servers.

3.2.3.3 Οικονομική επιβάρυνση

Εκτός από τις οπτικές του privilege escalation και του remote control είναι χρήσιμο επίσης να εντοπιστούν τα κίνητρα πίσω από την μόλυνση malware. Συγκεκριμένα, εάν το malware προκαλεί σκόπιμα οικονομικές επιβαρύνσεις στους μολυσμένους χρήστες. Ένας κερδοφόρος τρόπος για τους επιτιθέμενους είναι η κρυφή εγγραφή σε υπηρεσίες πρόσθετου τέλους που ελέγχουν οι ίδιοι, όπως π.χ. Η αποστολή μηνυμάτων SMS. Στο Android, υπάρχει η μέθοδος

28 Command and control *servers*

29 SH1: 663e8eb52c7b4a14e2873b1551748587018661b3

30 Κεφάλαιο 3.3.2 AnserverBot

sendTextMessage που μετά την χορήγηση του αντίστοιχου δικαιώματος επιτρέπει την αποστολή SMS μηνυμάτων στο παρασκήνιο εν αγνοία του χρήστη. Αυτός ο τύπος επίθεσης με στόχο τους χρήστες, συναντάται κυρίως στη Ρωσία, τις Ηνωμένες Πολιτείες και την Κίνα. Το πρώτο Android malware, το FakePlayer έστελνε SMS το μήνυμα "798657" σε πολλαπλούς αριθμούς υψηλής χρέωσης στη Ρωσία. Στις ΗΠΑ Το GGTracker εγγράφει αυτόματα τον χρήστη σε χρεωστικές υπηρεσίες, χωρίς τη γνώση του. Αντίστοιχα το zSone στέλνει μηνύματα SMS σε αριθμούς υψηλής χρέωσης στην Κίνα χωρίς τη συγκατάθεση του χρήστη. Συνολικά, εντοπίζονται **55 δείγματα (4.4%)** malware από 7 διαφορετικές οικογένειες (με ένδειξη ‡ στον πίνακα 3) που στέλνουν SMS μηνύματα σε αριθμούς υψηλής χρέωσης. Επιπλέον, μερικά malware επιλέγουν να μην κρατάνε τους τηλεφωνικούς αριθμούς ενσωματωμένους στον κώδικα τους, αντιθέτως με την χρήση ενός ευέλικτου remote control, οι αριθμοί αυτοί ενημερώνονται δυναμικά κατά το χρόνο εκτέλεσης. Υπάρχουν 13 τέτοιες οικογένειες malware (με ετικέτα † στον Πίνακα 3). Προφανώς, αυτές οι οικογένειες malware είναι περισσότερο κρυφές από τις προηγούμενες, επειδή ο αριθμός προορισμού δεν γίνεται γνωστός με μία απλή ανάλυση της μολυσμένης εφαρμογής. Επίσης παρατηρείται ότι για την αυτόματη εγγραφή σε υπηρεσίες πρόσθετου τέλους, αυτές οι οικογένειες malware πρέπει να απαντήσουν σε ορισμένα SMS μηνύματα. Αυτό μπορεί να οφείλεται στην πολιτική δεύτερης επιβεβαίωσης που απαιτείται σε ορισμένες χώρες, όπως όπως η Κίνα. Συγκεκριμένα, για την εγγραφή σε μία υπηρεσία υψηλής χρέωσης, ο χρήστης πρέπει να απαντήσει σε ένα SMS μήνυμα που αποστέλλεται από τον πάροχο της υπηρεσίας επιβεβαιώνοντας ότι έλαβε γνώση για να ολοκληρωθεί η διαδικασία εγγραφής ή/και να ενεργοποιηθεί η υπηρεσία. Για να αποφύγει αυτή την ενημέρωση προς τους χρήστες, το malware θα φροντίσει να απαντήσει σε αυτά τα μηνύματα επιβεβαιώνοντας από μόνο του. Ένα παράδειγμα αποτελεί το RogueSPPush, το οποίο απαντάει αυτόματα "Y" σε όλα τα εισερχόμενα μηνύματα επιβεβαίωσης στο παρασκήνιο. Ενώ το GGTracker απαντούσε "Yes" στα μηνύματα που προέρχονται από τον αριθμό 99735, ώστε να ενεργοποιηθεί η υπηρεσία. Ομοίως, για να αποτρέψουν την ενημέρωση των χρηστών για την επακόλουθη χρέωση που σχετίζεται με τα μηνύματα,τα malware φιλτράρουν αυτά τα ενημερωτικά μηνύματα SMS ώστε να μην τα βλέπει ο χρήστης. Αυτή η συμπεριφορά υιοθετήθηκε από μία σειρά από malware, συμπεριλαμβανομένων των zSone, RogueSPPush και GGTracker. Εκτός από την εγγραφή σε υπηρεσίες υψηλού κόστους, κάποια malware αξιοποιούν την ίδια λειτουργικότητα για αποστολή μηνυμάτων SMS και σε άλλους αριθμούς τηλεφώνου. Αν και λιγότερο σοβαρή απειλή από τις προηγούμενες, έχει επίσης αποτέλεσμα ορισμένες οικονομικές επιβαρύνσεις, ιδίως όταν ο χρήστης δεν έχει στο συμβόλαιο του απεριόριστα μηνύματα. Τέτοιο παράδειγμα αποτελεί το maleare DogWars που στέλνει SMS μηνύματα σε όλες τις επαφές στο τηλέφωνο χωρίς την ενημέρωση του χρήστη. Τέλος άλλα malware κάνουν τηλεφωνικές κλήσεις στο παρασκήνιο με διαχείριση remote control, ο αριθμός κλίσης μπορεί να παρέχεται από έναν απομακρυσμένο C&C server (πχ Geinimi).

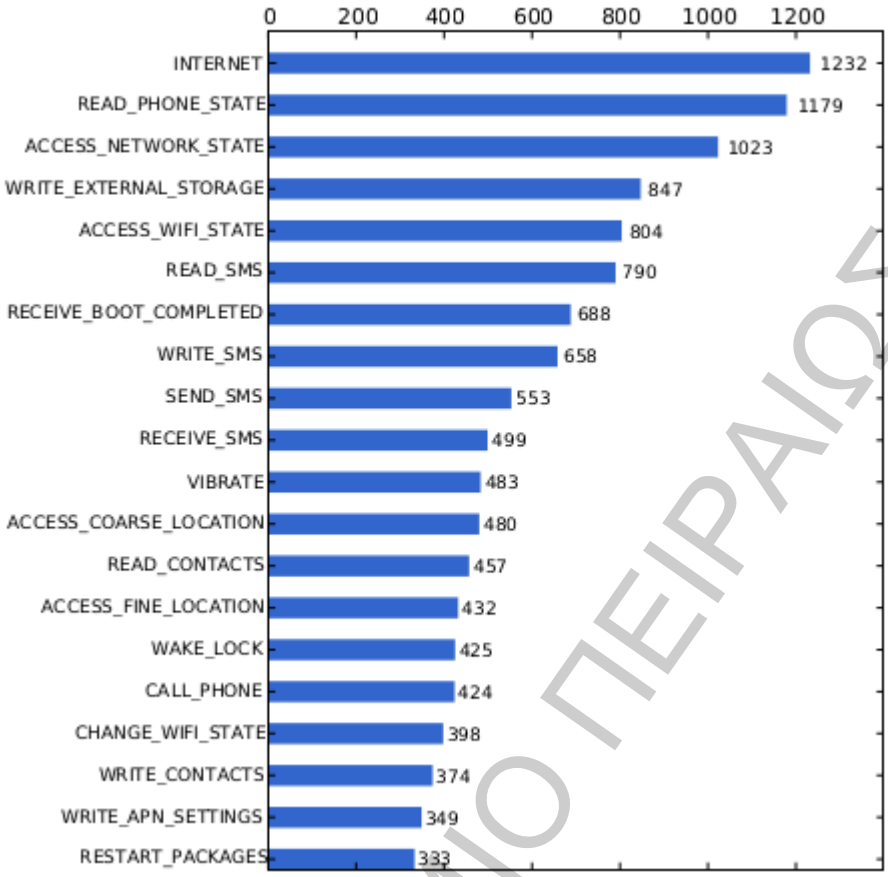
3.2.3.4 Συλλογή Πληροφοριών

Τα malware σαν επιπρόσθετο στόχο έχουν την συγκομιδή διάφορων πληροφοριών από τα μολυσμένα τηλέφωνα, οι πληροφορίες που υποκλέπτουν είναι σχετικές με τον χρήστη όπως τα SMS μηνύματα, οι αριθμοί τηλεφώνου, καθώς οι λογαριασμοί των χρηστών (email, Facebook κτλ). Ειδικότερα, υπάρχουν 13 οικογένειες κακόβουλου λογισμικού (138 δείγματα) στο σύνολο δεδομένων της έρευνας των Yajin Zhou και Xuxian Jiang [22] που συλλέγουν τα μηνύματα SMS, 15 οικογένειες (563 δείγματα) να συγκεντρώνουν αριθμούς τηλεφώνου και 3 οικογένειες

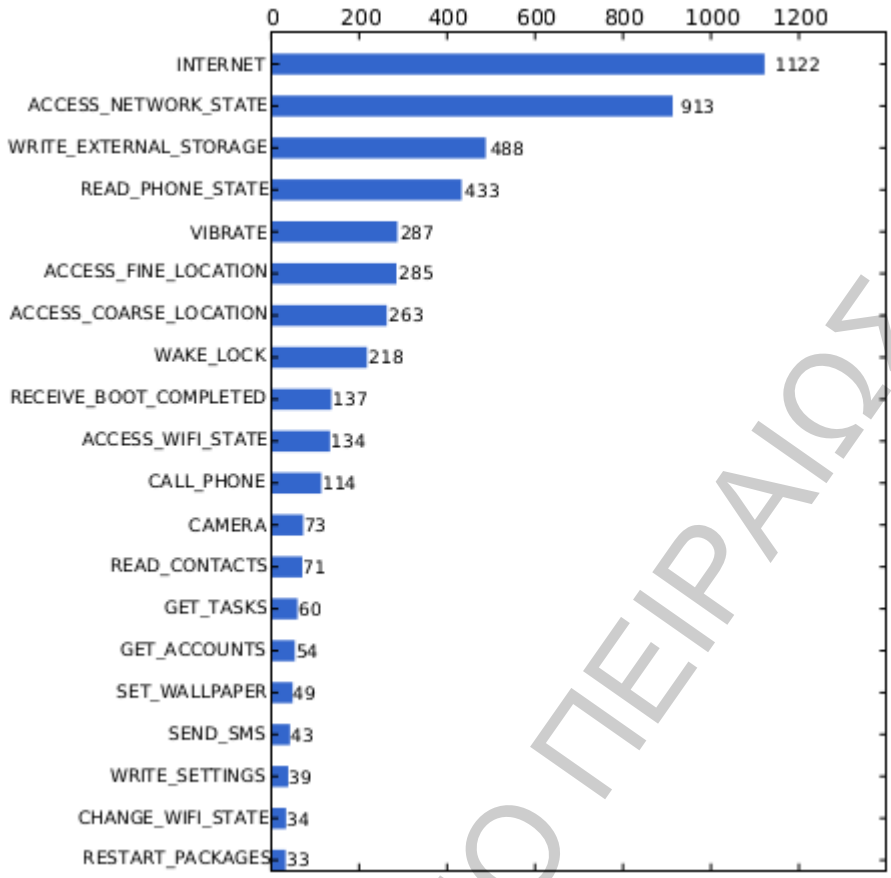
(43 δείγματα) αποκτούν και αποστέλλουν πληροφορίες σχετικά με τους λογαριασμούς χρηστών. Για παράδειγμα, το malware SndApps συλλέγει τις διευθύνσεις ηλεκτρονικού ταχυδρομείου των χρηστών και τις στέλνει σε έναν απομακρυσμένο server. Το FakeNetflix συγκεντρώνει τους λογαριασμούς Netflix των χρηστών καθώς και και τους κωδικούς πρόσβασης, παρέχοντας ένα ψεύτικο αλλά φαινομενικά πανομοιότυπο Netflix UI. Η συλλογή των SMS μηνυμάτων των χρηστών είναι μια πολύ ύποπτη συμπεριφορά. Πολλές φορές σε μηνύματα SMS περιλαμβάνονται επίσημα διαπιστευτήρια των χρηστών για κάποια υπηρεσία. Για παράδειγμα, τόσο Zitmo (η έκδοση του Zeus για Android) αλλά και το Spitmo (η έκδοση του SpyEpy στο Android) προσπαθούν να υποκλέψουν SMS μηνύματα επαλήθευσης εγγραφής σε μία υπηρεσία και στη συνέχεια τα αποστέλλουν σε έναν απομακρυσμένο server. Αν αυτή η διενέργεια είναι επιτυχής, ο επιτιθέμενος μπορεί να χρησιμοποιήσει το περιεχόμενο για να δημιουργήσει ψευδείς συναλλαγές για λογαριασμό των μολυσμένων χρηστών.

3.2.4 Δικαιώματα χρήσης

Οι δυνατότητές των εφαρμογών που δεν εφαρμόζεται κάποια μέθοδο privilege escalation για να πάρουν κάποιο root δικαίωμα, είναι αυστηρά περιορισμένες από τα δικαιώματα που τους χορηγεί ο χρήστης. Ως εκ τούτου, είναι ενδιαφέρον η σύγκριση των δικαιωμάτων που αιτούνται οι malware εφαρμογές με τις αντίστοιχες που αιτούνται οι αυθεντικές/νόμιμες εφαρμογές. Για το σκοπό αυτό, επιλέχθηκαν τυχαία 1.260 δημοφιλής δωρεάν εφαρμογές από το επίσημο Android Market (Οκτώβριος 2011). Τα αποτελέσματα παρουσιάζονται στην Εικόνα 15. Με βάση τη σύγκριση, τα δικαιώματα INTERNET, READ_PHONE_STATE, ACCESS_NETWORK_STATE και WRITE_EXTERNAL_STORAGE ζητούνται ευρέως τόσο στα malware όσο και στις νόμιμες εφαρμογές. Τα πρώτα δύο τυπικά απαιτούνται για να επιτρέπεται η σωστή λειτουργία των βιβλιοθηκών διαφήμισης. Αλλά οι malware εφαρμογές εμφανίζουν σαφώς την τάση να ζητούν πιο συχνά δικαιώματα που σχετίζονται με SMS, όπως READ_SMS, WRITE_SMS, RECEIVE_SMS και SEND_SMS. Συγκεκριμένα, στο σύνολο δεδομένων υπάρχουν 790 δείγματα (62,7%) που αιτούνται το δικαίωμα READ_SMS, ενώ μόνο 33 (2,6%) από αυτά τα δείγματα είναι καλοήθεις εφαρμογές. Αυτά τα αποτελέσματα συμφωνούν με το γεγονός ότι 28 οικογένειες malware στην ίδια έρευνα (ή 45,3% των δειγμάτων) έχουν κακόβουλη λειτουργικότητα που σχετίζεται με SMS. Επίσης, παρατηρούνται 688 δείγματα malware να αιτούνται το δικαίωμα RECEIVE_BOOT_COMPLETED. Αυτός ο αριθμός είναι πέντε φορές μεγαλύτερος από ότι στις καλοήθεις εφαρμογές (137 δείγματα). Αυτό θα μπορούσε να οφείλεται στο γεγονός ότι τα malware είναι πιο πιθανό να αποσκοπούν στην εκτέλεση υπηρεσιών στο παρασκήνιο χωρίς την παρέμβαση του χρήστη. Επίσης υπάρχουν 398 δείγματα malware που ζητούν το δικαίωμα CHANGE_WIFI_STATE σε πολύ υψηλότερο ποσοστό από τις νόμιμες εφαρμογές (34 δείγματα). Αυτό είναι κυρίως επειδή το root exploit Exploid απαιτεί ορισμένα system-wide γεγονότα, όπως την αλλαγή της κατάστασης του WIFI, η οποία σχετίζεται με αυτό το δικαίωμα. Τέλος, παρατηρείται ότι οι κακόβουλες εφαρμογές τείνουν να αιτούνται περισσότερα δικαιώματα από τις αντίστοιχες καλοήθεις. Ο μέσος όρος των δικαιωμάτων που αιτούνται οι mlware εφαρμογές είναι 11 ενώ ο μέσος όρων των νόμιμων εφαρμογών είναι 4. Μεταξύ των 20 πιο δημοφιλών δικαιωμάτων, 9 από αυτά αιτούνται από κακόβουλες εφαρμογές κατά μέσο όρο, ενώ 3 από αυτά αιτούνται από καλοήθεις εφαρμογές.



Εικόνα 15, Top 20 πιο δημοφιλή δικαιώματα σε ένα πλήθος 1260 Malware δειγμάτων



Εικόνα 16, Τα Top 20 πιο δημοφιλή δικαιώματα που αιτούνται 1260 καλοήθεις εφαρμογές

3.3 Εξέλιξη malware

Εικόνα 17, Το Android κόλλησε ιό

Από το καλοκαίρι του 2011 έχει παρατηρηθεί ραγδαία αύξηση των Android malware. Στην ενότητα αυτή, γίνεται μία παρουσίαση κάποιων αντιπροσωπευτικών δειγμάτων και μία πιο εμπεριστατωμένη ανάλυση της εξέλιξης τους. Συγκεκριμένα, του DroidKungFu (συμπεριλαμβανομένων των παραλλαγών του) και του AnserverBot λόγω της αντανάκλασης του στην τρέχουσα τάση του ανάπτυξη Android malware.

3.3.1 DroidKungFu

Η πρώτη έκδοση του DroidKungFu (ή DroidKungFu1) malware ανιχνεύθηκε τον Ιούνιο 2011 [29]. Θεωρήθηκε ένα από τα πιο εξελιγμένα Android malware εκείνη τη στιγμή. Λίγο αργότερα ανιχνεύθηκε και η δεύτερη έκδοση, το DroidKungFu2 και η τρίτη έκδοση το DroidKungFu3 τον Ιούλιο και τον Αύγουστο αντίστοιχα, ενώ η τέταρτη έκδοση DroidKungFu4 (LeNa) ανιχνεύθηκε τον Οκτωβρίου 2011 [30]. Λίγο μετά εμφανίστηκε η πέμπτη έκδοση DroidKungFuSapp, η οποία ήταν αρκετό καιρό μη ανιχνεύσιμη από τα λογισμικά anti-virus. Εν τω μεταξύ, υπάρχει και μια άλλη παραλλαγή που ονομάζεται DroidKungFuUpdate [31], που χρησιμοποιεί Update attack. Συνολικά έχουν εντοπιστεί 473 DroidKungFu δείγματα malware. Η εμφάνιση αυτών των παραλλαγών DroidKungFu σαφώς δείχνει την τρέχουσα τάση και την ταχεία ανάπτυξη του Android malware. Ακολουθεί περαιτέρω ανάλυση των διάφορων πτυχών του DroidKungFu malware.

3.3.1.1 Root Exploits

Τέσσερις από τις έξι παραλλαγές του DroidKungFu περιέχουν κρυπτογραφημένα root exploits. Τα κρυπτογραφημένα αυτά αρχεία βρίσκονται στον κατάλογο "assets", τα οποία μοιάζουν με κανονικά αρχεία δεδομένων. Στο DroidKungFu παρατηρήθηκε για πρώτη φορά σε Android

malware να συμπεριλαμβάνονται κρυπτογραφημένα root exploits. Η χρήση της κρυπτογράφησης χρησιμοποιείται από τα malware για αποφύγουν την ανίχνευση. Διαφορετικές παραλλαγές του DroidKungFu χρησιμοποιούν διαφορετικά κλειδιά κρυπτογράφησης για να προστατεύουν τον εαυτό τους καλύτερα. Για παράδειγμα, το κλειδί που χρησιμοποιήθηκε στο DroidKungFu1 είναι “Fuck_sExy-all! Pw”[31], το οποίο άλλαξε σε “Stak_yExy-ELT! Pw” στο DroidKungFu4. Επίσης στο DroidKungFu1 το όνομα του αρχείου του κρυπτογραφημένου root exploit είναι "ratc" – το αρκτικόλεξο του RageAgainstTheCage. Ενώ στα DroidKungFu2 και DroidKungFu3, αυτό το αρχείο με το ίδιο root exploit έχει έχει αλλάξει όνομα σε "myicon" και προσποιείται ότι είναι ένα αρχείο εικόνας.

3.3.1.2 C&C Servers

Όλες οι εκδόσεις του DroidKungFu έχουν ωφέλιμο φορτίο που επικοινωνεί με απομακρυσμένους C&C servers (διακομιστές) και λαμβάνουν εντολές από αυτούς. Το malware αλλάζει συνεχώς τους τρόπους που αποθηκεύει τις διευθύνσεις των C&C servers. Για παράδειγμα, στον DroidKungFu1, η διεύθυνση url του C&C server αποθηκεύεται σε μορφή απλού κειμένου σε ένα αρχείο Java. Ενώ στο DroidKungFu2, η C&C διεύθυνση του διακομιστή έχει μετακινηθεί σε μία τρίτη εφαρμογή σε κανονική μορφή κειμένου. Επίσης, ο αριθμός των απομακρυσμένων διακοσμητών C&C αυξήθηκε από ένας σε τρεις. Σε αντίθεση το DroidKungFu3 κρυπτογραφεί την διεύθυνση του C&C Server σε ένα αρχείο Java. Στο DroidKungFu4 κρύβει τη διεύθυνση C&C σε μία τρίτη εφαρμογή όπως στο DroidKungFu2 αλλά σε κρυπτογραφημένη μορφή. Τέλος στο DroidKungFuSapp χρησιμοποιεί ένα νέο διακομιστή C&C και ένα διαφορετικό δικό του σύστημα κρυπτογράφησης.

3.3.1.3 Κρυφό ωφέλιμο φορτίο

Το DroidKungFu περιέχει επίσης μία ενσωματωμένη εφαρμογή, η οποία θα εγκατασταθεί κρυφά όταν το root exploit εκτελεστεί επιτυχώς. Κοινώς, η ενσωματωμένη εφαρμογή εγκαθίσταται εν αγνοία του χρήστη. Μια εξέταση του κώδικα της ενσωματωμένης εφαρμογής δείχνει ότι περιέχει σχεδόν πανομοιότυπο ωφέλιμο φορτίο με αυτό που έχει το ίδιο το DroidKungFu. Η εγκατάσταση αυτή εξασφαλίζει όμως ότι ακόμη στην περίπτωση που η αρχική μολυσμένη εφαρμογή αφαιρεθεί, θα συνεχίσει να είναι μολυσμένη η συσκευή και υφίσταται η λειτουργικότητα του malware. Συγκεκριμένα, στο DroidKungFu1, το ενσωματωμένο app εμφανίζει ένα ψεύτικο Google Search εικονίδιο ενώ το DroidKungFu2, το ενσωματωμένο app είναι κρυπτογραφημένο και δεν εμφανίζεται κανένα εικονίδιο στο τηλέφωνο.

3.3.1.4 Σύγχυση κώδικα (Obfuscation), Java Native Interface (JNI)

Όπως αναφέρθηκε εν συντομία νωρίτερα, το DroidKungFu κάνει έντονη χρήση κρυπτογράφησης για να κρύψει ύπαρξή του, ενώ το Geinimi κρυπτογραφεί τις σημβολοσειρές που χρησιμοποιεί ώστε να είναι δύσκολο να αναλυθεί. Το DroidKungFu

31 http://about-threats.trendmicro.com/malware.aspx?language=apac&name=ANDROIDOS_DROIDKUNGFU.B

αντί να κρυπτογραφεί μόνο τις σημβολοσειρές και τους C&C servers κρυπτογραφεί και τα ωφέλιμα φορτία και την ενσωματωμένη εφαρμογή. Επιπλέον, αλλάζει γρήγορα τα κλειδιά της κρυπτογράφησης και υλοποιεί τεχνικές obfuscation στα ονόματα των κλάσεων, στο κακόβουλα ωφέλιμο φορτίο και χρησιμοποιεί διασυνδέσεις JNI exploit για να αυξήσει τη δυσκολία της ανάλυσης και την ανίχνευση του. Για παράδειγμα, το DroidKungFu2 αλλά και το DroidKungFu4 χρησιμοποιούν ένα εγγενές πρόγραμμα (μέσω JNI) για αμφίδρομη επικοινωνία με τους απομακρυσμένους διακομιστές. Η τελευταία έκδοση, δηλαδή, το DroidKungFuUpdate, εφαρμόζει update attack. Οι μηχανισμοί απόκρυψης του ήταν τόσο επιτυχημένοι που κατάφερε να δημοσιευθεί στο επίσημο το Android Market με τους χρήστες να την κατεβάζουνε σε μεγάλο βαθμό αντικατοπτρίζοντας την τάση εξέλιξης του Android maware να εντοπίζεται όλο και πιο δύσκολα.

3.3.2 AnserverBot

Το AnserverBot ανακαλύφθηκε το Σεπτέμβριο του 2011. Αυτό το malware ενσωματωνόταν σε κάποιες νόμιμες εφαρμογές και δημοσιεύτηκε σε κάποια εναλλακτικά Android market στην Κίνα. Το συγκεκριμένο malware θεωρείται ένα από τα πιο εξελιγμένα Android malware, καθώς αξιοποιούσε αρκετά εξελιγμένα τεχνικά exploit για να αποφύγει τον εντοπισμό και την ανάλυση, τα οποία ήταν Zero day (δεν είχαν εμφανιστεί στο παρελθόν). Η πλήρης ανάλυση του διήρκεσε πάνω από μία εβδομάδα για να ολοκληρωθεί [32]. Πιστεύεται ότι αυτό το malware αποτελεί εξέλιξη του προγενέστερου malware BaseBridge. Στο υποκεφάλαιο αυτό υπογραμμίζονται οι βασικές τεχνικές που χρησιμοποιούνται από AnserverBot.

3.3.2.1 Αντι-Ανάλυση

Αν και το AnserverBot μολύνει υπάρχουσες εφαρμογές με ανασυσκευασία, έχει ως στόχο να προστατεύσει τον εαυτό του από την ανίχνευση του εάν η ανασκευασμένη εφαρμογή έχει αλλοιωθεί ή όχι. Πιο συγκεκριμένα, όταν εκτελείται, θα ελέγξει την υπογραφή ή τη σύνοψη ακεραιότητας της εφαρμογής που είναι υπό ανασυσκευασία πριν εκτελέσει το ωφέλιμο φορτίο του. Με αυτό το μηχανισμό είναι σε θέση είναι να ματαιώσει τις πιθανές προσπάθειες ανάλυσης ή/και αντίστροφης μηχανικής. Επιπλέον, το AnserverBot χρησιμοποιεί obfuscate τεχνικές στις εσωτερικές του κλάσεις, μεθόδους και πεδία ώστε να καταστούν ανθρωπίνως δυσανάγνωστα. Επίσης, διαχωρίζει σκόπιμα το κύριο φορτίο σε τρεις συνδεόμενες εφαρμογές: στην εφαρμογή υποδοχής και σε άλλες δύο ενσωματωμένες εφαρμογές. Οι δύο ενσωματωμένες εφαρμογές μοιράζονται το ίδιο όνομα com.sec.android.touchScreen.server αλλά έχουν διαφορετική λειτουργικότητα. Η μία από τις δύο εγκαθίσταται μέσω update attack, ενώ η άλλη θα φορτωθεί δυναμικά χωρίς να εγκατασταθεί στην πραγματικότητα (όπως το Plankton). Η τμηματοποίηση και ο συντονισμός της λειτουργικότητας, καθώς και η έντονη χρήση obfuscation, κάνει την ανάλυση πολύ δύσκολη. Ειδικότερα, οι μηχανισμοί για ανάκτηση και να εκτέλεση απομακρυσμένου κώδικα που δεν ήταν διαθέσιμοι πριν το BaseBridge malware. Εκμεταλλεύεται τη δυνατότητα φόρτωσης κλάσεων της εικονικής μηχανής Dalvik για να φορτώσει και να εκτελέσει το κακόβουλο ωφέλιμο φορτίο κατά το χρόνο εκτέλεσης. Χρησιμοποιώντας αυτή τη δυναμική συμπεριφορά φόρτωσης, το AnserverBot μπορεί να προστατεύσει τον εαυτό του σε μεγάλο βαθμό από το να ανιχνευθεί από τα υπάρχοντα λογισμικά anti-virus (Πίνακας 4). Επιπλέον, με τέτοιο δυναμισμό να υφίσταται, οι συγγραφείς

κακόβουλου λογισμικού μπορούν να αναβαθμίσουν άμεσα τα ωφέλιμα φορτία τους ενώ παράλληλα κρατούν ακέραιο το πλήθος της τρέχουσας βάσης μολυσμένων συσκευών.

3.3.2.2 Ανίχνευση Λογισμικού Ασφαλείας

Ένα άλλο χαρακτηριστικό αυτοπροστασίας που χρησιμοποιείται στο AnserverBot είναι ότι μπορεί να ανιχνεύσει την παρουσία ορισμένων anti-virus λογισμικών. Ειδικότερα, περιλαμβάνει κρυπτογραφημένα ονόματα τριών mobile anti-virus, τα com.qihoo360.mobilesafe, com.tencent.qqpimsecure και com.lbe.security και προσπαθεί να τα ταιριάξει με τις εγκατεστημένες εφαρμογές στο τηλέφωνο. Στην περίπτωση που κάποιο από τα τρία anti-virus ανιχνευτεί, το AnserverBot το σταματάει καλώντας τη μέθοδο restartPackage και εμφανίζει ένα παράθυρο διαλόγου ενημερώνοντας το χρήστη ότι η συγκεκριμένη εφαρμογή έχει σταματήσει απροσδόκητα.

3.3.2.3 C&C Servers

Μια ενδιαφέρουσα πτυχή του AnserverBot είναι οι C&C servers του. Ειδικότερα, υποστηρίζει δύο τύπους C&C servers. Ο πρώτος είναι παρόμοιος με τον παραδοσιακό C&C server τύπο (να λαμβάνει εντολές). Ο δεύτερος αντιθέτως χρησιμοποιεί την αναβάθμιση του ωφέλιμου φορτίου του ή / και για να ενημερώσει για τη νέα διεύθυνση του πρώτου τύπου C&C server. Απροσδόκητα, ο δεύτερος τύπος βασίζεται σε κρυπτογραφημένο περιεχόμενο, το οποίο διατηρείται σε δημοφιλείς παρόχους υπηρεσιών blog (πχ, Sina και Baidu). Το AnserverBot συνδέεται σε αυτό το δημόσιο blog και ενημερώνεται για τον (κρυπτογραφημένο) τρέχων C&C server και λαμβάνει το νέο (κρυπτογραφημένο) ωφέλιμο φορτίο. Αυτή η λειτουργία μπορεί να διασφαλίσει ότι ακόμη και αν ο πρώτος τύπος C&C server δεν είναι συνδεδεμένος, ένας νέος C&C server μπορεί ακόμα να συνδεθεί με το malware, μέσω αυτών των δημόσιων blog.

3.3.3 Rage Against the Cage (RAtC)

Το Rage Against the Cage (RatC) [33] εκμεταλλεύεται το γεγονός ότι το Android Debug Bridge daemon (adb)[32] εκκινεί την συσκευή σαν root και στην συνέχεια καλεί μία διεργασία setuid, που περιορίζει τα δικαιώματα σαν απλού χρήστη. Το ADB daemon εκτελείται και στην συσκευή ώστε να επιτρέπει στους προγραμματιστές να επικοινωνούν με τις εφαρμογές τους κατά την διάρκεια της εκτέλεσης τους στην συσκευή. Σε κάθε σύστημα Linux υπάρχει ένα όριο των πόρων το RLIMIT_NPROC που καθορίζει το μέγιστο αριθμό των ταυτόχρονων διαδικασιών που επιτρέπονται από το σύστημα Στο Android, το RAtC ελέγχει το όριο αυτό και γεννά και διαδικασίες που δεν κάνουν τίποτα, μέχρι έχει φτάσει στο όριο RLIMIT_NPROC του Android. (Επίθεση fork bomb). Σε εκείνο το χρονικό σημείο το RAtC τερματίζει το adb στην συσκευή. Αλλά υπάρχει η διεργασία που ξεκινά και φροντίζει ώστε το adb να εκτελείται πάντα. Και επανεκκινεί το adb. Όμως σε αυτή την φάση, υπάρχει μία συνθήκη που πρέπει να

32 Κεφάλαιο: 6.6.1.1 Android Debug Bridge (adb)

ξεπεραστεί από την στιγμή που πρέπει να εκκινηθεί μία γονική διεργασία ενώ παράλληλα το RLIMIT_NPROC έχει φτάσει σχεδόν στο όριο. Το Adb επανεκκίνηται όπως πάντα ως root, αφού κάνει τους απαραίτητους ελέγχους και κρίνει ότι δεν χρειάζεται να είναι root ως συνήθως. (Γεγονός που ισχύει όταν το Android δεν είναι σε sandbox περιβάλλον αλλά σε κανονική συσκευή με το debug ενεργοποιημένο). Για να περιορίσει τα δικαιώματα του το Adb ξεκινάει την διεργασία setuid αλλά το fork bomb που έχει υλοποιηθεί από το RAtC έχει καλύψει τον αριθμό των διεργασιών που μπορούν να εκτελεστούν. Με αποτέλεσμα να μην μπορεί να εκτελεστεί το setuid. Αφήνοντας έτσι το adb να εκτελείται ως root. Μέσω το adb και ο χρήστης πλέον είναι root.

3.4 Ανίχνευση του malware

Η ταχεία ανάπτυξη και εξέλιξη των πρόσφατων Android malware αποτελούν σημαντικές προκλήσεις για τον εντοπισμό τους. Σε αυτή την ενότητα, παρουσιάζεται μία έρευνα [22] για την αποτελεσματικότητα των υφιστάμενων mobile anti-virus. Για το σκοπό αυτό, επιλέχθηκαν τέσσερα αντιπροσωπευτικά mobile anti-virus, συγκεκριμένα,το AVG Antivirus v2.9 (ή AVG) [34], Lookout Security & Antivirus v6.9 [35], το Norton Mobile Security Lite v2.5.0.379 (Norton) [36] και το TrendMicro Mobile Security Personal Edition v2.0.0.1294 (TrendMicro) [37] από το επίσημο Android Market την πρώτη εβδομάδα του Νοεμβρίου του 2011. Για την μέτρηση εγκαταστάθηκε το καθένα από αυτά σε ένα ξεχωριστό τηλέφωνο Nexus One με Android έκδοση 2.3.7. Τα λογισμικά antivirus είχαν ενημερωμένη την βάση δεδομένων τους. Κατά την διάρκεια της δοκιμής υφίστανται οι προεπιλεγμένες ρυθμίσεις και ο μηχανισμός προστασίας σε πραγματικό χρόνο. Το πείραμα ήταν ως εξής: ένα script πρόγραμμα, εγκαθιστά σε όλες συσκευές με την σειρά κάποιο από τα δείγματα ιού, περιμένει 30 δευτερόλεπτα και μετά εγκαθιστά και άλλο δείγμα. Αν εντοπιστεί ο ιός θα εμφανιστεί ένα αναδυόμενο παράθυρο και αυτή η ενέργεια θα καταγραφεί από το script πρόγραμμα. Μετά την πρώτη επανάληψη δίνεται η δυνατότητα εντοπισμού σε ένα δεύτερο γύρο μέσω σάρωσης αυτών των δειγμάτων που δεν ανιχνεύθηκαν στον πρώτο γύρο. Τα αποτελέσματα της σάρωσης φαίνονται στον Πίνακα 4. Όπου οι δύο πρώτες στήλες λίστα αναφέρουν την οικογένεια του malware και τον αριθμό των δειγμάτων σε αυτήν την οικογένεια. Οι υπόλοιπες στήλες δείχνουν τον αριθμό των δειγμάτων, καθώς και το ποσοστό που ανιχνεύτηκε από το αντίστοιχο λογισμικό ασφαλείας - antivirus. Στο τέλος του πίνακα φαίνεται ο αριθμός των δειγμάτων που εντοπίστηκαν από κάθε λογισμικό anti-virus και το αντίστοιχο ποσοστό ανίχνευσης του.

Τα αποτελέσματα δεν είναι ενθαρρυντικά:

Το Lookout ανίχνευσε 1003 δείγματα malware σε 39 οικογένειες.

Το TrendMicro εντόπισε 966 δείγματα σε 42 οικογένειες.

Το AVG εντόπισε 689 δείγματα σε 32 οικογένειες.

Και το Norton εντόπισε 254 δείγματα σε 36 οικογένειες.

Προφανώς, αυτά τα λογισμικά ασφαλείας έχουν διαφορετικές προσεγγίσεις στο σχεδιασμό και την εφαρμογή τους, οι οποίες οδηγούν σε διαφορετική αναλογία ανίχνευσης ακόμη και για malware που ανήκουν στην ίδια οικογένεια. Για παράδειγμα, το AVG ανιχνεύει όλα τα δείγματα, της οικογένειας ADRD, ενώ το Lookout ανιχνεύει μόνο το 59% από αυτή την

οικογένεια. Επίσης, το Lookout ανιχνεύει τα περισσότερα DroidKungFu3 δείγματα και όλα τα DroidKungFu4 δείγματα, ενώ το AVG δεν μπορεί να ανιχνεύσει κανένα DroidKungFu3 (0,0%) ή και λίγα από το DroidKungFu4 (4,1%).

Υπάρχουν κάποιες οικογένειες malware που δεν εντοπίζονται καθόλου με αυτά τα τέσσερα antivirus. Παραδείγματα αποτελούν BeanBot, CoinPirate, DroidCoupon, DroidKungFuSapp, NickyBot και RogueLemon. Ένας λόγος είναι ότι είναι σχετικά νέες οικογένειες (Ανακαλύφθηκαν Αύγουστο - Οκτώβριο 2011). Συνεπώς, οι εταιρείες που αναπτύσσουν mobile anti-virus να μην είχαν την ευκαιρία να αποκτήσουν ένα αντίγραφο αυτών των δειγμάτων ή κάποιες από τις υπογραφές τους. Από άλλη οπτική, πλέον η παραδοσιακή προσέγγιση του να υπάρχει μια βάση δεδομένων με τις υπογραφές των δειγμάτων malware δεν είναι λειτουργική. Διότι σε περιπτώσεις που ένα δείγμα δεν είναι διαθέσιμο, είναι πολύ πιθανόν να μην θα είναι ανιχνευθεί.

Συμπεράσματα 3ου κεφαλαίου

Σε αυτό το κεφάλαιο παρουσιάστηκε μία κατηγοριοποίηση των υπαρχόντων Android malware με σκοπό την κατανόηση του τρόπου λειτουργίας τους. Τα αποτελέσματά της έρευνας αυτής δείχνουν ότι:
(1) Το 86,0% των δειγμάτων αποτελεί προϊόν ανασυσκευασίας κάποιας νόμιμης εφαρμογής με προσθήκη κάποιου κακόβουλου ωφέλιμου φορτίου.

(2) Το 36,7% περιέχει exploits τύπου κλιμάκωσης προνομιών σε επίπεδο πλατφόρμας.

(3) Το 93,0% εμφανίζει ιδιότητες bot.

Μια περαιτέρω ανάλυση σε βάθος της εξέλιξης του κακόβουλου λογισμικού στο Android δείχνει την ταχεία αύξηση και την ανάπτυξη της πολυπλοκότητας, που θέτουν σημαντικές προκλήσεις για τον εντοπισμό τους. Δυστυχώς, οι αξιολογήσεις τεσσάρων υφιστάμενων Android anti-virus δείχνει ότι στην καλύτερη περίπτωση ανιχνεύεται το 79,6% από αυτούς, ενώ στην χειρότερη περίπτωση ανιχνεύεται μόνο το 20,2%. Τα αποτελέσματα αυτά αναδεικνύουν την ανάγκη για περαιτέρω ανάπτυξη στον τομέα της αντιμετώπισης Android malware.

Κεφάλαιο 4 Ανάλυση κακόβουλου λογισμικού

4.1 Εισαγωγή στην ανάλυση malware

Τα malware εξελίσσονται με ταχύ τρόπο και τα αντίμετρα πού λαμβάνονται κατέστησαν ανεπαρκή επειδή πλέον τα malware χρησιμοποιούν νέες τεχνικές όπως διαφορετικές υπογραφές και ενθυλάκωση καθιστώντας την ανίχνευση τους όλο και πιο δύσκολη. Τα προϊόντα Anti-Virus καθημερινά εκδίδουν ενημερώσεις που ανιχνεύουν ένα σημαντικό αριθμό από τις επιθέσεις malware, όμως υπάρχει και ένας μη αμελητέος αριθμός από malware που διαφεύγει του εντοπισμού. Είναι σημαντικό για έναν αναλυτή να μπορεί να εξετάσει αυτά τα malware που αλλάζουν τις τιμές registry που παραποιούν δεδομένα ή κατεβάζουν κακόβουλα ωφέλιμα φορτία με ασυνήθιστη συμπεριφορά. Ο αναλυτής πρέπει να μπορεί να αναλύσει το κακόβουλο λογισμικό στο συγκεκριμένο λειτουργικό σύστημα και να μελετήσει τις μεταβλητές περιβάλλοντος και την δραστηριότητα που ασκείται από το κακόβουλο λογισμικό. ([38], [39])

4.2 Το πλαίσιο της ανάλυσης malware

Η απειλή του κακόβουλου λογισμικού θεωρείται η μεγαλύτερη απειλή για την ασφάλεια στο Διαδίκτυο αυτές τις μέρες. Παλαιότερα, οι ιοί ήταν η μόνη μορφή κακόβουλου λογισμικού. Ωστόσο, στις μέρες μας, η απειλή έχει μεγαλώσει και περιλαμβάνει ένα ευρύ φάσμα από ιδιαίτερα εξελιγμένες εφαρμογές όπως worms, Trojans, πράκτορες DDoS, ελεγχόμενα IRC bots, spywares, rootkits και πολλά άλλα. Οι φορείς μόλυνσης έχουν επίσης αλλάξει δραστικά και αναπτύσσονται ως κακόβουλοι παράγοντες που πλέον χρησιμοποιούν μηχανισμούς, όπως email harvesting, browser exploits, αδυναμίες του λειτουργικού συστήματος και P2P δίκτυα όπως και πολλές καινούργιες και τεχνολογικά προηγμένες τεχνικές μετάδοσης. Ένα σχετικά μεγάλο ποσοστό του λογισμικού που συναντά ένας απλός χρήστης Internet στο είναι εν δυνάμει κακόβουλο σε ποικίλες μορφές. Τα περισσότερα από αυτά τα Malware ανιχνεύονται από λογισμικό Antivirus, εφαρμογές αφαίρεσης Spyware και άλλα παρόμοια εργαλεία. Ωστόσο, η προστασία αυτή δεν είναι πάντα αρκετή και υπάρχουν αρκετές φορές που μία μικρή φαινομενικά καλοήθης εφαρμογή διαφεύγει από όλα αυτά τα μέτρα προστασίας και θέτει σε κίνδυνο το σύστημα και τα δεδομένα του χρήστη. Οι λόγοι για αυτήν την παραβίαση μπορεί να είναι:

- Οι χρήστες δεν ενημερώνουν τα Antivirus τους τακτικά
- Οι χρήστες δεν διατηρούν τα συστήματα τους αναβαθμισμένα και ενημερωμένα
- Η αποτυχία των ευρηστικών αλγορίθμων των Antivirus
- Νέα ή χαμηλού προφίλ Malware που δεν έχουν ακόμη ανακαλυφθεί από τους προμηθευτές Antivirus.
- Προσαρμοσμένα κωδικοποιημένο malware μη ανιχνεύσιμο από Antivirus
- Έλλειψη ή μη ορθή ρύθμιση Firewall.

Τα Malware εξελίσσονται συνεχώς και οι κατασκευαστές Antivirus δυσκολεύονται να συμβαδίσουν με αυτή τη συνεχώς αυξανόμενη απειλή. Σε ορισμένες περιπτώσεις επιλέγουν να μην συμπεριλάβουν μία υπογραφή ενός συγκεκριμένου malware. Ωστόσο, αυτό δεν πρέπει να εμποδίζει τους αναλυτές malware να εξετάζουν με τη χρήση freeware εργαλείων και τεχνικών τα αρχεία και να αναπτύξουν τους δικούς τους μηχανισμούς πρόληψης και ανίχνευσης. Αν και τα λογισμικά Antivirus εξελίσσονται όλο και περισσότερο, ένα μικρό, αλλά πολύ σημαντικό ποσοστό των Malware διαφεύγει από αυτήν την προκαθορισμένη διαδικασία ελέγχου και καταφέρνει να εισέλθει και να θέσει σε κίνδυνο τόσο το σύστημα αλλά και το ίδιο το δίκτυο. Δυστυχώς, το ποσοστό των Malware διαφεύγει από αυτήν των Antivirus αυξάνεται επίσης καθημερινά. Είναι σημαντικό για τους χρήστες και απολύτως απαραίτητο για τους διαχειριστές συστημάτων να είναι σε θέση να καθορίσουν εάν ένα αρχείο είναι επιβλαβές εξετάζοντας το χωρίς να στηρίζονται μόνο στις αυτοματοποιημένες μηχανές σάρωσης. Το επίπεδο της πληροφορίας που απαιτείται μετά από την ανάλυση διαφέρει ανάλογα με τις ανάγκες του χρήστη. Για παράδειγμα, ένας κανονικός χρήστης μπορεί να θέλει να γνωρίζει εάν μία εφαρμογή είναι κακόβουλη ή όχι, ενώ ένας διαχειριστής μπορεί να χρειάζεται να μάθει περισσότερα, όπως για τις τιμές registry που έχει επηρεάσει το εν δυνάμει κακόβουλο αρχείο ή τα αντίγραφα των μολυσμένων αρχείων που πιθανώς να έχουν δημιουργηθεί ή ακόμα τα είδη των αρχείων που μπορεί να έχουν μολυνθεί ή και επίσης το πραγματικό κακόβουλο ωφέλιμο φορτίο και είναι και τι κάνει. Αυτό σημαίνει ότι, μπορεί να χρειάζεται μια ολοκληρωμένη ανάλυση του δυαδικού αρχείου.

4.3 Στατική ανάλυση - Ανάλυση Κώδικα

Η Στατική Ανάλυση ή Ανάλυση κώδικα είναι μία από τις κύριες τεχνικές που χρησιμοποιούνται για την εξέταση malware. Ο καλύτερος τρόπος για την κατανόηση του τρόπου που ένα πρόγραμμα λειτουργεί είναι, φυσικά, η μελέτη του πηγαίου κώδικα του προγράμματος. Ωστόσο, ο πηγαίος κώδικας για τα περισσότερα malware δεν είναι διαθέσιμος. Τα malware συχνά διανέμονται με τη μορφή εκτελέσιμων, δυαδικού κώδικα. Παρ’ αυτά μπορεί ακόμη να εξεταστεί χρησιμοποιώντας προγράμματα εντοπισμού σφαλμάτων (debuggers) και αποσυναρμολογητές (disassembles). Ωστόσο, η χρήση αυτών των εργαλείων είναι συχνά περίπλοκη και απαιτεί ιδιαίτερες γνώσεις και ικανότητες, τα άτομα όμως που διαθέτουν αυτή την ικανότητα δοθέντος επαρκούς χρόνου κάθε δυαδικό αρχείο ανεξαρτήτου μεγέθους και πολυπλοκότητας μπορεί να αντιστραφεί πλήρως με τις τεχνικές της Στατικής ανάλυσης. [40]

4.4 Δυναμική ανάλυση - Ανάλυση συμπεριφοράς

Η Δυναμική ανάλυση ή Ανάλυση συμπεριφοράς έχει να κάνει περισσότερο με τις συμπεριφοριακές πτυχές του κακόβουλου λογισμικού. Όπως ένα θηρίο υπό παρακολούθηση σε ένα ζωολογικό κήπο, ένα δυαδικό αρχείο μπορεί να διατηρηθεί σε ένα στενά ελεγχόμενο περιβάλλον ώστε να ελεγχθεί η συμπεριφορά του. Αυτό γίνεται κυρίως σε περιβάλλον εικονικών λειτουργικών συστημάτων ή προσομοιωτών, έτσι ώστε τα αποτελέσματα του malware να μπορούν να διατηρηθούν υπό έλεγχο. Η ανάλυση των δραστηριοτήτων ή των αλλαγών που κάνει το malware στο περιβάλλον (σύστημα αρχείων, το μητρώο, δίκτυο, κλπ), η επικοινωνία του με το υπόλοιπο δίκτυο, η επικοινωνία του με απομακρυσμένες συσκευές και ούτω καθεξής παρακολουθούνται στενά και οι πληροφορίες που εξάγονται συλλέγονται. Τα συλλεγόμενα αυτά δεδομένα τεκμηριώνονται και αναλύονται έτσι ανασκευάζεται η πλήρης

εικόνα του malware από αυτά τα διαφορετικά κομμάτια. Ένα προτέρημα της Δυναμικής ανάλυσης είναι ότι εμπίπτει στο πεδίο εφαρμογής και γνώσεων του μέσου διαχειριστή ή ακόμα και ενός έμπειρου χρήστη. Και παρόλο που οι reverse engineering τεχνικές στην Στατική ανάλυση οδηγούν στη εξαγωγή του κώδικα της εφαρμογής, δεν είναι επαρκείς για τις ανάγκες των περισσότερων αναλυτών. Με τη Στατική ανάλυση μελετάται ένα πρόγραμμα χωρίς να εκτελείται. Τα εργαλεία που χρησιμοποιούνται είναι disassemblers, decompilers, αναλυτές πηγαίου κώδικα και ακόμη και κάποια βασικά βοηθητικά προγράμματα όπως το strings και το grep. Η Στατική ανάλυση έχει το πλεονέκτημα ότι μπορεί να αποκαλύψει πώς ένα πρόγραμμα συμπεριφερθεί κάτω από ασυνήθιστες συνθήκες, γιατί στην Στατική ανάλυση δίνεται η δυνατότητα της εξέτασης τμημάτων ενός προγράμματος που κανονικά δεν εκτελούνται. Στην εφαρμογή, η Στατική ανάλυση δίνει μια εικόνα κατά προσέγγιση. Είναι αδύνατον να προβλεφθεί πλήρως η συμπεριφορά του συνόλου μίας εφαρμογής. Σε αντίθεση κατά την Δυναμική ανάλυση μελετάται ένα πρόγραμμα καθώς εκτελείται. Εδώ, τα εργαλεία που έχουν εφαρμογή είναι debuggers, function call tracers, machine emulators, logic analyzers και network sniffers. Το κύριο πλεονέκτημα της δυναμικής ανάλυσης είναι ότι μπορεί να είναι γρήγορη και ακριβής. Ωστόσο, η δυναμική ανάλυση έχει το μειονέκτημα ότι "βλέπει αυτό που φαίνεται". Για τον ίδιο λόγο με την Στατική ανάλυση, δεν είναι δυνατόν να προβλεφθεί πλήρως η συμπεριφορά ενός μη τετριμμένου προγράμματος. Μια ειδική περίπτωση είναι η δυναμική ανάλυση "μαύρου κουτού"(black box), όπου το σύστημα μελετάται χωρίς καμία αρχική γνώση. Τα μόνα παρατηρήσιμα είναι οι είσοδοι, έξοδοι, οι συσχετίσεις τους και το χρονοδιάγραμμα τους. Σε ορισμένες περιπτώσεις, οι είσοδοι και έξοδοι περιλαμβάνουν την κατανάλωση ισχύος καθώς και την ηλεκτρομαγνητική ακτινοβολία. Η black box ανάλυση μπορεί να αποδώσει εξαιρετικά χρήσιμα αποτελέσματα παρά την φαινομενικά περιορισμένη του κατάσταση της. Τέλος, υπάρχει και η μεταθανάτια (post-mortem) ανάλυση, η μελέτη της συμπεριφοράς του προγράμματος που εξετάζει τις επιπτώσεις της εκτέλεσης. Παραδείγματα αποτελούν οι τοπικές ή απομακρυσμένες συνδέσεις χρηστών, οι αλλαγές στα περιεχόμενα αρχεία ή αλλαγές σε δικαιώματα πρόσβασης σε αρχεία, πληροφορίες για αρχεία που έχουν πρόσφατα διαγραφεί, τα δεδομένα που γράφτηκα στον swap χώρο του δίσκου, τα δεδομένα που εξακολουθούν βρίσκονται στην μνήμη, καθώς και πληροφορίες από εξωτερικούς παράγοντες από μία συσκευή. Η post-mortem ανάλυση είναι συχνά ο μόνος διαθέσιμος τρόπος μετά από ένα περιστατικό. Το μειονέκτημά της είναι ότι η πληροφορία εξαφανίζεται με την πάροδο του χρόνου στα πλαίσια της κανονικής συμπεριφοράς ενός συστήματος που όσο περνάει ο χρόνος διαβρώνονται αποδείξεις (πχ όταν τα στοιχεία βρίσκονται στην μνήμη RAM, μετά από κάποιο χρονικό διάστημα δεν είναι αναγνώσιμα).

4.4.1 Κίνδυνοι από τη δυναμική ανάλυση του προγράμματος

Η δυναμική ανάλυση υπονοεί την εκτέλεση του ύποπτου προγράμματος, για την εξέταση και την μελέτη του. Όμως υπάρχουν πολλά πιθανά προβλήματα με αυτή την προσέγγιση. Η ύποπτη εφαρμογή κατά πάσα πιθανότητα είναι κακόβουλη (γιαυτό και η ανάγκη για ανάλυση). Για να ελεγχθεί δυναμικά εκτελείται σχετικά ανεξέλεγκτα με το κακόβουλο φορτίο του να δρα, που αυτό μπορεί να σημαίνει από μία ενοχλητική μαζική αποστολή ηλεκτρονικού ταχυδρομείου έως και την καταστροφή του συστήματος ξενιστή. Για την αποφυγή, της οποιαδήποτε δυσάρεστης έκπληξης που συχνά κρύβουν τα malware, συνιστάται η χρήση sandbox. Ο όρος "sandbox" προέρχεται είτε:

Α) Από την βαλλιστική, όπου δοκιμάζονται οι βολές όπλων μέσα σε ένα κουτί γεμάτο με άμμο, έτσι ώστε οι σφαίρες να μην προκαλέσουν καμία ζημιά.

Β) Από το κουτί αφόδευσης της γάτας, όπου καθαρίζεται, η άμμος από τα περιττώματα.

Ένα λογισμικό sandbox είναι ένα ελεγχόμενο περιβάλλον για την εκτέλεση του λογισμικού και μπορεί να υλοποιηθεί με ποικίλους τρόπους. Η πιο απλή προσέγγιση αποτελεί το λεγόμενο “πρόβατο για σφαγή”(sacrificial lamb): Αποτελεί ένα πραγματικό λειτουργικό σύστημα μιας χρήσης, με περιορισμένη ή χωρίς καθόλου πρόσβαση στο δίκτυο. Αυτή είναι η πιο ρεαλιστική προσέγγιση Sandbox, αλλά μπορεί να είναι δύσχρηστη και αργή σε περιπτώσεις που χρειάζονται δοκιμαστικές επαναλήψεις. Αντί να αφιερώνει κανείς στο άγνωστο, εν δυνάμει malware, πρόγραμμα ένα ολόκληρο μηχάνημα, μπορεί να χρησιμοποιήσει πιο ανεπαίσθητες τεχνικές. Αυτές κυμαίνονται από παθητικά sandbox παρακολούθησης ενός προγράμματος, έως sandbox που μετατρέπουν το πρόγραμμα που εκτελείται να φαίνεται σαν μαριονέτα, που ελέγχεται εξ ολοκλήρου από τον ερευνητή.

4.4.2 Κίνδυνοι που εγκυμονούν με τις εικονικές μηχανές

Όταν μια εικονική μηχανή χρησιμοποιείται για την ανάλυση κακόβουλου κώδικα, δεν πρέπει να επιτρέπει στο μη αξιόπιστο λογισμικό να ξεφύγει. Εκτελώντας το malware περιορισμένο σε μία εικονική μηχανή απαιτεί όχι μόνο την ορθή εφαρμογή των χαρακτηριστικών προστασίας του hardware ή του επεξεργαστή, αλλά και την ορθή εγκατάσταση και υλοποίηση του λογισμικού που μεσολαβεί μεταξύ της εικονικής μηχανής και του πραγματικού συστήματος που φιλοξενεί το εικονικό μηχάνημα. Εάν ένα κακόβουλο λογισμικό είναι σε θέση να αναγνωρίσει ότι εκτελείται σε εικονικό περιβάλλον, τότε μπορεί να εκμεταλλευτεί τις πιθανές αδυναμίες ή κάποια χαρακτηριστικά της εκάστοτε εικονικής μηχανής, με πιθανά αποτελέσματα την διαφυγή του από αυτήν ή την καταστροφή της ή ακόμα και και την απενεργοποίηση του, κατανοώντας το πλέον μη εντοπίσιμο. Σε ορισμένες περιπτώσεις, συγκεκριμένες λεπτομέρειες μπορεί να προδώσουν ότι ένα λογισμικό εκτελείται σε ένα εικονικό μηχάνημα. Για παράδειγμα, μπορεί να παρατηρηθεί μια σχετική καθυστέρηση σε ορισμένους οδηγούς μηχανής (drivers). Όπως επίσης και οι χρόνοι εγγραφής και διαγραφής στον δίσκο είναι πιο αργοί, λόγο της μετάβασης από τον εικονικό δίσκο στον φυσικό. Επιπρόσθετα, πχ. το εικονικό περιβάλλον VMware είναι πραγματικά εύκολο να αναγνωριστεί πχ. Εικόνα 18. Μερικές λεπτομέρειες, όπως σημβολοσειρές ταυτοποίησης της συσκευής μπορεί να αναγνωριστούν από οποιαδήποτε διαδικασία που εκτελείται στην εικονική μηχανή, ενώ άλλα χαρακτηριστικά μπορούν να αναγνωριστούν ακόμη και από απόσταση. Ειδικότερα, η εικονική κάρτα ethernet έχει διεύθυνση με πρόθεμα 00:50:56, η οποία προορίζεται για όλα τα VMware, γεγονός που μπορεί να αναγνωρίζεται από απόσταση ως διεύθυνση IP (Version 6).

```
$ dmesg
. . .
lnc0: PCnet-PCI II address 00:50:56:10:bd:03
ad0: 1999MB <VMware Virtual IDE Hard Drive> [4334/15/63] at ata0-
master UDMA33
acd0: CDROM <VMware Virtual IDE CDROM Drive> at atal-master PIO4
. . .

$ ifconfig lnc0
lnc0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        address: 00:50:56:10:bd:03
        . . .
        inet6 fe80::250:56ff:fe10:bd03%le0 prefixlen 64 scopeid 0x1
        inet6 2001:240:587:0:250:56ff:fe10:bd03 prefixlen 64
```

Εικόνα 18, Οι σημβολοσειρές ταυτοποίησης της συσκευής

4.5 Συνιστώσες της ανάλυσης του Malware

Οι δημιουργοί malware χρησιμοποιούν διαφορετικές τεχνικές για την ανάπτυξη του κακόβουλου λογισμικού έτσι ώστε να είναι δύσκολο να καθοριστεί ένας κοινός άξονας σε όλα τα malwares. Κάθε κακόβουλο λογισμικό έχει διαφορετική υπογραφή, διαφορετική γλώσσα προγραμματισμού αλλά και συσκευαστή (packer). [41] Περίπου 500 συσκευαστές έχουν δημοσιευθεί και χρησιμοποιούνται από τους επιτιθέμενους, προκειμένου να αποτρέπουν την ανίχνευση του κακόβουλο λογισμικού από τα anti-virus. Οι Packers χρησιμοποιούνται έτσι ώστε ο κώδικας συμπιέζεται χρησιμοποιώντας εργαλεία όπως το 7-zip ή Win Rar κλπ. Ο συμπιεσμένος κώδικας είναι πιο δύσκολο να ανιχνευθεί δεδομένου ότι θα υπάρξει μια διεργασία με λιγότερη χρήση CPU σε αντίθεση με μεγάλα κομμάτια κώδικα που χρησιμοποιούν την μέγιστη χρήση της CPU ανά διεργασία. Μια άλλη μέθοδος για περισσότερη προστασία απέναντι στην ανίχνευση των Antivirus είναι η χρήση κρυπτογράφησης. Έτσι ώστε ακόμα κι εάν το anti-virus καταφέρει να αποσυμπιέσει το κακόβουλο λογισμικό, θα σαρώσει μόνο την κρυπτογραφημένη εκδοχή του κώδικα. Τα malware μπορεί να περιέχουν ταυτόχρονα πολλαπλά κακόβουλα στοιχεία όπως backdoors, exploits, Scripts, Spyware, Adware κλπ. Διασφαλίζοντας έτσι ότι τουλάχιστον ένας από αυτά θα προκαλέσει ζημιά στο σύστημα.

4.6 Εργαλεία Ανάλυσης Android Malware

Διάφορα εργαλεία επιλέγονται από τους αναλυτές malware για την ανάλυση και την εφαρμογή Δυναμικής και Στατικής ανάλυσης ενός κακόβουλου λογισμικού. Τα ακόλουθα είναι μια λίστα με τα πιο δημοφιλή εργαλεία που χρησιμοποιούνται για malware reverse engineering.

Περιβάλλοντα εργασίας :

Virtual Box, VMware, Sandbox GFI.

Στατική ανάλυση:

Dex2jar, Apktool, Dexdump, Axmlprinter, Jad, JD-GUI, Ded, Smali/Baksmali, Androguard APKInspector, IDA proDex2jar,, Apktool, Process Monitor, Wireshark, PEiD, TCPView, WinHex, Process Explorer, Winanalysis, Strings, κ.α.

Δυναμική ανάλυση:

Droidbox, Android SDK, Wireshark, Ollydbg, IDA Pro.

4.7 Στόχοι της ανάλυσης malware

Ο στόχος της ανάλυσης malware [42] είναι η έρευνα και η μελέτη των νέων απειλών ώστε να μπορεί να δομηθεί η ασφάλεια και η προστασία ενός ευρύτερου συστήματος από κακόβουλες επιθέσεις. Το πρώτο ερώτημα που τίθεται είναι πώς εκτίθεται ένα σύστημα σε κίνδυνο; Και το δεύτερο, τι αδυναμίες εκμεταλλεύτηκε το κακόβουλο λογισμικό; Ανάλογα με τον τύπο του malware και την ανάλυση που θα εφαρμοστεί, απαντώνται αυτά τα δύο αυτά ερωτήματα. Χρησιμοποιώντας τις πληροφορίες που εξάγονται από την ανάλυση, ένας αναλυτής είναι σε θέση να προσδιορίσει κατά πόσον το ύποπτο λογισμικό είναι ένα κακόβουλο λογισμικό ή όχι.

Συμπεράσματα 4ου κεφαλαίου

Η ανάλυση ενός malware μπορεί να εφαρμοστεί ως μία απλή και γρήγορη διαδικασία που μπορεί να εκτελέσει ένας απλός χρήστης αλλά επίσης μπορεί να είναι μία σύνθετη διαδικασία που εκτελείται σχεδιασμένα από ομάδα έμπειρων αναλυτών. Η στατική ανάλυση μπορεί να δώσει χρήσιμες πληροφορίες για μία εφαρμογή με βάση την μελέτη του κώδικα της. Ενώ η δυναμική ανάλυση μελετάει την εφαρμογή καθώς αυτή εκτελείται. Γίνεται κατανοητό ότι για καλύτερα αποτελέσματα χρειάζεται να γίνει ο συνδυασμός των δύο αναλύσεων με την στατική να τροφοδοτεί την δυναμική με τα ευρήματα της και την δυναμική να σχεδιάζεται ανάλογα με βάση την είσοδο αυτή.

Κεφάλαιο 5 Εργαλεία ανάλυσης

5.1 Γενικά εργαλεία

Notepad++ [43]

Το Notepad++ αποτελεί έναν εξελιγμένο κειμενογράφο για δημιουργία, επισκόπηση και εκτέλεση κώδικα. Στην ανάλυση malware χρησιμοποιείται για την μελέτη των αρχείων Java, xml και για κάθε τύπο αρχείου που θα κριθεί απαραίτητο.

WinZip /rar [44]

Τα Winzip - Winrar- Tar-gz 7zip (κτλ) χρησιμοποιούνται ευρύτατα στην ανάλυση είτε για την αποσυμπίεση των πακέτων apk είτε για οποιοδήποτε συμπιεσμένο αρχείο μπορεί να περιέχεται στην εφαρμογή υπό ανάλυση.

VirusTotal [45]

Το VirusTotal είναι μια δωρεάν υπηρεσία που ελέγχει ύποπτα αρχεία και διευθύνσεις URL και διευκολύνει στην γρήγορη ανίχνευση ιών, worms, trojans αλλά δεν καλύπτει όλα τα είδη malware. Το VirusTotal αποθηκεύει όλες τις αναλύσεις που πραγματοποιεί, αυτό επιτρέπει στους χρήστες να αναζητούν τις εκθέσεις εισάγοντας μόνο κάποιο MD5, SHA1, SHA256 ή URL. Οι απαντήσεις των αναζητήσεων επιστρέφουν την τελευταία σάρωση που εκτελέστηκε στοιχείο του ενδιαφέροντος. Το VirusTotal επιτρέπει επίσης την πραγματοποίηση αναζήτησης στα σχόλια που οι χρήστες αναρτούν για τα αρχεία και τις διευθύνσεις URL.

5.2 Εργαλεία Στατικής ανάλυσης

Apktool [46]

Το ApkTool είναι ένα εργαλείο reverse engineering για binary εφαρμογές Android είτε κλειστού κώδικα είτε 3rd party. Μπορεί να αποκωδικοποιήσει τους πόρους σχεδόν στην αρχική τους μορφή και να ανοικοδομήσει σε ένα μεγάλο βαθμό την αρχική εφαρμογή. Επίσης, κάνει την διαδικασία της ανάλυσης ευκολότερη λόγω της προσαρμογής που εφαρμόζει στην υπό-ανάλυση εφαρμογή ώστε να μοιάζει με τη δομή των αρχείων της υπό-δημιουργίας εφαρμογής. Δεν προορίζεται για πειρατεία και άλλες μη νόμιμες χρήσεις. Μπορεί να χρησιμοποιηθεί για περαιτέρω ανάπτυξη και υποστήριξη σε διάφορες προσαρμοσμένες πλατφόρμες, εφαρμογές ή βιβλιοθήκες. Με σεβασμό πάντα στον δημιουργό της υπό-ανάλυσης εφαρμογής.

Axmlprinter [47]

Τα αρχεία xml παρέχουν χρήσιμες πληροφορίες για την εφαρμογή. Όμως τα xml στο apk πακέτο είναι σε εκτελέσιμη δυαδική (binary) μορφή και σαφώς αυτό το καθιστά δυσανάγνωστο, το AXMLPrinter μετατρέπει τα binary xml στην αρχική τους ευανάγνωστη μορφή.

Ded [48]

Το Ded είναι ένα έργο που στοχεύει στην μεταγλώττιση εφαρμογών Android. Το Ded εργαλείο απευθύνεται Android εφαρμογές στην μορφή .Dex και τις μετατρέπει στα παραδοσιακά java Class αρχεία. Αυτά τα .Class τα αρχεία μπορούν στη συνέχεια να υποβληθούν σε επεξεργασία από διάφορα εργαλεία πχ. decompilers. Έτσι, οι Android εφαρμογές μπορούν να αναλυθούν χρησιμοποιώντας ένα ευρύ φάσμα από τεχνικές και εργαλεία που αναπτύχθηκαν για την ανάλυση των παραδοσιακών Java εφαρμογών.

Jad [49]

Το Jad (Java Decompiler) είναι ένα εγκαταλειμμένο πλέον decompiler για τη γλώσσα Java. Το Jad παρέχει ένα περιβάλλον γραμμής εντολών και εξάγει τον πηγαίο κώδικα από τα Class αρχεία. Μια γραφική διεπαφή χρήστη για Jad αποτελεί το JadClipse το οποίο είναι ένα plugin για το Eclipse IDE. Το Jad επίσημα έληξε στις 25 Φεβρουαρίου 2009. Η πιο πρόσφατη έκδοση του JAD λέει ότι υποστηρίζει μόνο Java εκδόσεις αρχείων Class 45,3, 46,0 και 47,0 και όχι αυτά που παράγονται από την Java 5 και έπειτα.

Smali/Baksmali [50]

Το smali/baksmali είναι ένας assembler/disassembler για αρχεία dex που χρησιμοποιούνται από το dalvik VM. Το συντακτικό είναι βασισμένο στους decompilers Jasmin και dedexer και υποστηρίζει πλήρως dex format (annotations, debug, lines,κτλ.). Τα ονόματα "smali" και "baksmali" είναι ισλανδικές αποδώσεις των "assembler" και "disassembler" αντίστοιχα.

Mobile Sandbox [51]

Παρέχει στατική ανάλυση σε αντίγραφα malware με ένα εύχρηστο web nterface.

IDA pro [52]

Πρόκειται για ένα πολύ γνωστό εργαλείο στους αναλυτές malware και αποτελεί disassembler και debugger. Υποστηρίζει Android bytecode μετά την έκδοση 6.1 pro.

Dex2jar [53]

Είναι ένα εργαλείο για μετατροπή αρχείων Android μορφής .dex σε μορφή Java .class

Dexdump [54]

Είναι ένας decompiler για .dex αρχεία.

JD-GUI [55]

Το JD-GUI είναι ένα αυτόνομο εργαλείο που εμφανίζει πηγαίους κώδικες Java από αρχεία .class δίνει την δυνατότητα περιήγησης στον πηγαίο κώδικα της ανακατασκευασμένης εφαρμογής και άμεση πρόσβαση στις μεθόδους και τα πεδία των κλάσεων.

Androguard [56]

Αποτελεί μια σουίτα για Android reverse engineering.

APKInspector [57]

Το APKinspector αποτελεί ένα πολύ δυνατό εργαλείο με γραφικό interface για ανάλυση Android εφαρμογές και malware. (Το Androguard είναι πλέον ενσωματωμένο στο APKInspector).

5.3 Εργαλεία Δυναμικής Ανάλυσης

Droidbox [58]

Αποτελεί ένα Sandbox περιβάλλον για δυναμική ανάλυση εφαρμογών Android. Το Droidbox παρακολουθεί την εφαρμογή κατά την εκτέλεση και καταγράφονται όλες οι δραστηριότητες της εφαρμογής. Παρατηρούνται οι αλληλεπιδράσεις της εφαρμογής με το δίκτυο και την συσκευή. Όπως επίσης τι πόρους χρησιμοποιεί και τι τιμές επιστρέφει στο σύστημα.

Android SDK [59]

Το Android Software Development Kit είναι η πλατφόρμα ανάπτυξης λογισμικού Android. Το Android SDK περιλαμβάνει demo projects με τον πηγαίο κώδικα, εργαλεία ανάπτυξης, έναν εξομοιωτή και τις απαιτούμενες βιβλιοθήκες για τη δημιουργία εφαρμογών Android. Οι εφαρμογές γράφονται με την χρήση της γλώσσας Java και εκτελούνται στην εικονική μηχανή Dalvik. Με την χρήση του Android SDK είναι δυνατή η δημιουργία μίας εικονικής συσκευής Android σχεδόν πανομοιότυπα σε λειτουργικότητα και δυνατότητες μίας συσκευής Android και δίνει την δυνατότητα ανάπτυξης εφαρμογών για οποιαδήποτε επιθυμητή έκδοση. Επίσης χρησιμοποιείται ως ασφαλές περιβάλλον που μπορούν να εγκατασταθούν και να εκτελεστούν malware έτσι ώστε να είναι δυνατή η μελέτη της συμπεριφορά τους.

Wireshark [60]

Το Wireshark είναι ένας αναλυτής κίνησης δικτύου που συλλαμβάνει όλη την κίνηση του δικτύου και παρέχει δυνατότητες ανάλυσης των πακέτων.

5.4 Διανομές ανάλυσης

Android Reverse Engineering (A.R.E.) [61]

Το A.R.E αποτελεί μία παραμετροποιημένη Ubuntu διανομή που διατίθεται σε εικονική μορφή και περιέχει όλα τα απαραίτητα εργαλεία για Android Reverse Engineering και ανάλυση malware.

Συγκεκριμένα περιέχει τα εξής: Androguard, Android sdk/ndk,APKInspector,Apktool, Axmlprinter, Ded, Dex2jar, DroidBox, Jad,Smali/Baksmali.

OSAF Virtual Machine [62]

Το OSAF όπως και το A.R.E αποτελεί μία παραμετροποιημένη Ubuntu διανομή και περιέχει όλα τα απαραίτητα εργαλεία για Android malware και forensics ανάλυση.

Κεφάλαιο 6 Ανάλυση Android malware

6.1 Προετοιμασία εργαστηρίου ανάλυσης

6.1.1 Θέματα λειτουργικών συστημάτων

Κάθε malware συμπεριφέρεται ανάλογα με το λειτουργικό σύστημα που εκτελείται. Κάποιες κατηγορίες malware μπορούν να μολύνουν μόνο ένα είδος λειτουργικού, ενώ άλλα μπορεί να έχουν ως στόχο τον πυρήνα του λειτουργικού ή να δρουν μέχρι κάποια έκδοση. Υπάρχουν και περιπτώσεις όπου το ίδιο malware λειτουργεί διαφορετικά σε διαφορετικές εκδόσεις. Για παράδειγμα μπορεί ένα συγκεκριμένο malware να είναι κατασκευασμένο έτσι ώστε εάν βρίσκεται σε Windows Server να προκαλεί κατάρρευση στο σύστημα, ενώ αν βρίσκεται σε Windows 7, να λειτουργεί ως μέλος ενός botnet και να εκτελεί απομακρυσμένες ελεγχόμενες DDos επιθέσεις. Τα malware που μολύνουν Android δεν μπορούν να μολύνουν κάποιο άλλο τύπο λειτουργικού όπως iOS, ή Windows, Linux κτλ. Το θέμα όμως που τίθεται είναι το κατά πόσο ένα Android malware μπορεί να μολύνει μία έκδοση ή περισσότερες εκδόσεις. Για αυτό και κατά την ανάλυση Android malware χρειάζεται να υπάρχουν πολλαπλές εκδόσεις του λειτουργικού. Τα μηχανήματα αυτά θα αποτελέσουν τους μολυσμένους ξενιστές. Επίσης σε διαφορετικά μηχανήματα θα εγκατασταθούν στην πραγματικότητα τα εργαλεία της ανάλυσης έτσι ώστε να μπορεί να εξετάζεται το κακόβουλο λογισμικό καθώς εκτελείται. Επιπλέον των μολυσμένων hosts συχνά χρειάζεται και ένα μηχάνημα με διάφορους διακομιστές εφαρμογών (application servers), έτσι ώστε τα μολυσμένα μηχανήματα να μπορούν να αλληλεπιδράσουν με αυτό. Για παράδειγμα, ένα δείγμα malware μπορεί να προσπαθήσει να επικοινωνήσει με κάποιον άλλον αποσκορακισμένο server μέσω IRC, οπότε θα χρειαστεί ένα μηχάνημα που θα τρέχει ένα IRC server ώστε να ανακατευθυνθεί η επικοινωνία από τον μολυσμένο host έτσι ώστε μπορεί να εξεταστεί. [63].

6.1.2 Απομόνωση δικτύου

Ιδιαίτερη προσοχή θα πρέπει να ληφθεί σχετικά με τη θέση των malware ανάλυσης host σε ένα δίκτυό. Οι host ανάλυσης όντας μη Android κινδυνεύουν από worm και άλλα malware. Στα Android (όπως και σε όλα τα λειτουργικά) υπάρχει ο κίνδυνος του να χρησιμοποιείται κάποιο 0-day vulnerability που σε εικονικό περιβάλλον όπως και στις περιπτώσεις worm απλά δυσχεραίνεται ή γίνεται αδύνατη η έρευνα με πιθανές καταστροφές των host.

Η απομόνωση των host ανάλυσης malware από άλλους υπολογιστές του δικτύου συχνά δεν είναι αρκετή. Συνήθως, θα πρέπει να απομονώνονται και από το Internet. Ο πρώτος λόγος για αυτό είναι ότι κάποια malware μπορεί να έχουν κατασκευαστεί έτσι ώστε όταν ξεκινάνε να εκτελούν μια επίθεση DOS εναντίον ενός άλλου υπολογιστή μέσω Internet. Γεγονός που μπορεί να προκαλέσει μεγάλη αναστάτωση και να έχει αρνητικές επιπτώσεις στον ίδιο τον ερευνητή, μίας και αυτός θα χρεωθεί την επίθεση. Ένα επιπλέον επιχείρημα για την απομόνωση των μηχανών του εργαστήριο από το Internet είναι να αποτρέψει το κακόβουλο λογισμικό από το να ενημερώσει τον συγγραφέα του για την ύπαρξη του σε αυτό το περιβάλλον ανάλυσης. Είναι πολύ πιθανό ένα malware που εκτελείται να έχει ρυθμιστεί ώστε να κάνει το λεγόμενο "phone home " την κλήση σε ένα διακομιστή διοίκησης και ελέγχου που επιτρέπει στον συγγραφέα του να γνωρίζει ότι εκτελείται το συγκεκριμένο αντίγραφο malware στον συγκεκριμένο υπολογιστή. Σε αυτό το σημείο, ο εισβολέας- συγγραφέας αν αντιληφθεί ότι το malware είναι υπό ανάλυση μπορεί να αρχίσει την εκτέλεση εντολών στο σύστημα

εργαστηρίου ώστε να το απενεργοποιήσει ή να ματαιώσει τις προσπάθειες ανάλυσής. Με όλες αυτές τις ανησυχίες να λαμβάνονται υπόψη, ένα host ανάλυσης malware θα πρέπει να είναι εντελώς απομονωμένο από το δίκτυο. Αυτό επιτυγχάνεται καλύτερα όταν τα συστήματα του εργαστηρίου δεν είναι συνδεδεμένα σε κανένα απολύτως δίκτυο.

6.1.3 Φυσικά ή εικονικά εργαστήρια

Σε ορισμένες περιπτώσεις, μπορεί να μην υπάρχει η χρηματοδότηση ή η δυνατότητα για την αγορά ή τη χρήση πολλαπλών σταθμών εργασίας σε ένα εργαστήριο. Σε αυτή την περίπτωση, Το εικονικό λογισμικό μπορεί να αποβεί σωτήρια λύση. Επίσης η χρήση των εικονικών μηχανών ως hosts σε ένα εργαστήριο έχει και μερικά άλλα πλεονεκτήματα τα οποία συχνά παραβλέπονται:

Στιγμιότυπα (Snapshots)

Το εικονικό λογισμικό επιτρέπει στον αναλυτή να αποθηκεύσει την τρέχουσα κατάσταση μίας εικονικής μηχανής ως ένα snapshot, έτσι ώστε να μπορεί να επιστρέψει σε αυτή όταν είναι αυτό κριθεί απαραίτητο. Τα Snapshots είναι πολύ χρήσιμα όταν πρόκειται για την ανάλυση malware, επειδή επιτρέπουν την επαναφορά του host πίσω σε μια ασφαλή κατάσταση ή μια κατάσταση προηγούμενη της μόλυνσης. Ένας αναλυτής χρησιμοποιώντας snapshots μπορεί έχει μια έτοιμη εικονική μηχανή που περιέχει ένα λειτουργικό σύστημα με προ-εγκατεστημένα τα εργαλεία στατικής και δυναμικής ανάλυσης. Αυτή η κατάσταση θα μπορούσε να αποτελεί ένα snapshot εκκίνησης. Και θα χρησιμοποιείται κάθε φορά που χρειάζεται να αναλυθεί ένα νέο malware. Επίσης ένα καλό χρονικά σημείο για αποτύπωση ενός snapshot αποτελεί η στιγμή που εγκαθίσταται το malware έτσι ώστε να μπορεί να συγκριθεί με την τελική κατάσταση. Όταν τελειώσει η εξέταση του δείγματος malware, ο αναλυτής μπορεί να επιλέξει είτε να αποθηκεύσει ή να απορρίψει αυτό το Snapshot και να επιστρέψει σε μια καθαρή κατάσταση. Είναι ασφαλές να υπάρχει τουλάχιστον ένα ζευγάρι snapshots για κάθε εξέταση malware. Πιο πολύπλοκα δείγματα μπορεί να χρειάζονται τη δημιουργία δεκάδων Snapshots.

Ταχεία ανάπτυξη λειτουργικού συστήματος

Χρησιμοποιώντας το λογισμικό virtualization επιτρέπει στον αναλυτή να χτίσει και να αποθηκεύσει μια βιβλιοθήκη εικονικών μηχανών ώστε να έχει πρόσβαση σε οποιοδήποτε λειτουργικό σύστημα επιθυμεί είναι μόνο μερικά κλικ. Χρησιμοποιώντας αυτή τη στρατηγική, σε συνδυασμό με την τεχνολογία snapshots είναι σε θέση να αντιμετωπίσει οποιοδήποτε τύπο malware. Στην ανάλυση Android malware αυτό υφίσταται σε επίπεδο εκδόσεων από την στιγμή που το Android SDK επιτρέπει την δημιουργία εικονικών συσκευών με οποιαδήποτε εγκατεστημένη έκδοση επιθυμεί ο αναλυτής.

Εικονικά Δίκτυα

Οι πιο κοινές εικονικές πλατφόρμες, όπως VMWare Workstation ή VirtualBox παρέχουν προηγμένες δυνατότητες δικτύωσης δίνουν την δυνατότητα δημιουργίας segmented δικτύων ή απομονωμένων δικτύων με δικό τους εύρος διευθύνσεων και διακομιστές DHCP. Το γεγονός αυτό καθιστά την απομόνωση μολυσμένων host πολύ εύκολη διαδικασία.

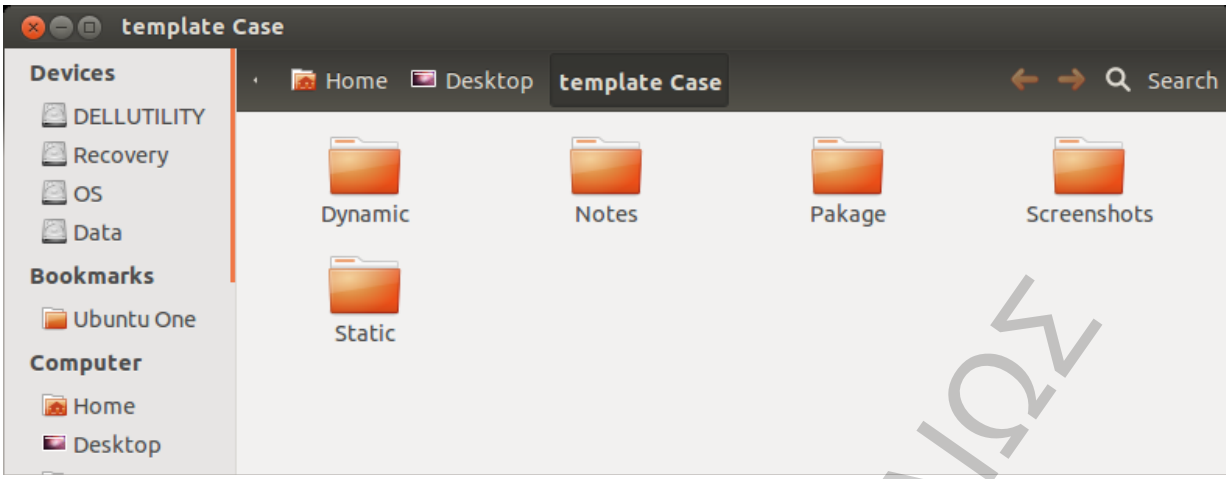
Τυποποιημένο Hardware

Το όφελος αυτό συνήθως παραβλέπεται αλλά δεν παύει να είναι σημαντικό. Η ύπαρξη τυποποιημένου hardware κατά την εκτέλεση ανάλυσης malware εξασφαλίζει ότι τα αποτελέσματά της ανάλυσης είναι επαναλαμβανόμενα και συνεπής και ανεξάρτητα από τυχόν αποτυχίες του hardware.

Εικόνα 19. Εικονικές συσκευές Android

6.2 Προετοιμασία ανάλυσης

Κατά την διάρκεια της ανάλυσης παράγεται μεγάλος όγκος πληροφορίας σε διάφορες μορφές αρχείων (κείμενα, εικόνες, κώδικες, εκτελέσιμα κτλ) για βέλτιστα αποτελέσματα και την αποφυγή μίας χαώδους κατάστασης απαιτείται καλός σχεδιασμός και οργάνωση της υπόθεσης προς ανάλυση. Μία καλή πρακτική αποτελεί η δημιουργία ενός φακέλου σε ένα εύκολα προσβάσιμο σημείο και με όνομα σχετικό με την υπόθεση προς ανάλυση. Ένα σύνηθες όνομα είναι το όνομα της εφαρμογής προς ανάλυση ακολουθούμενο από τον όρο “Case”. Μέσα στο φάκελο, η οργανωτική δομή οφείλει να είναι επικεντρωμένη στην διαδικασία της ανάλυσης έτσι ώστε να καλύπτει τις απαιτήσεις ταξινόμησης και οργάνωσης του όγκου πληροφορίας που θα παραχθεί. Μία λειτουργική λύση αποτελεί η ακόλουθη εικόνα:



Εικόνα 20, Δομή οργάνωσης των φακέλων για την ταξινόμηση των εξαγόμενων πληροφοριών σε μία ανάλυση

Όπου η πληροφορίες αποθηκεύονται ανάλογα με τον τύπο της ανάλυσης που έχει διενεργηθεί. Οι σημειώσεις, οι εικόνες όπως και τα αρχικά αρχεία της εφαρμογής τοποθετούνται σε ξεχωριστούς φακέλους. Φυσικά ανάλογα με την περίπτωση μπορεί να μην χρησιμοποιηθούν όλοι οι φάκελοι ή μπορεί να χρειαστεί η δημιουργία νέων φακέλων ή υποφακέλων (πχ αν προκύψουν πολλές εικόνες ο φάκελος Screenshots μπορεί να περιέχει τους υποφακέλους Dynimic_screens, Static_screens) [64].

6.3 Στατική ανάλυση

Σκοπός της στατικής ανάλυσης είναι η μελέτη της εφαρμογής χωρίς αυτή να εκτελείται. 1^{Ο} βήμα σε αυτήν είναι η ανάγνωση του AndroidManifest, ακολουθούν η ανάκτηση και η μελέτη του πηγαίου κώδικα. Με σκοπό την κατανόηση της λειτουργικότητας και την ανάλογη εξαγωγή συμπερασμάτων όσο αφορά την κακόβουλη διάθεση της εφαρμογής. [65]

Δείγματα ανάλυσης

για την υλοποίηση της ανάλυσης θα χρησιμοποιηθούν τρία δείγματα κακόβουλου λογισμικού, τυχαία επιλεγμένα από το Contagio moblie ένα malware repository [66].

Δείγμα 1ο: **LeNA** [67]

Android DKFBootKit aka LeNa.b and LeNa.c DroidKungFu variant)

Όνομα αρχείου: com.rovio.new.ads-LeNa.c.apk

MD5: 3B524DD4A7BBD2DE633EBFCFF167FED2

Δείγμα 2ο **Liveprints** [68]

Android PJApps - 2011 - Liveprints wallpaper

Όνομα αρχείου: Newfpwap_com_liveprintslivewallpaper.apk

Μέγεθος: 1316981 bytes

MD5: A84997B0D220E6A63E2943DA64FFA38C

Δείγμα 3ο **sbooster** [69]

Android Gamex Trojan

Όνομα αρχείου: de.mehrmannd.sdbooster-GAMEX.apk

Μέγεθος: 256139 bytes

MD5: 50836808A5FE7FEBB6CE8B2109D6C93A

Sample Credits: with many thanks to Tim Strazzere, April 30, 2012

Research: Security Alert: Gamex Trojan Hides in Root-Required Apps – Tricking Users into Downloads - Lookout

Δείγμα 4ο **VDloader Android** [70]

Όνομα αρχείου: waterfall3dLive.boa.liveWPcube.apk

Μέγεθος: 723022 bytes

MD5: 6AF90ADD478E4D27B4170FA791E635EE

Δείγμα 5ο **Zitmo** Android Edition (Zeus for mobile) [71]

Όνομα αρχείου: zitmo.apk

Μέγεθος 172722 bytes

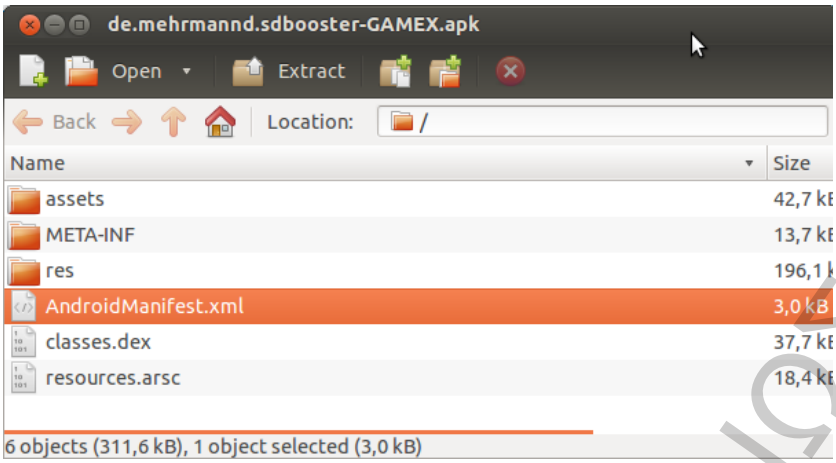
MD5: ecbbce17053d6eaf9bf9cb7c71d0af8d

6.3.1 Βήμα 1ο: Ανάγνωση του AndroidManifest

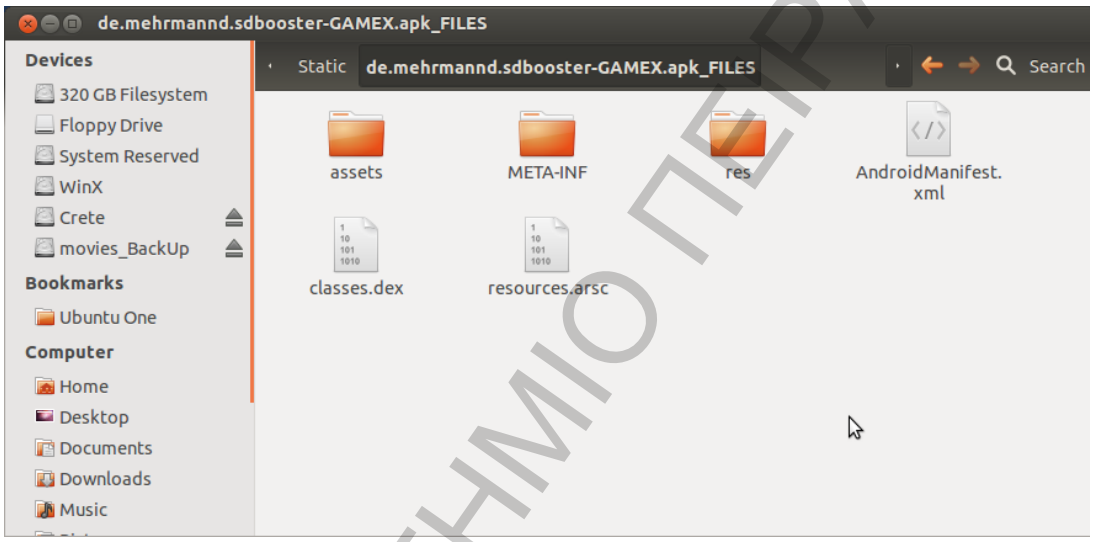
Ανάγνωση του Android Manifest

Εργαλεία : winRar, Axmlprinter, Notepad++, getid, κτλ.

Το AndroidManifest.xml [72] παρέχει αρκετά χρήσιμα στοιχεία για την δομή και λειτουργία μίας εφαρμογής. Πληροφορίες όπως το όνομα της εφαρμογής, οι εκδόσεις του SDK που χρησιμοποιεί, ποια είναι η εναρκτήρια κλάση, ποία services ξεκινούν κατά την εκκίνηση και παράλληλα τα δικαιώματα που απαιτεί. Το androidManifest.xml βρίσκεται στο πακέτο .apk της εφαρμογής. Το .apk αποσυμπιέζεται με οποιαδήποτε πρόγραμμα αποσυμπίεσης.

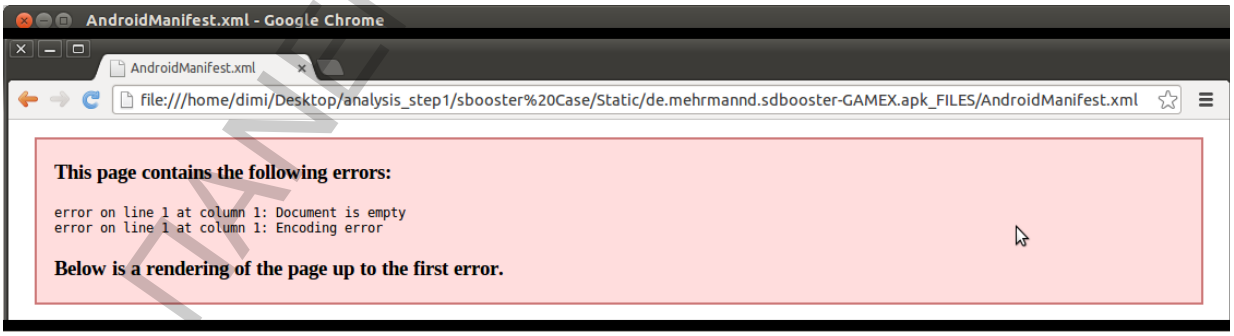


Εικόνα 21. Τα περιεχόμενα του πακέτου apk

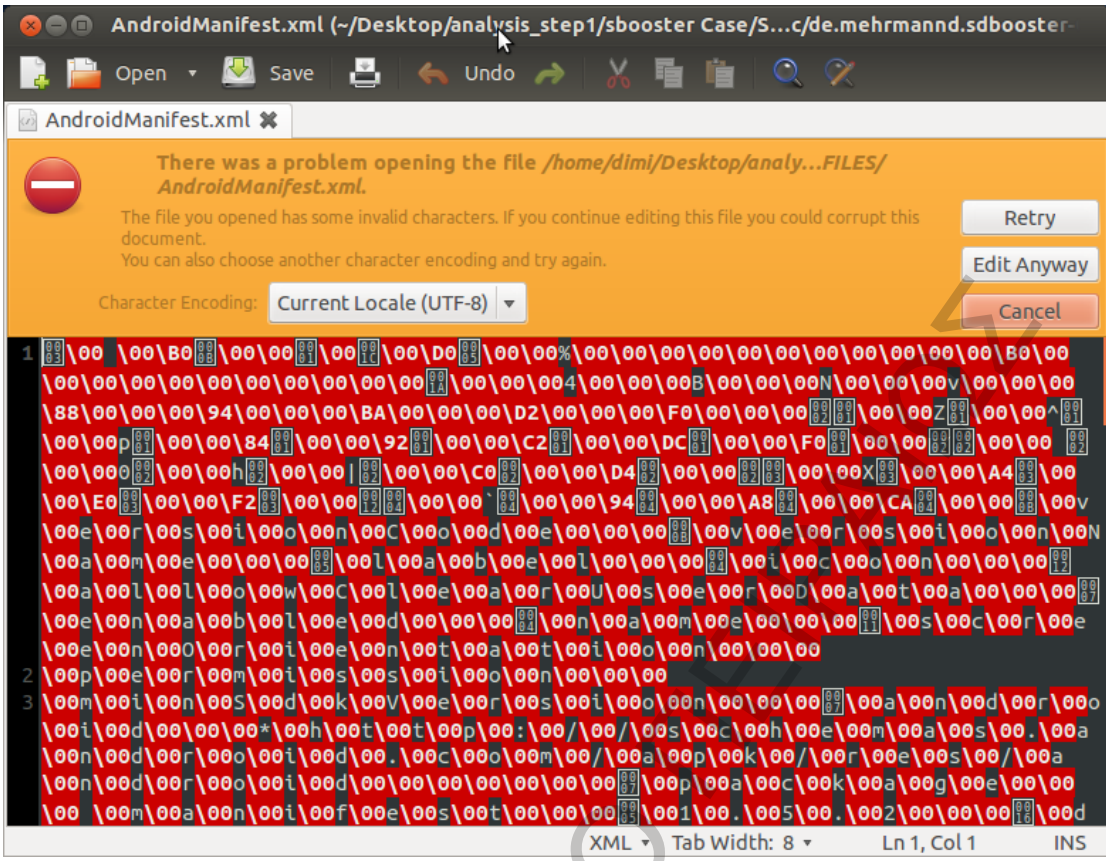


Εικόνα 22. Ο φάκελος με τα περιεχόμενα μετά από την αποσυμπίεση

Κατά την ανάπτυξη (deploy) του πακέτου apk, το andoroidManifest.xml μετατρέπεται σε δυαδική, μη αναγνώσιμη μορφή. Αυτό συμβαίνει για να είναι βέλτιστη η απόδοσή της εφαρμογής, η οποία το χρησιμοποιεί σαν εκτελέσιμο αρχείο.



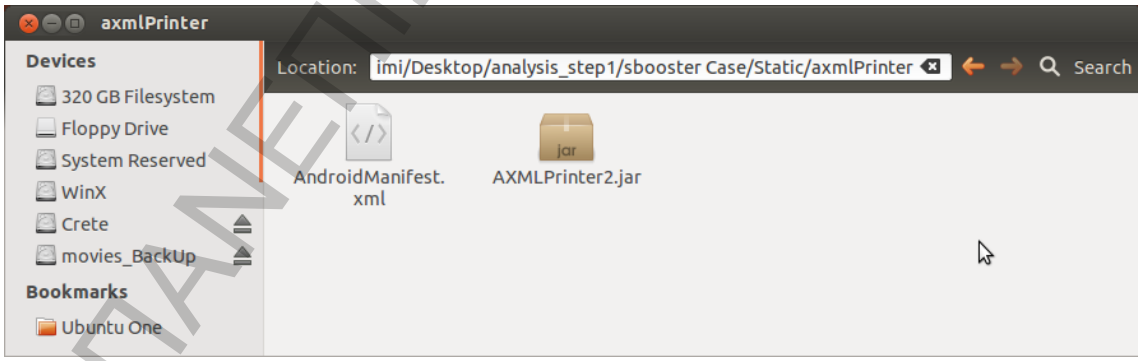
Εικόνα 23. Το andoroidManifest δεν είναι αναννώσιμο από τον Chrome



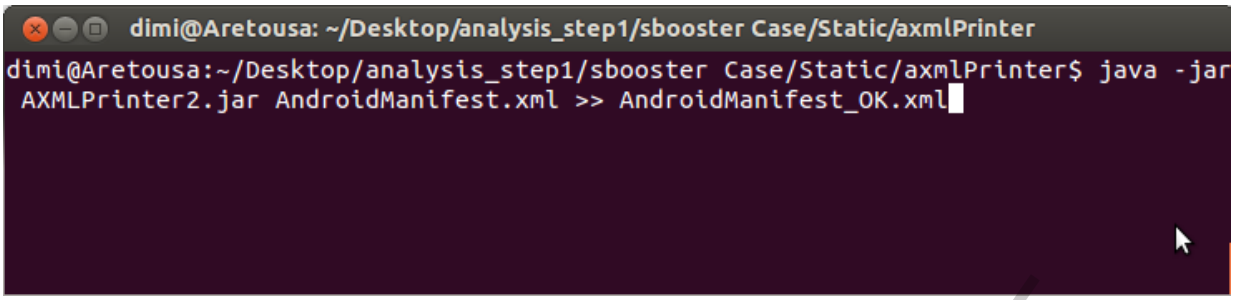
Εικόνα 24, Το andoroidManifest από το gedit

Για την μετατροπή του σε αναγνώσιμη μορφή χρησιμοποιείται το εργαλείο AXMLPrinter2.jar που μετατρέπει το androidManifest από δυαδική (byte) μορφή σε μορφή κείμενου.

Αντιγράφεται το androidManifest.xml στον ίδιο φάκελο με το jar και εκτελείται η ακόλουθη διαδικασία:

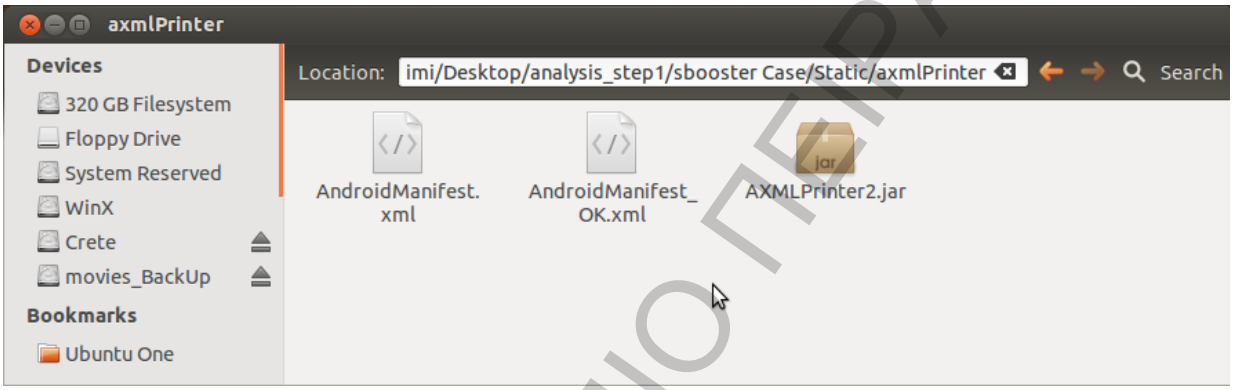


Εικόνα 25, Αντιγραφή του androidManifest στον home φάκελο του AXMLPrinter2



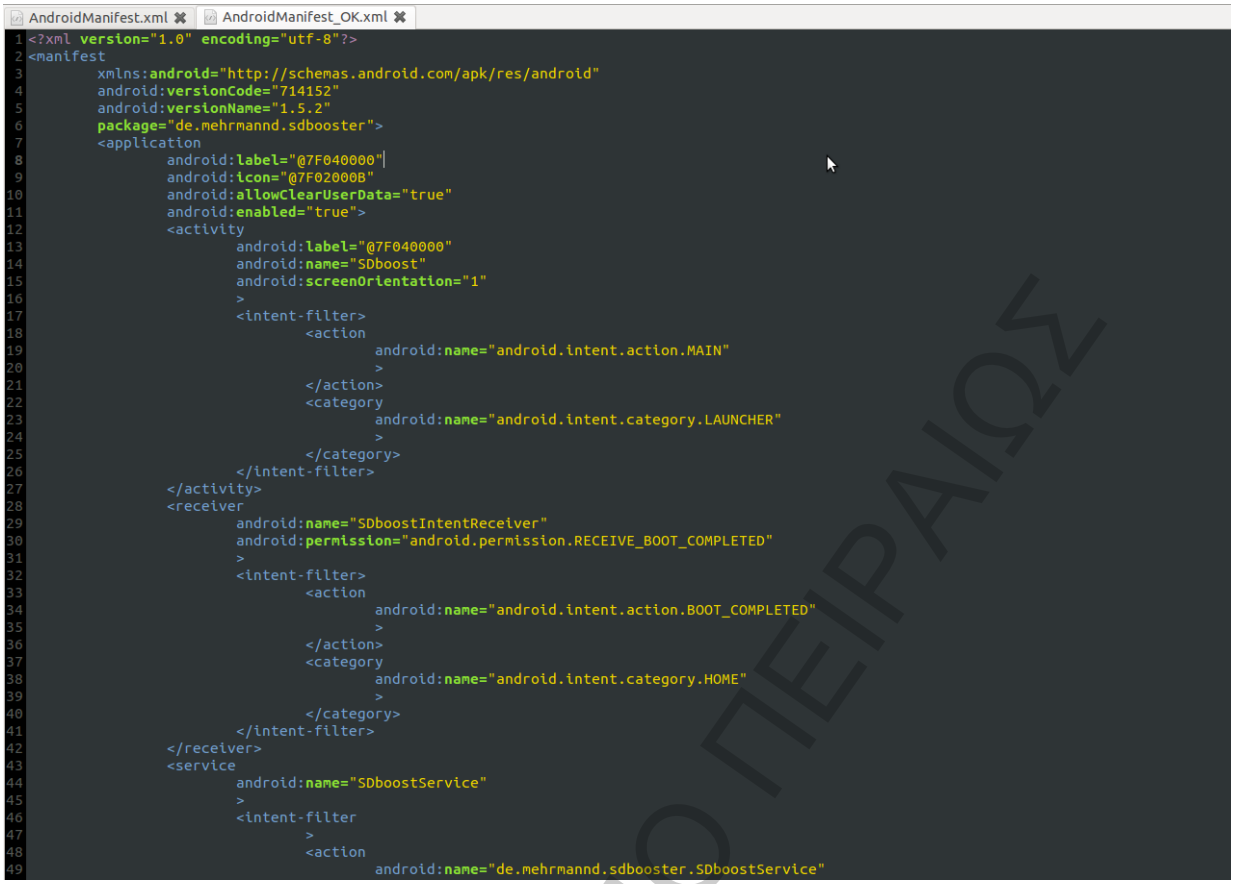
Εικόνα 26, Εντολή παρανωής της αναννώσιμης μορφής του androidManifest.xml

Με αυτή την εντολή, εκτελείται το AXMLPrinter2.jar και ως αποτέλεσμα της εκτέλεσης εξάγεται η αναγνώσιμη μορφή του AndroidManifest.xml σε ένα νέο αρχείο το AndroidManifest_OK.xml



Εικόνα 27, Μετά την εκτέλεση του AXMLPrinter2 έχει παραχθεί το AndroidManifestOK.xml

Εικόνα 28, Το AndroidManifestOK.xml από τον Chrorme



Εικόνα 29, Το AndroidManifestOK.xml από το gedit

Το αναγνώσιμο πλέον androidManifest.xml ως AndroidManifestOK.xml, περιέχει ιδιαίτερα σημαντικές πληροφορίες διότι συνήθως τα κακόβουλα λογισμικά στα Android εκμεταλλεύονται είτε κάποιο δικαίωμα είτε κάποια υπηρεσία είτε συνδυασμό των δύο.

Συγκεκριμένα εντοπίζεται το ακόλουθο ύποπτο δικαίωμα:

```
<uses-permission android:name = "android.permission.RECEIVE_BOOT_COMPLETED">
      </uses-permission>
```

Το οποίο χρησιμοποιείται συχνά από malware[33] για να εκκινήσει κάποιο κακόβουλο Service.

6.3.2 Βήμα 2ο: Ανάκτηση πηγαίου κώδικα

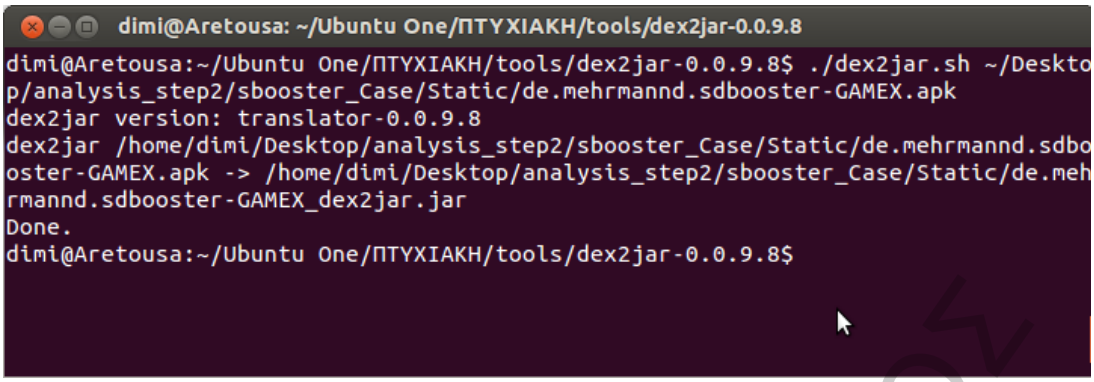
Εργαλεία: Dex2jar, Dexdump, Smali/Baksmali, JD-GUI, Ded, Jad

Από .apk σε .jar

Dex2jar:

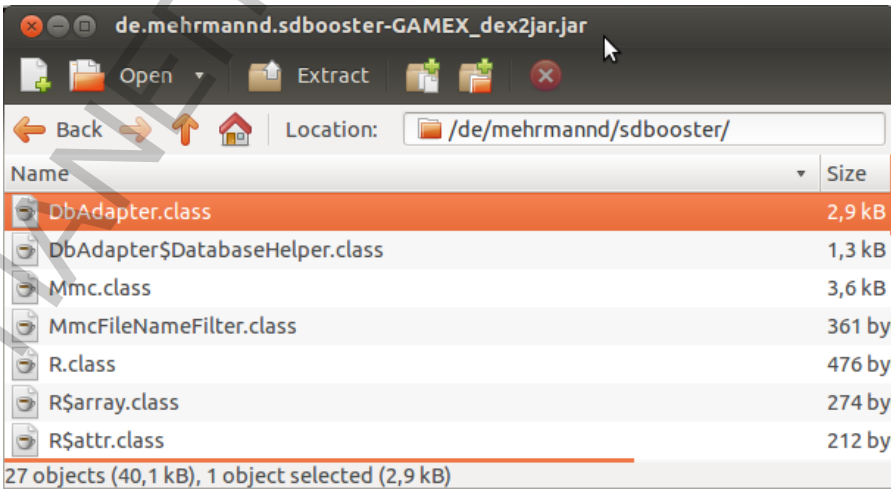
Χρήση: `dex2jar.sh αρχείο_εφαρμογής.apk`

33 Κεφάλαιο 3.2.4 Δικαιώματα χρήσης



Εικόνα 30. Εντολή εκτέλεσης του Dex2iar

και παράγει ως έξοδο το: de.mehrmannd.sdbooster-GAMEX_dex2jar.jar που περιέχει τα αρχεία class της εφαρμογής (Εικόνα 31).



Εικόνα 31. Τα περιεχόμενα του παραχθέντος jar

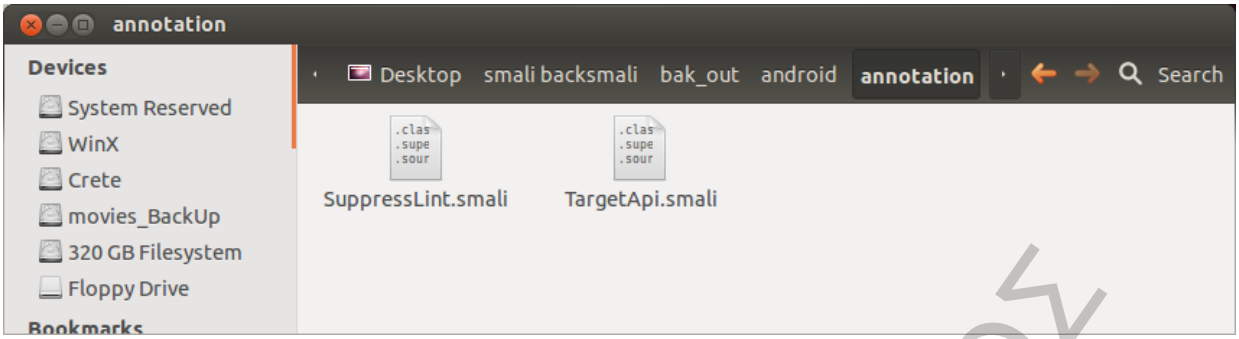
smali/baksmali:

(dex → smali) [50]

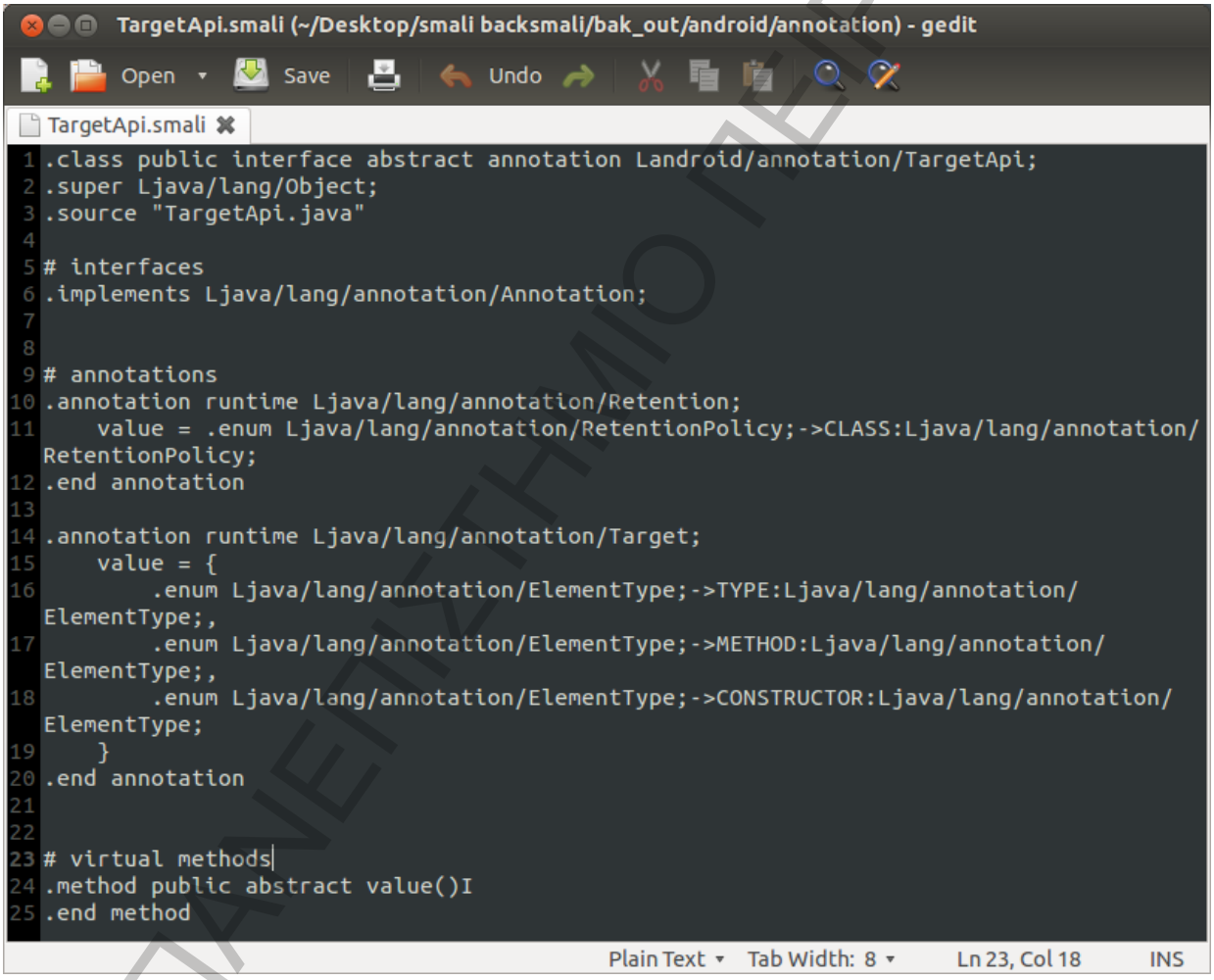
Το αρχείο classes.dex περιέχει όλα τα αρχεία class. Χρήση: java -jar baksmali-0.93.jar -o <output directory> <το αρχείο.dex, συνήθως είναι το classes.dex>

παράδειγμα:`~/Desktop/smali backsmali$ java -jar baksmali-1.3.2.jar -o bak_out/ classes.dex`

Εξάγει τον φάκελο bak_out που περιέχει τα αρχεία της εφαρμογής στην μορφή αρχείου της γλώσσας μηχανής smali:



Εικόνα 32. Τα αρχεία smali που παράχθηκαν



Εικόνα 33. Το TargetApi.smali

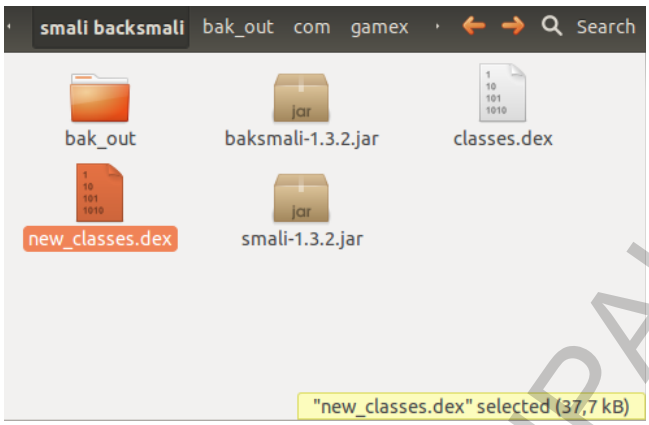
Η αντίστροφη διαδικασία (smali) είναι η παραγωγή ή η ανασυσκευασία του .dex αρχείου:

χρήση:

```
java -Xmx512M -jar smali-0.92.jar <output directory from above step> -o <name for new.dex output file>
```

παράδειγμα:

```
dimi@Aretousa:~/Desktop/smali backsmali$ java -Xmx512M -jar smali-1.3.2.jar./bak_out -o new_classes.dex
```



Εικόνα 34, Η παραγωγή του νέου dex αρχείου

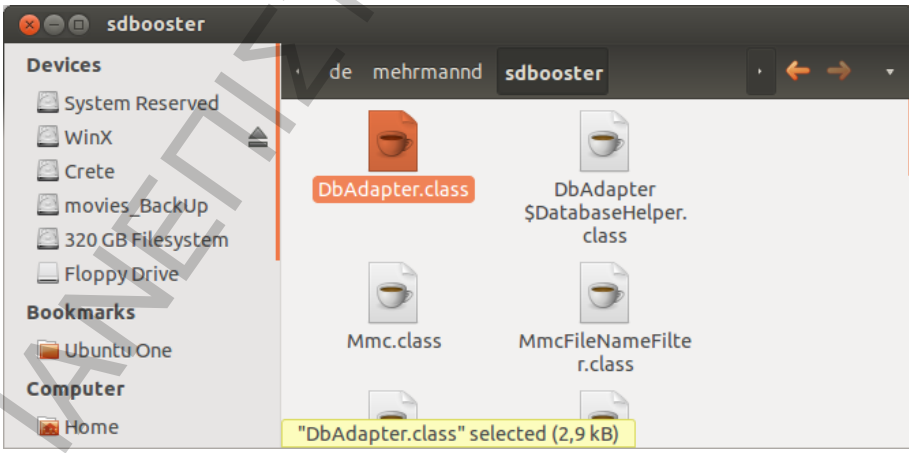
Από .dex σε .class

JAD: "Java Decompiler"[34] [49]

Χρήση: jad -o -d <class files>

παράδειγμα:

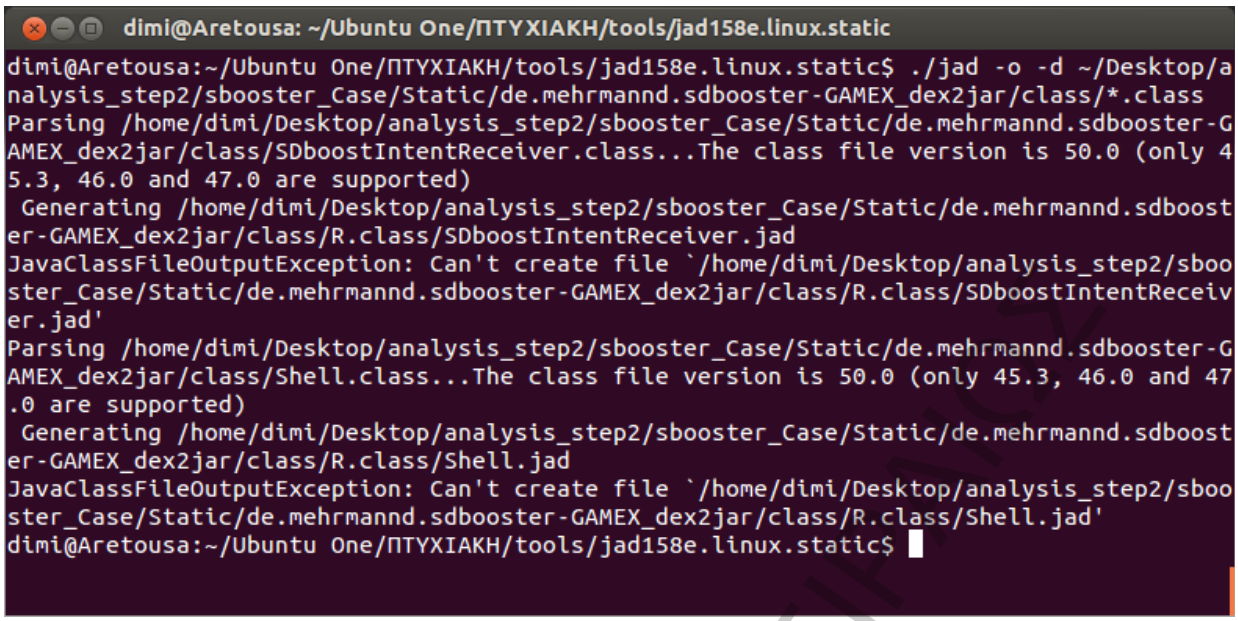
Σαν είσοδος χρησιμοποιούνται τα περιεχόμενα του πακέτου .jar που δημιουργήθηκε με την χρήση του **dex2jar**. Τυχαία επιλέγεται ο φάκελος sdbooster που περιέχει 27 class αρχεία, από το αποσυμπιεσμένο jar:



Εικόνα 35, Τα εξαγόμενα class αρχεία

Στον home φάκελο του jad: "~/Ubuntu One/ΠΤΥΧΙΑΚΗ/tools/jad158e.linux.static" Με ανεπιτυχές αποτέλεσμα για όλες τις κλάσεις:

34 Code: http://www.varaneckas.com/jad



Το class file version[35] είναι η έκδοση των μεταγλωττισμένων προγραμμάτων της Java και έχει να κάνει με την δομή του αρχείου class. Η τιμή του βρίσκεται στα headers του αρχείου και εξαρτάται από ποια έκδοσή του Java develpment kit (jdk) έχει γίνει η μεταγλώττιση.

Τα class version ορίζονται ως εξής:

J2SE 7 = 51 (0x33 hex)

J2SE 6.0 = 50 (0x32 hex)

J2SE 5.0 = 49 (0x31 hex)

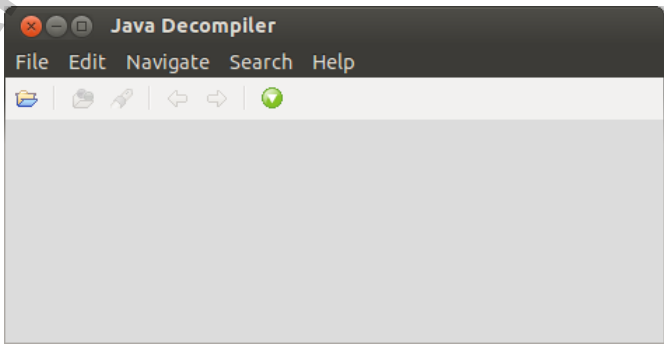
JDK 1.4 = 48 (0x30 hex)

JDK 1.3 = 47 (0x2F hex)

Το JAD υποστηρίζει την δυνατότητα εξαγωγής του πηγαίου κώδικα, μέχρι 47.0 class file version. Στην περίπτωση της εφαρμογής de.mehrmannd.sdbooster-GAMEX δεν έχει εφαρμογή λόγω του ότι το class file version είναι level 50.

Jd-gui [55]

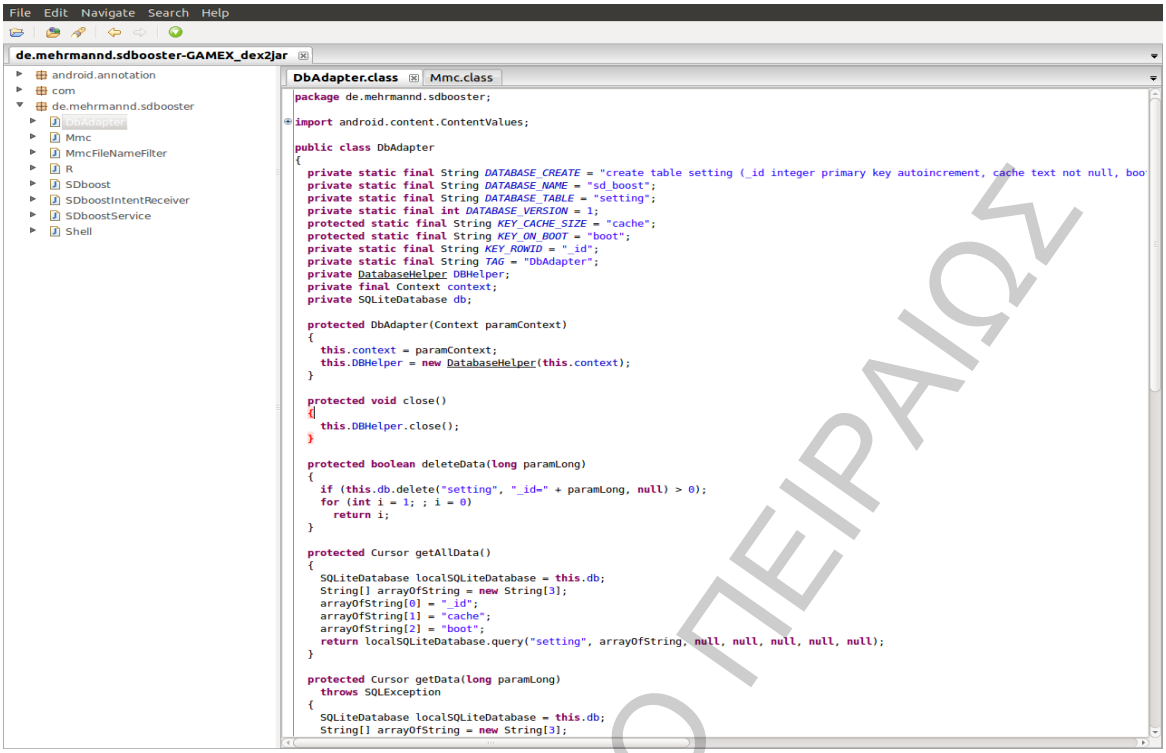
χρήση: `./jd-gui`



Εικόνα 36, Η αρχική οθόνη του Java Decompiler

35 http://en.wikipedia.org/wiki/Java_class_file

Δέχεται ως όρισμα .class αρχεία και ανοίγει το γραφικό περιβάλλον κα επιτυχώς από-μεταγλωττίζει το class αρχείο στον πηγαίο κώδικα:

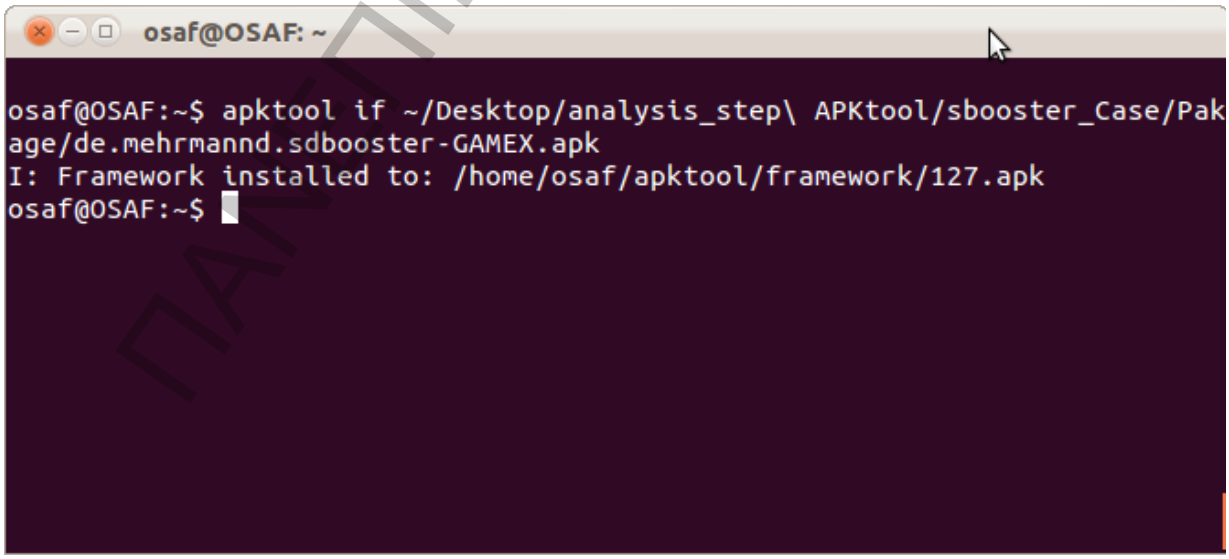


Εικόνα 37, Ο εξαγόμενος πηγαίος κώδικας της κλάσης DbAdapter

6.3.3 Βήμα 3ο: Χρήση πλατφόρμας ανάλυσης

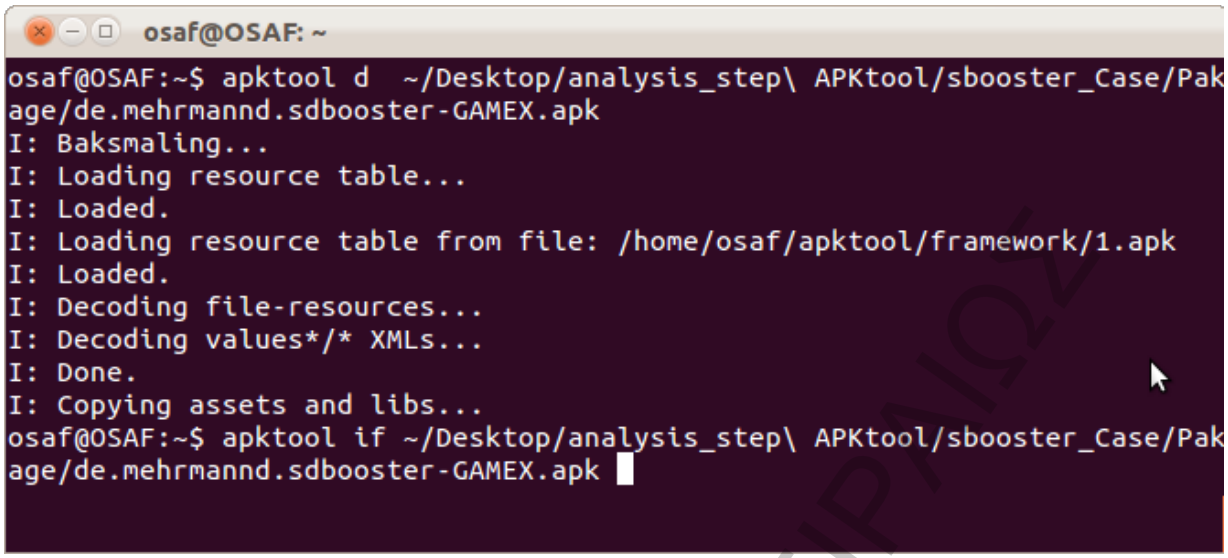
ApkTool [73]

Ορισμός του ανάλογου framework ανάλογα με το αρχείου εισόδου:



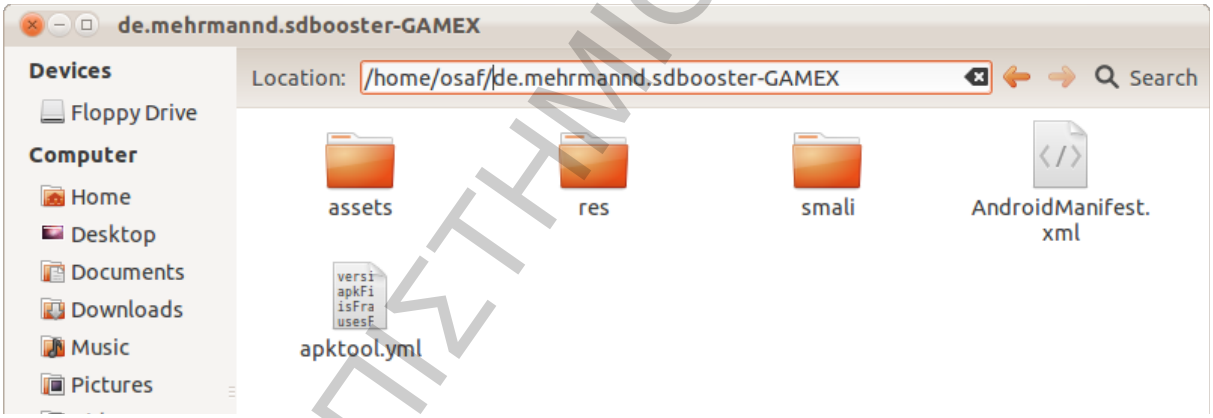
Εικόνα 38, Δημιουργία του framework (127.apk)

Μετά ακολουθεί η από-μεταγλώττιση για την παραγωγή του πηγαίου κώδικα από το εκτελέσιμο:

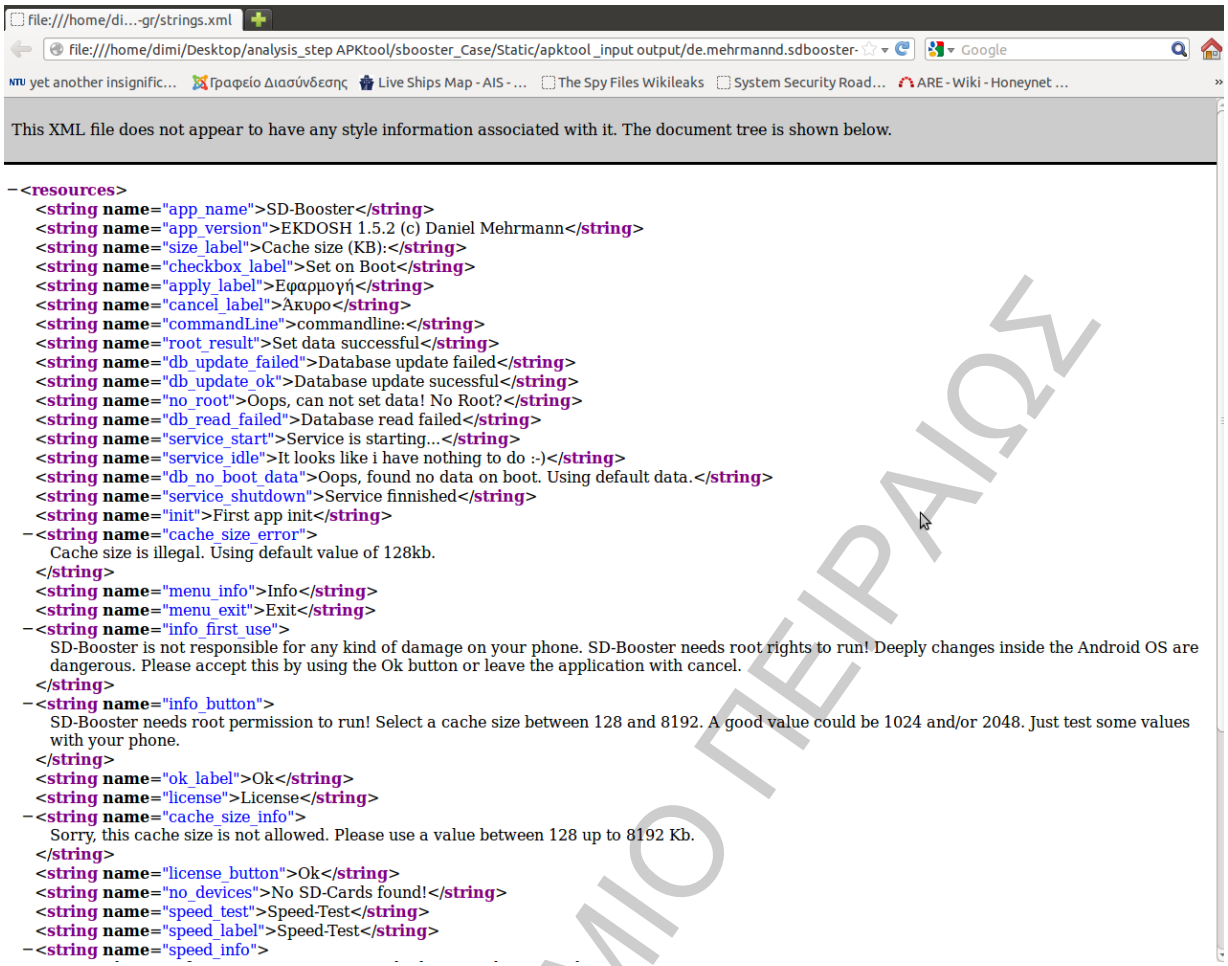


Εικόνα 39, Η από-μεταγλώττιση (decompilation)

Η εφαρμογή έχει αποσυναρμολογηθεί στον αντίστοιχο φάκελο :

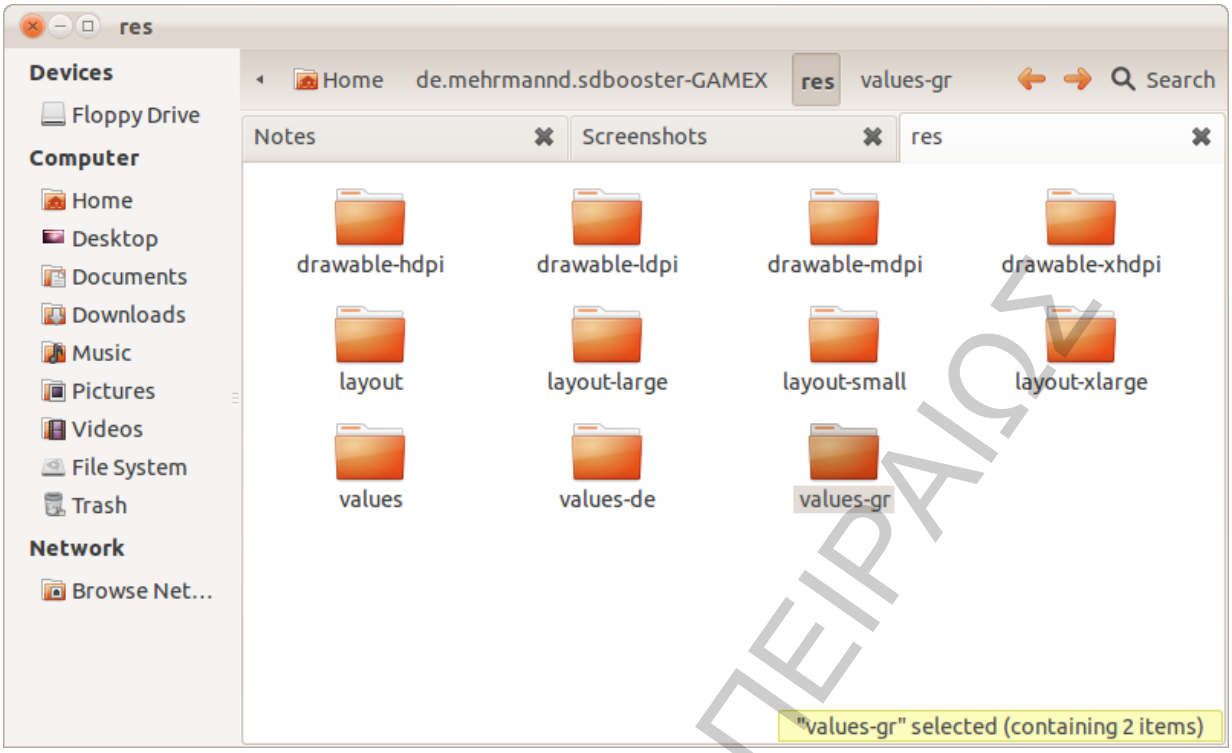


Εικόνα 40, Τα περιεχόμενα της εφαρμογής



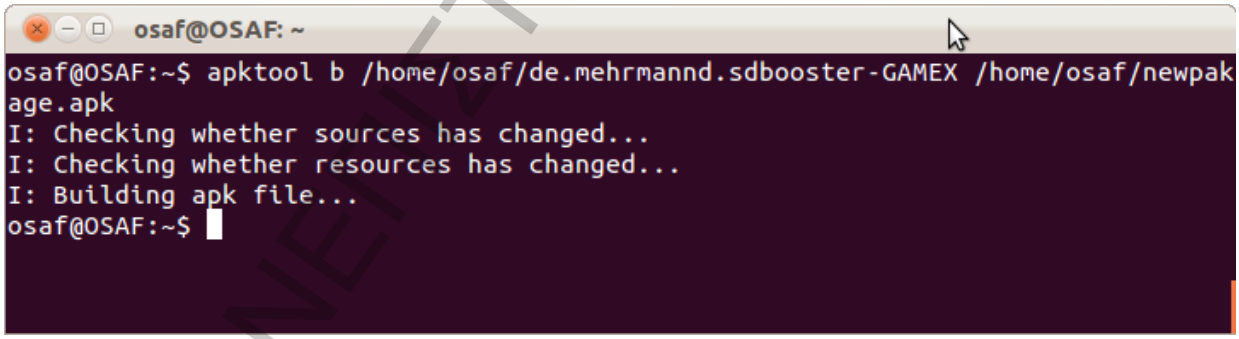
Εικόνα 41. Το ευανάννωστο AndroidManifest.xml

Τα περιεχόμενα μπορούν να υποστούν οποιαδήποτε αλλαγή:

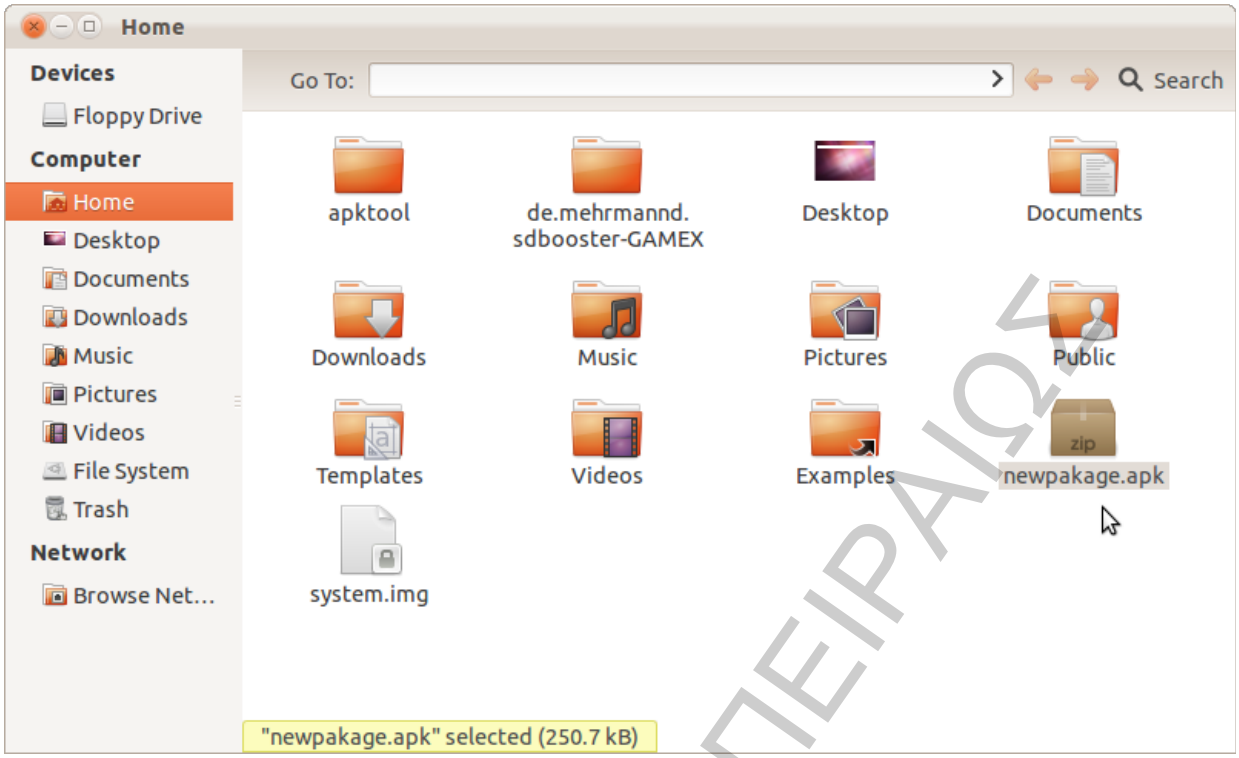


Εικόνα 42, Προσθήκη του φακέλου values-gr που θα μπορούσε να περιέχει το xml με όλες τις συμβολοσειρές στην ελληνική γλώσσα

Η λειτουργικότητα του apktool ολοκληρώνεται με την ανασυσκευασία του πακέτου:



Εικόνα 43, Ανασυσκευασία των αρχείων σε πακέτο apk



Εικόνα 44, Το νέο πακέτο newpakage.apk

6.3.3.1 APKinspector – Androguard

Εικόνα 45. Το APKinspector

Το APKinspector αποτελεί ένα ολοκληρωμένο πακέτο λειτουργιών στατικής ανάλυσης με γραφική διεπαφή και περιλαμβάνει τα εξής:

Γραφική διεπαφή παρουσίασης του διαγράμματος ροής του κώδικα (CFG[36]) (Εικόνες 49 - 51).

36 Ακρώνυμο του Control flow graph

Συνδέσμους από το διάγραμμα ροής στον πηγαίο κώδικα.

Λίστα μεθόδων ανά αντικείμενο και γενική λίστα μεθόδων. (Εικόνα 49)

Λίστα σημβολοσειρών

Διάγραμμα εισόδου/εξόδου για ένα δοθέν σημείο. (Εικόνα 55)

Σαν διεπαφή περιέχει δύο κύριες οθόνες το mainView και το sideView.

Το mainView περιλαμβάνει τις προβολές:

Του CFG σε επίπεδο μεθόδου

Του κώδικα smali μίας μεθόδου

Του κώδικα Java (όσο αυτό είναι εφυκτό) (παράδειγμα: Εικόνα 56)

Του bytecode μίας μεθόδου (Εικόνα 53)

Τις πληροφορίες που σχετίζονται με το πακέτο APK

Το sideView περιλαμβάνει:

-Το Files, που έχει όλα τα αρχεία της εφαρμογής (διάγραμμα αρχείων, κλάσεων, εικόνων).

-Το Strings, που έχει όλες τις συμβολοσειρές από το strings.xml και ο αναλυτής μπορεί να επεξεργαστεί αυτές τις τιμές για ευκολία κατανόησης.

-Το Classes που περιέχει το διάγραμμα των κλάσεων σε δενδροειδή μορφή.

-Το Methods που περιέχει όλες τις μεθόδους με κάποια στοιχεία εισόδου εξόδου

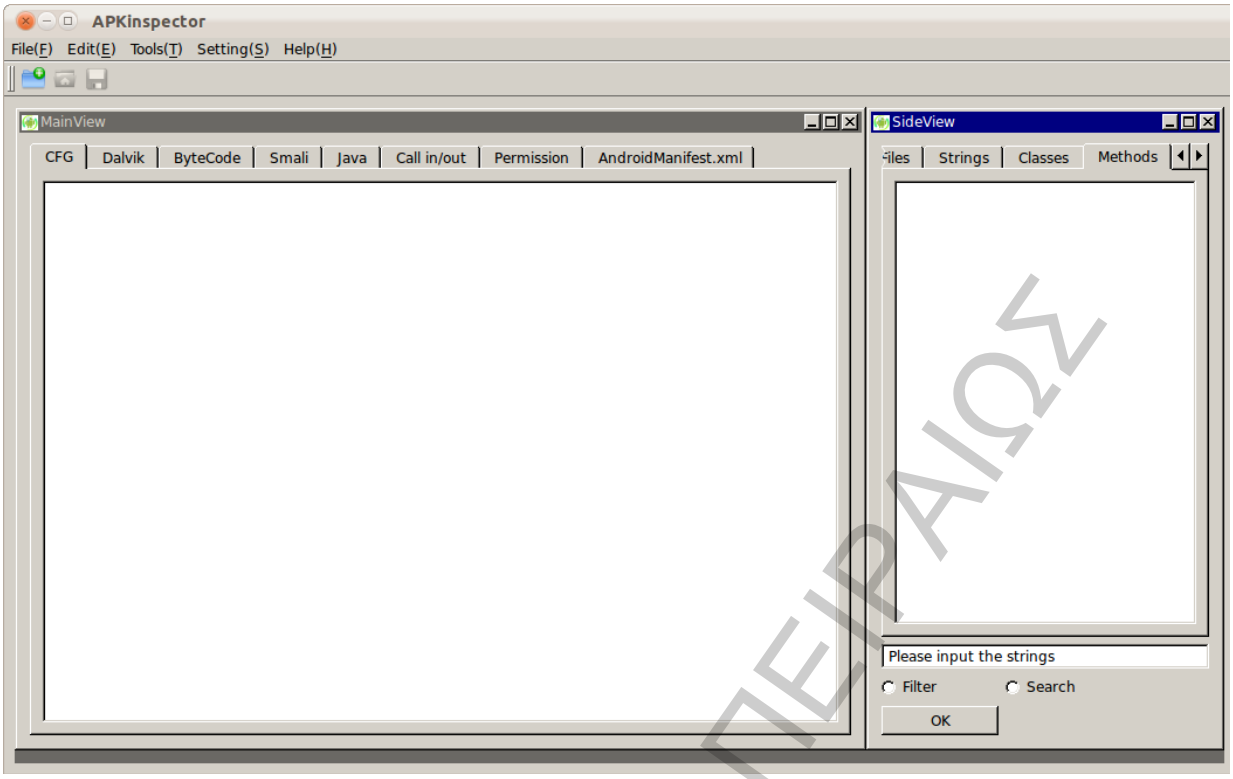
- Το Androidmanfiest.xml που περεχεί το ομώνυμο αρχείο.

-Το APKInfo που περιέχει γενικές πληροφορίες για την εφαρμογή, τι servises ξεκινάνε και τι δικαιώματα αιτούνται. Ουσιαστικά αποτελεί μία σύνοψη του Androidmanfiest.xml.

- Ο κώδικας smali για μία επιλεγμένη μέθοδο

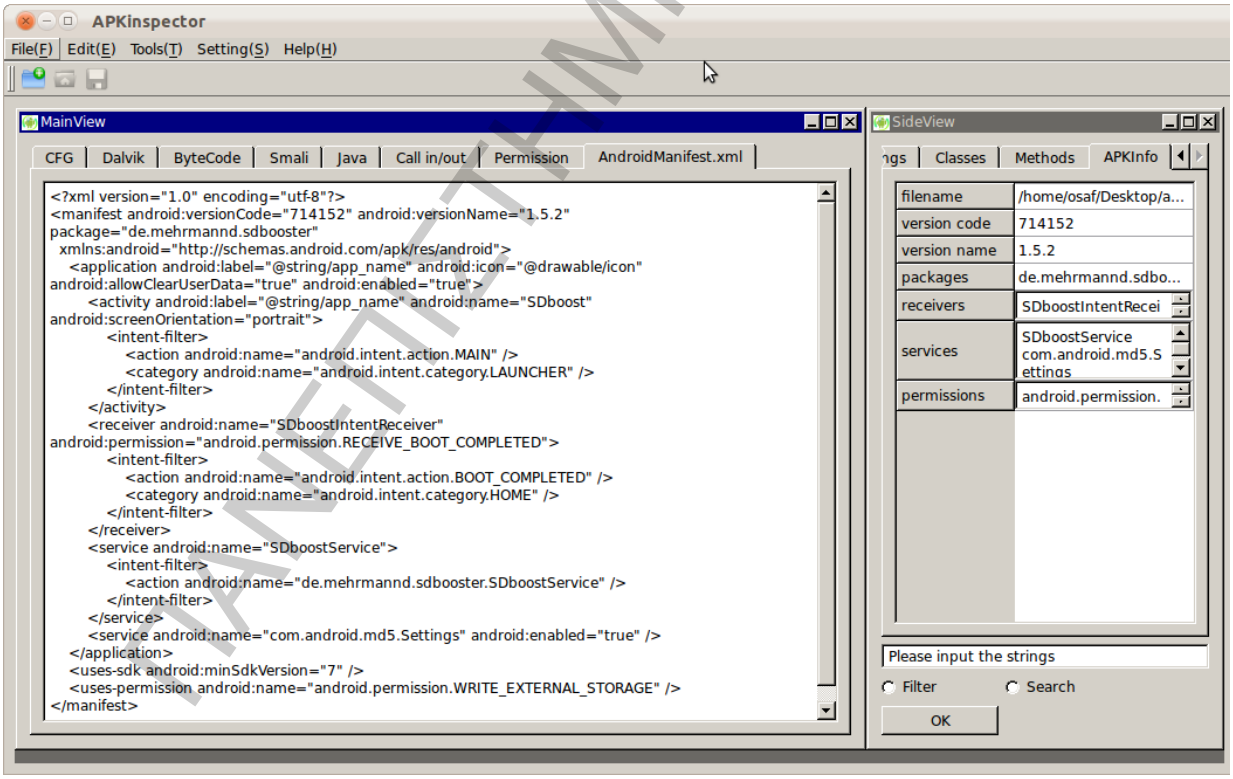
- Ο κώδικας Java για ένα επιλεγμένο αρχείο .java

Στις ακόλουθες εικόνες (46 – 56) παρουσιάζεται ενδεικτικά η λειτουργικότητα του APKinspector:

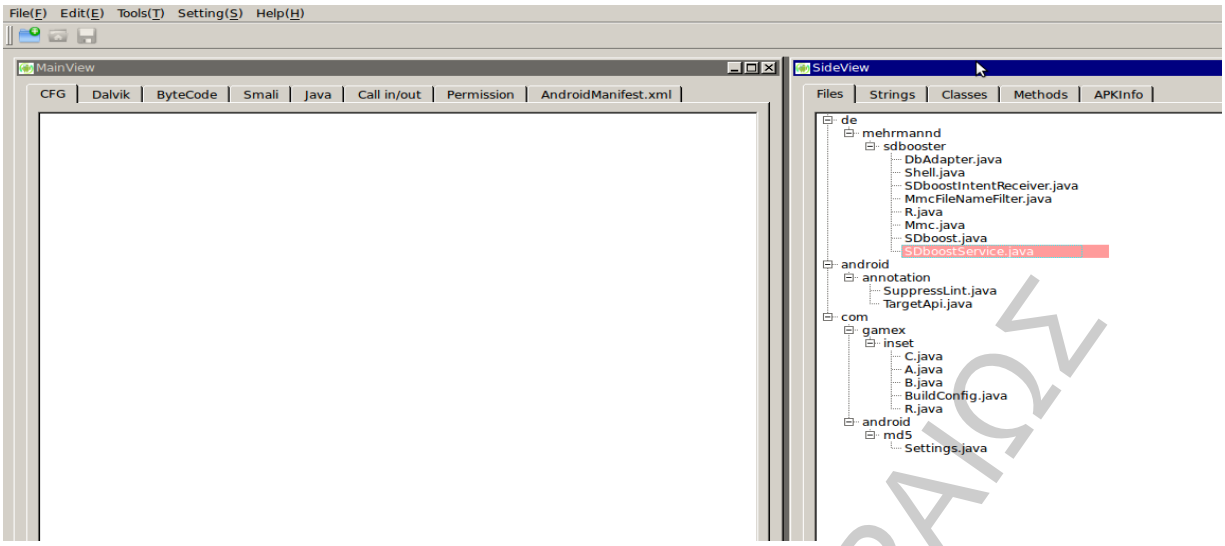


Εικόνα 46, Η αρχική οθόνη του APKinspector

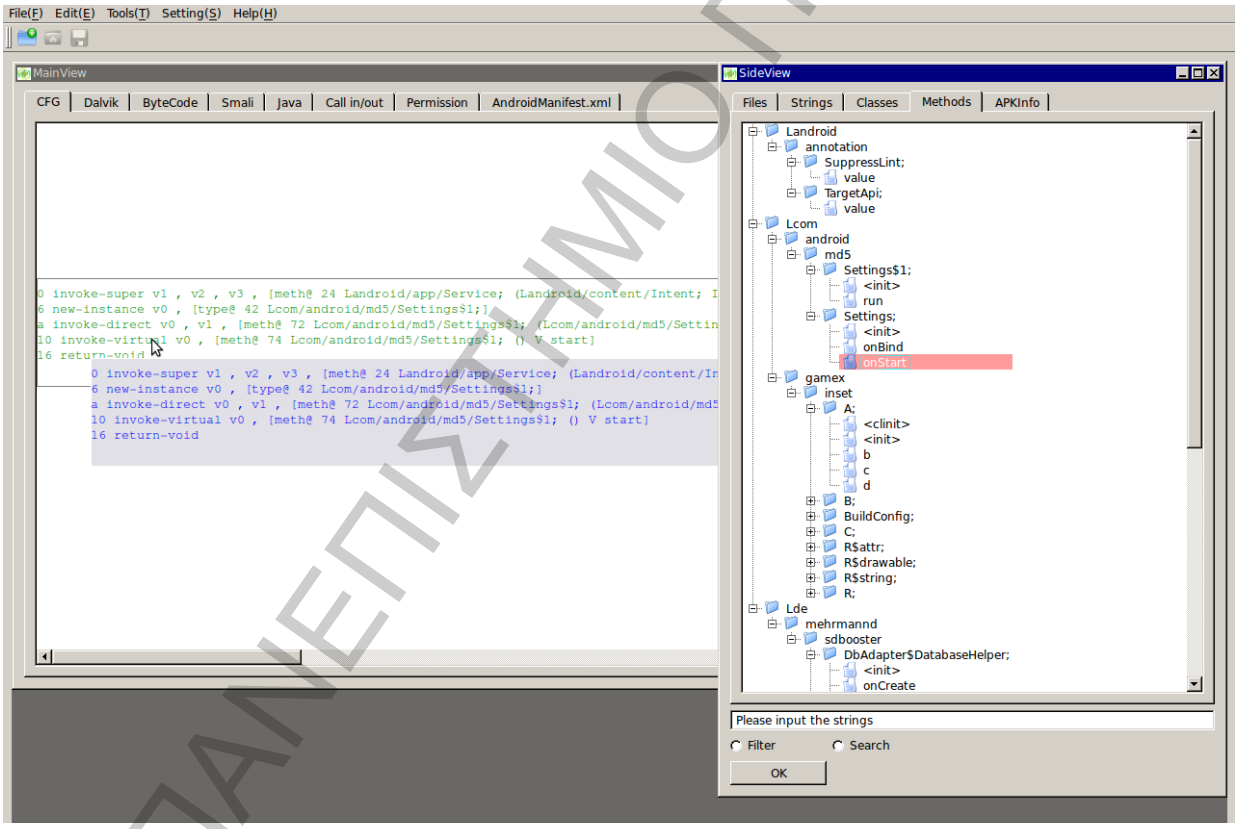
Επιλέγεται το αρχείο apk προς ανάλυση.



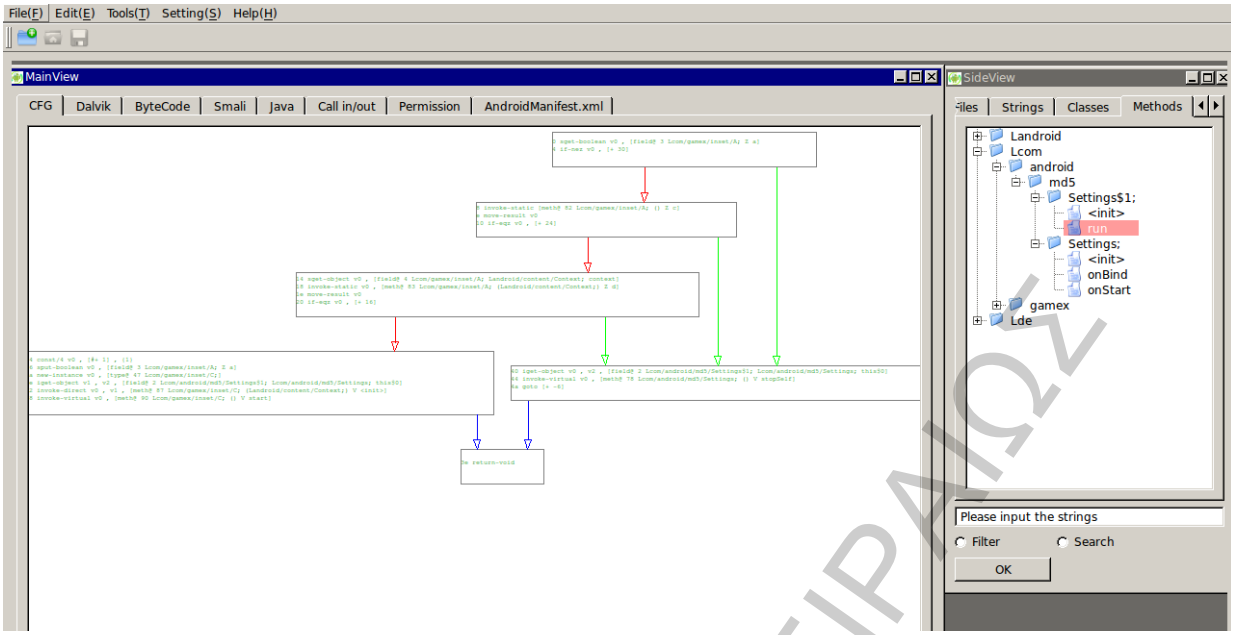
Εικόνα 47, Η προβολή του Androidmanfiest.xml σε αναγνώσιμη μορφή και το APKinfo (SideView)



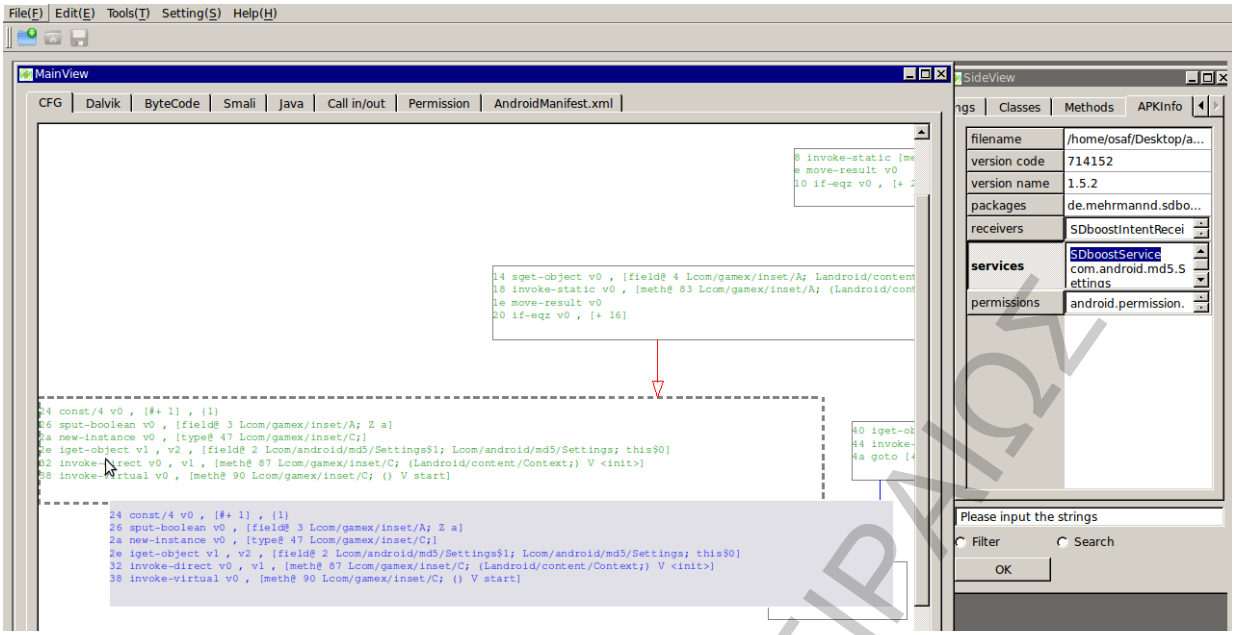
Εικόνα 48, Το δεντροδιάγραμμα των αρχείων του πακέτου της εφαρμογής



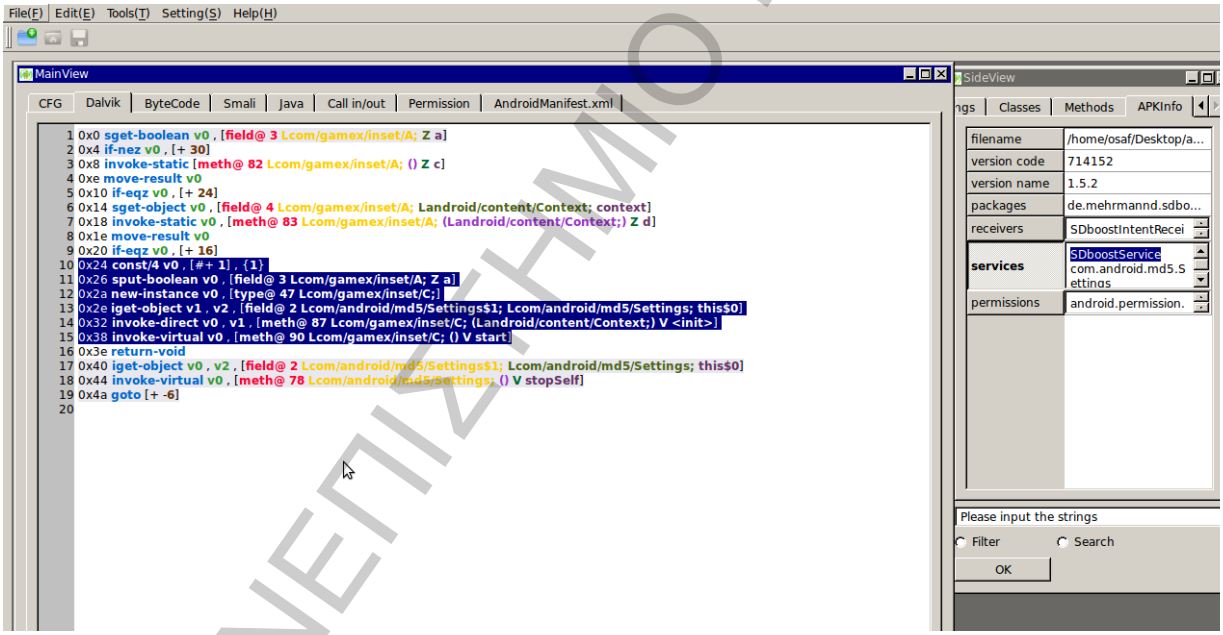
Εικόνα 49, Το διάγραμμα ροής μίας μεθόδου



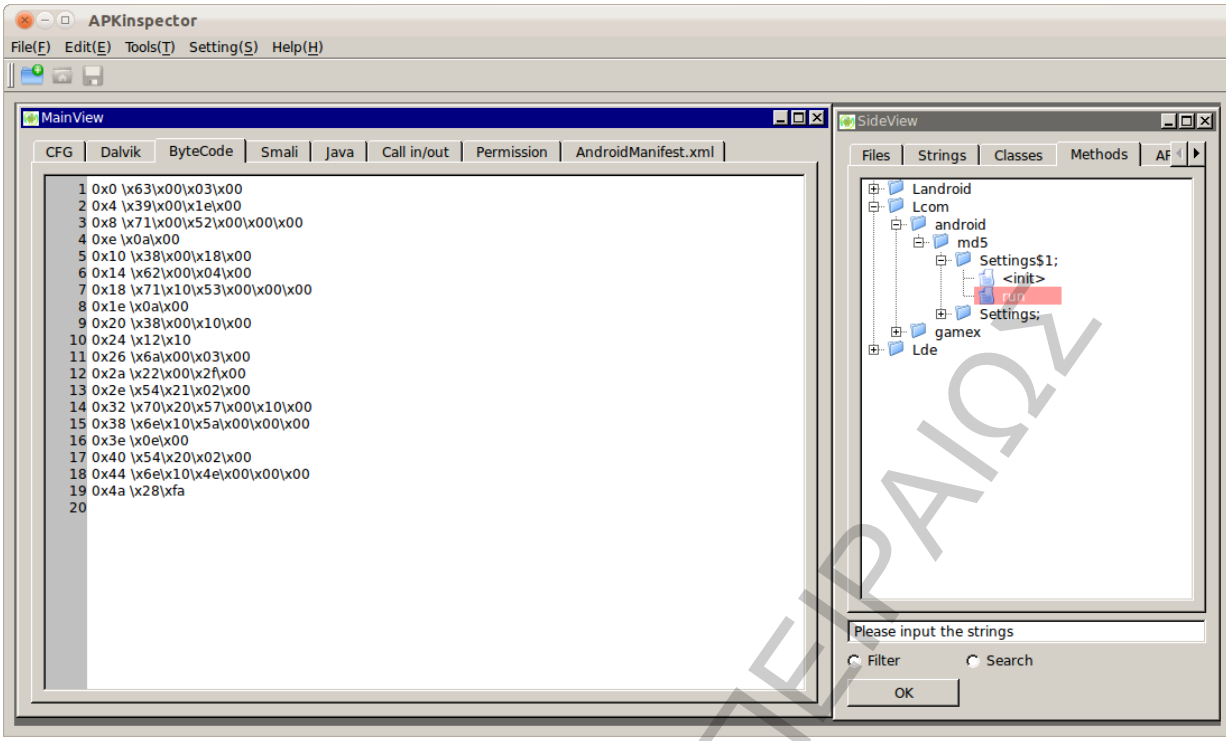
Εικόνα 50, Το διάγραμμα ροής της αρχικής μεθόδου



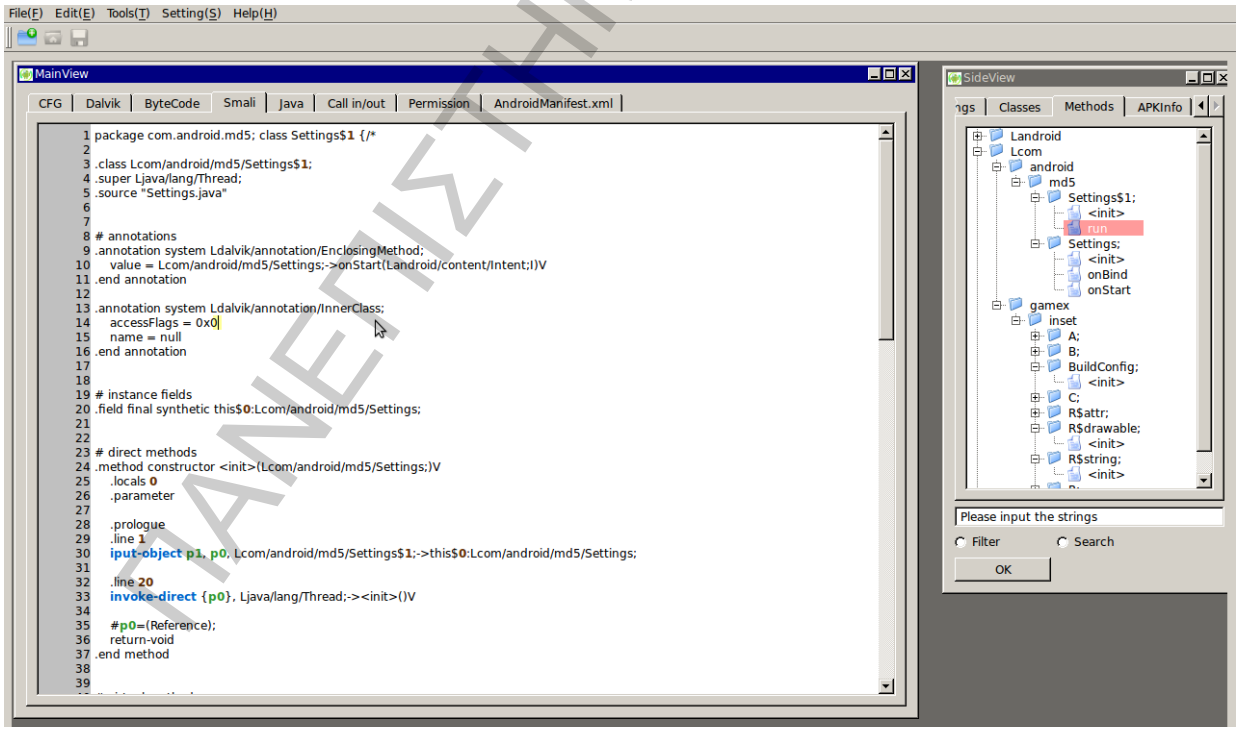
Εικόνα 51, Επισκόπηση της μεθόδου, παρατιέται ότι ξεκινάει κάποια services



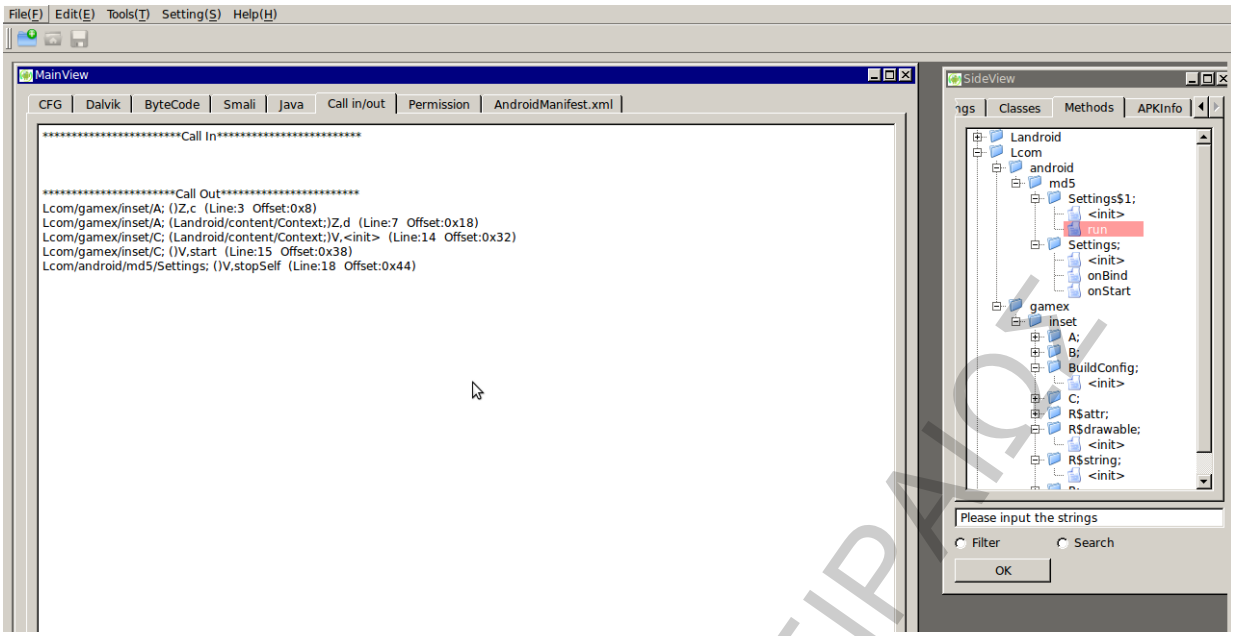
Εικόνα 52, ο κώδικας που ξεκινάει το ύποπτο service σε ψευδογλώσσα μηχανής Dalvik



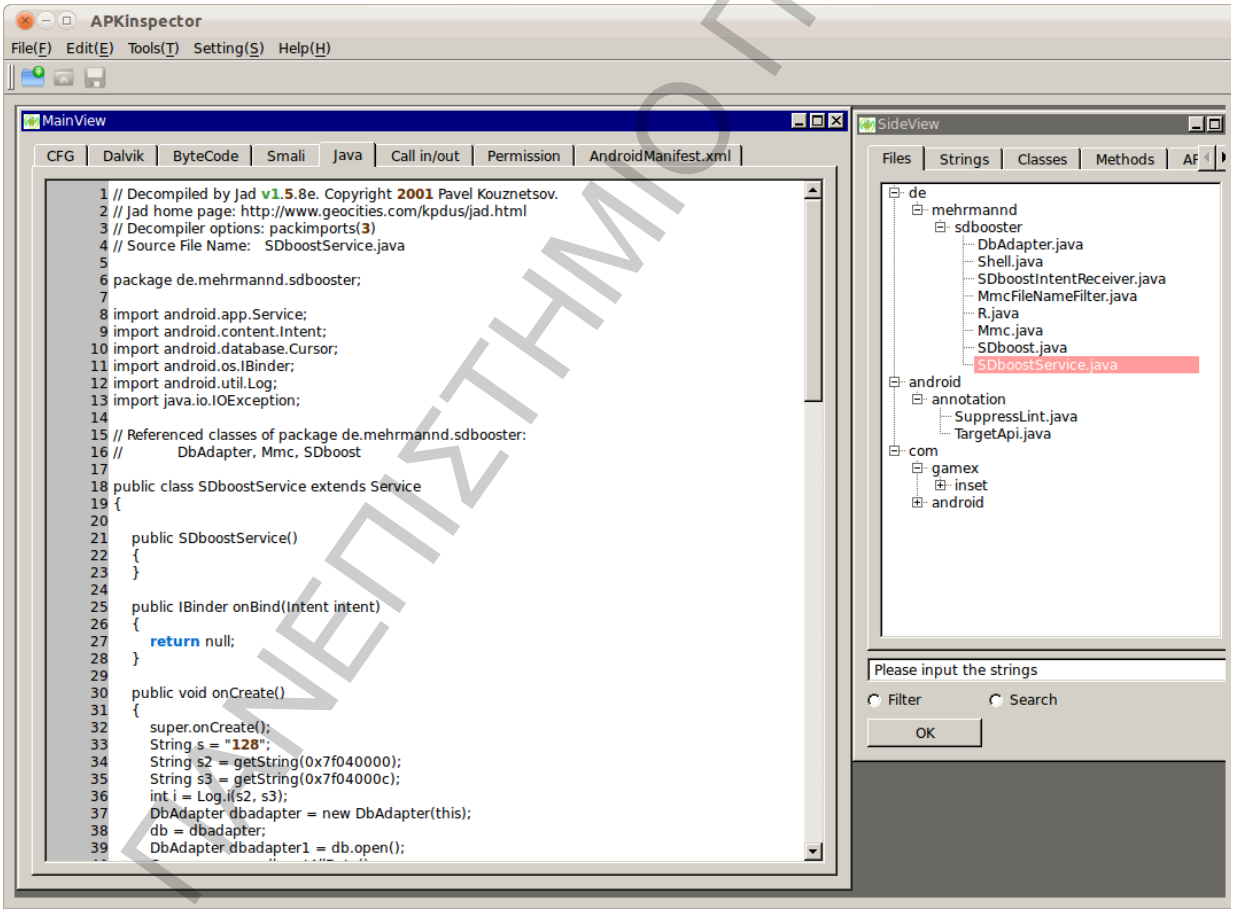
Εικόνα 53, Το Bytecode μίας μεθόδου



Εικόνα 54, Η μέθοδος σε μορφή smali



Εικόνα 55, Οι είσοδοι και οι έξοδοι της μεθόδου



Εικόνα 56, Ο Κώδικας java της μεθόδου

6.3.4 Βήμα 4ο: Ανάλυση ευρημάτων

Σε αυτό το σημείο ο αναλυτής πρέπει:

-Να εντοπίσει την χρήση των δικαιωμάτων που δηλωθήκαν στο Andoidmanifest.xml στον κώδικα Java.

- Να ελέγξει τις μεθόδους που χρησιμοποιούν τα δικαιώματα ως προς την λειτουργία τους και τον σκοπό τους.

- Να εντοπίσει, με βάση τον κώδικα που αποθηκεύονται τα δεδομένα, στην συσκευή στην κάρτα μνήμης ή κάπου αλλού. Ή εάν μετατρέπονται κάποια δεδομένα σε μορφή πίνακα και αποθηκεύονται με αυτή την μορφή.

-Τέλος, βάση των στοιχείων που έχει συλλέξει σε αυτό το σημείο, ο αναλυτής οφείλει να αναπτύξει μία υπόθεση για το τι κακόβουλο πιστεύει ότι μπορεί να εκτελεί η υπό-ανάλυση εφαρμογή. Η υπόθεση αυτή θα αποτελέσει βάση της δυναμικής ανάλυσης.

6.4 Στατική ανάλυση: Βέλτιστη πρακτική

1) Ανάκτηση του αρχείου .apk της ύποπτης εφαρμογής.

2) δημιουργία του φακέλου “όνομα_εφαρμογής Case”.

3) Μετατροπή των αρχείων με την χρήση dex2jar.

4) Ανάλυση και επεξεργασία του apk με την χρήση του APK Inspector

5) Με βάσει τις εξαγόμενες πληροφορίες από το APK Inspector ο αναλυτής επεξεργάζεται το .jar (από το dex2jar) με το JD GUI.

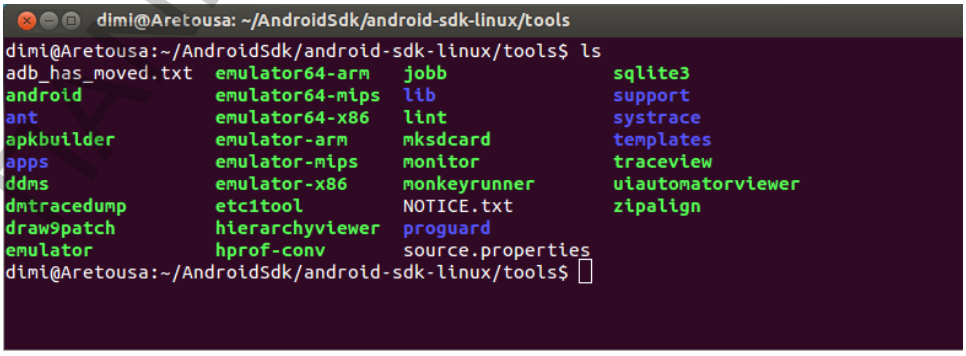
6) Ανάλυση ευρημάτων, παραγωγή υπόθεσης

6.5 Δυναμική ανάλυση

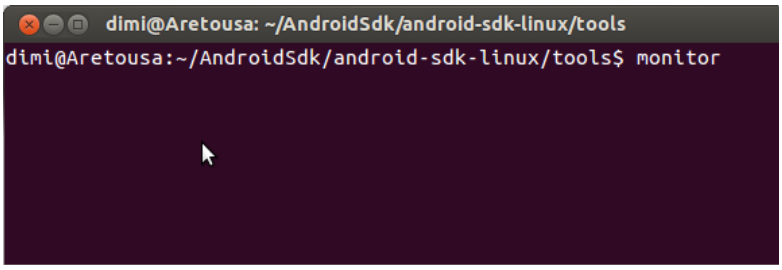
παρουσίαση εργαλείων :

6.5.1 Android SDK

Στην περίπτωση της ανάλυσης μίας εφαρμογής το **Android SDK,** μπορεί να χρησιμοποιηθεί για δυναμική ανάλυση [81], [82]

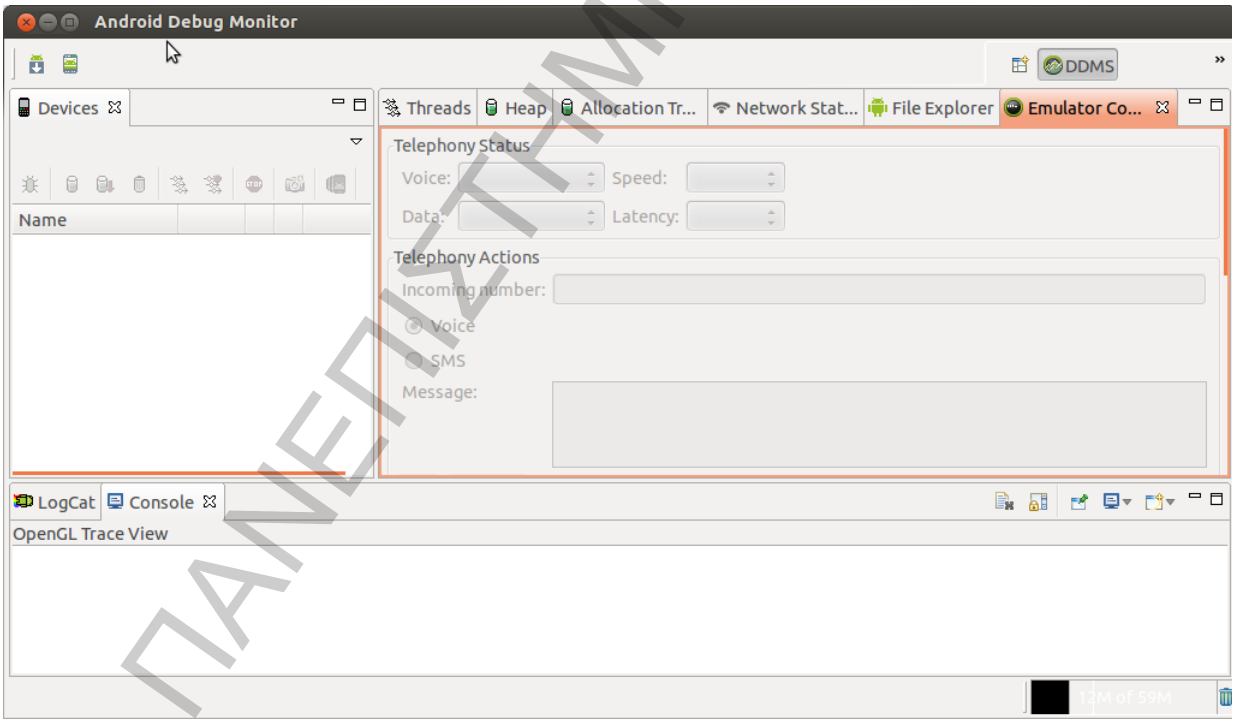


Εικόνα 57, Ο home φάκελος του Android SDK

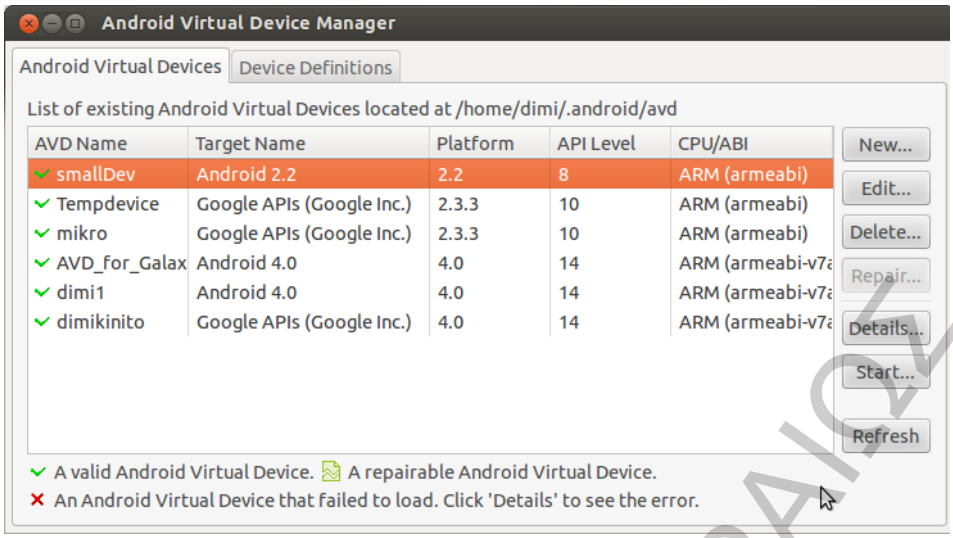


Εικόνα 58, Εκκίνηση του Android Device Monitor

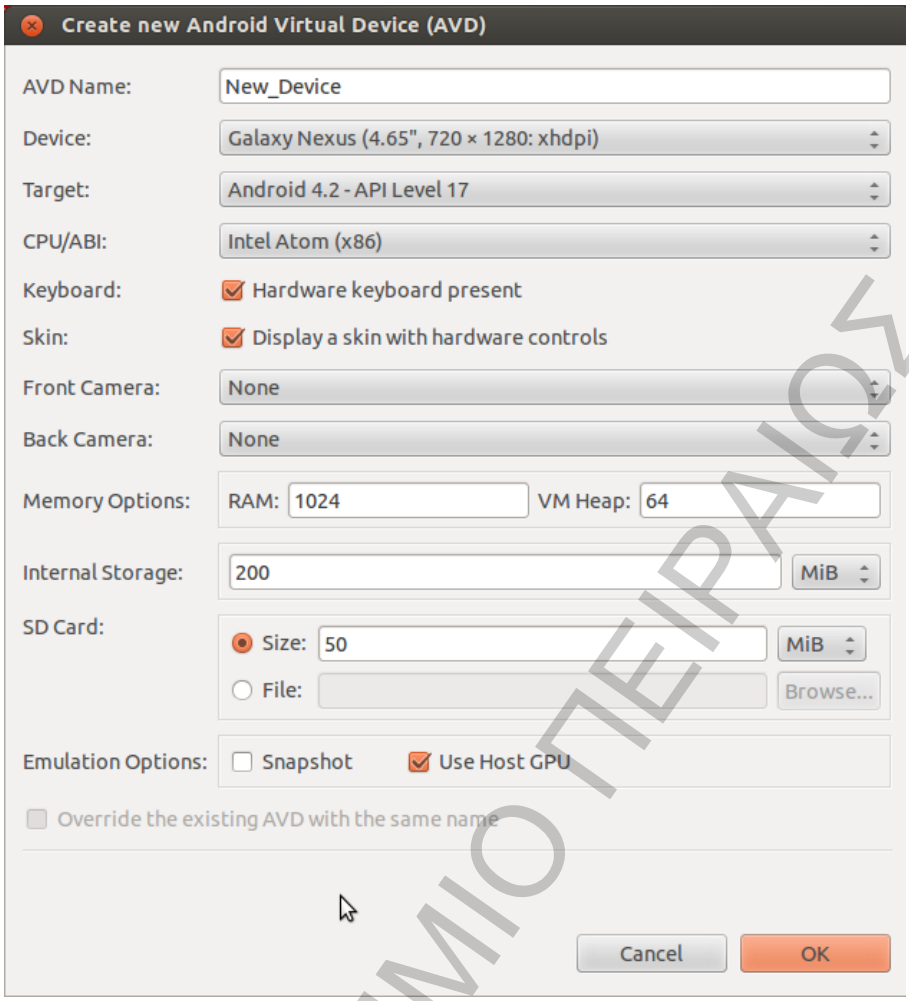
Εικόνα 59, Android Device Monitor Logo



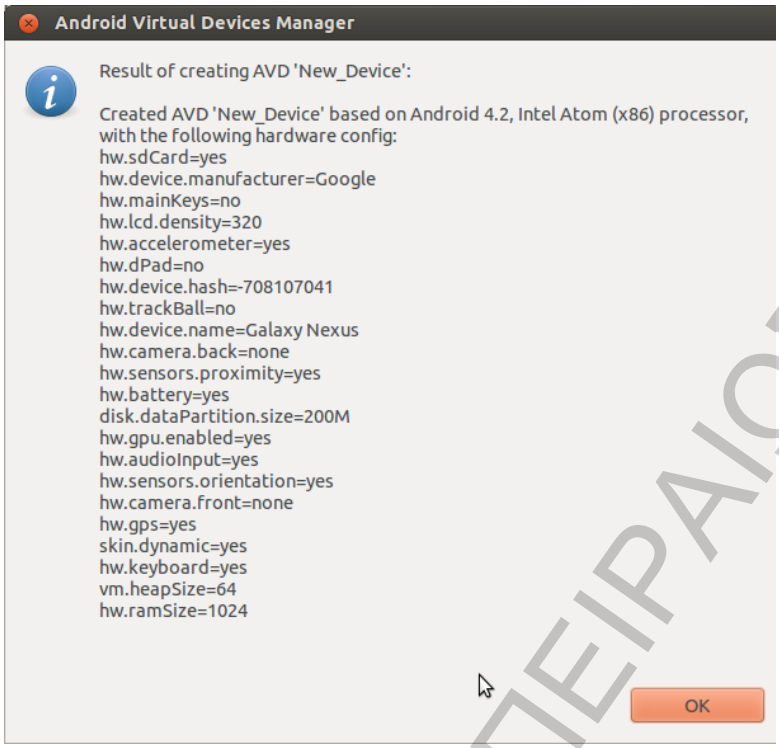
Εικόνα 60, Η αρχική οθόνη του Android Device Monitor



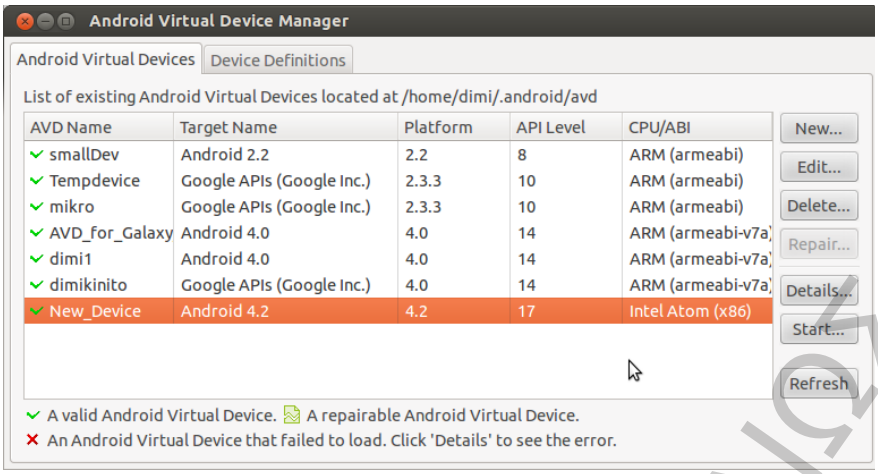
Εικόνα 61, Ο Android Virtual Device Manager



Εικόνα 62, Δημιουργία νέας εικονικής συσκευής με το όνομα New Device



Εικόνα 63. Τα στοιχεία της νέας εικονικής συσκεύης



Εικόνα 64, Εικόνα 55, Η εικονική συσκευή New_Device έχει δημιουργηθεί

6.5.1.1 Android Debug Bridge (adb)

```
device commands:
  adb push <local> <remote>    - copy file/dir to device
  adb pull <remote> [<local>]  - copy file/dir from device
  adb sync [ <directory> ]     - copy host->device only if changed
                                 (-l means list but don't copy)
                                 (see 'adb help all')
  adb shell                    - run remote shell interactively
  adb shell <command>          - run remote shell command
  adb emu <command>            - run emulator console command
  adb logcat [ <filter-spec> ] - View device log
  adb forward <local> <remote> - forward socket connections
                                 forward specs are one of:
                                   tcp:<port>
                                   localabstract:<unix domain socket name>
                                   localreserved:<unix domain socket name>
                                   localfilesystem:<unix domain socket name>
                                   dev:<character device name>
                                   jdwp:<process pid> (remote only)
  adb jdwp                     - list PIDs of processes hosting a JDWP transport
  adb install [-l] [-r] [-s] <file> - push this package file to the device and install it
                                 ('-l' means forward-lock the app)
                                 ('-r' means reinstall the app, keeping its data)
                                 ('-s' means install on SD card instead of internal storage)
  adb uninstall [-k] <package> - remove this app package from the device
                                 ('-k' means keep the data and cache directories)
  adb bugreport                - return all information from the device
                                 that should be included in a bug report.

  adb help                     - show this help message
  adb version                  - show version num
```

Εικόνα 65. Η λίστα εντολών του ADB

Το Android Debug Bridge (ADB) [74], [75] είναι ένα ευέλικτο command line εργαλείο που επιτρέπει στον προγραμματιστή να επικοινωνεί με έναν εξομοιωτή ή και με μία συνδεδεμένη Android συσκευή. Είναι ένα πρόγραμμα client-server που περιλαμβάνει τρία συστατικά: 1. Έναν client, που εκτελείται στον υπολογιστή ανάπτυξης της εφαρμογής. Με αυτόν τον client μπορεί ο προγραμματιστής να αλληλεπιδράσει με την συσκευή (εικονική ή όχι) και να εκτελέσει διάφορες διαδικασίες στην συσκευή (πχ ADT plugin ή/και DDMS).

2. Έναν server, που τρέχει σαν μία διεργασία παρασκηνίου (background process) στον υπολογιστή ανάπτυξης της εφαρμογής. Ο server αυτός διαχειρίζεται την επικοινωνία μεταξύ του client και της διεργασίας ADB που εκτελείται σε έναν εξομοιωτή ή τη συσκευή. 3. Μία διεργασία (ADB), που τρέχει σαν ένα background process στον κάθε εξομοιωτή ή τη συσκευή που είναι συνδεδεμένη.

Το adb αποτελεί κομμάτι της πλατφόρμας Adnroid sdk, (<sdk path>/platform-tools/)

παραδείγματα χρήσης του adb:

Εγκατάσταση της εφαρμογής App (App.apk)

```
Μεταφέρεται το αρχείο App.apk στον home φάκελο του adb:

adb install App.apk
```

`adb install [-l] [-r] <file>` - Εκτελεί push του πακέτου apk στην συσκευή και το εγκαθιστά

('-l' Να κάνει lock την εφαρμογή)

('-r' Επανεγκατάσταση, διατηρώντας τα προηγούμενα δεδομένα)

Απεγκατάσταση της εφαρμογής App (App.apk)

Για την απεγκατάσταση δεν χρησιμοποιείται η υπάρχουσα εντολή uninstall, καθώς πάντα αποτυγχάνει. Αντ' αυτού χρησιμοποιείται το shell:

```
adb shell

# cd data/app

# ls

# rm –r App.apk

# ls
```

(Ctrl+c για να κλείσει το shell)

`adb uninstall [-k] <package>` - Αφαιρεί το πακέτο apk από την συσκευή

('-k' για την διατήρηση των φακέλων δεδομένων και της cache)

Εντολές Push & Pull

Οι εντολές Push και Pull χρησιμοποιούνται για την αντιγραφή αρχείων ή φακέλων από και προς τον προσομοιωτή/συσκευή.

Το Pull αντιγράφει από την συσκευή ή τον προσομοιωτή

Ενώ το Push αντιγράφει στην συσκευή ή τον προσομοιωτή, από το σύστημα.

Pull:

```
adb pull /data/app/aFile.txt
```

Για την αντιγραφή όλων των περιεχόμενων του φακέλου test

Ο test φάκελος (adb home)/test:

```
adb pull /data/app test
```

Push:

```
adb push appname.apk /data/app
```

Για την αντιγραφή όλων των περιεχόμενων του φακέλου test:

```
adb push test /data/app
```

Για να κλείσει ο adb:

```
adb kill-server (για τερματισμό του adb server)
exit
```

Λοιπές εντολές:

```
adb reboot - Για reboot της εικονικής συσκευής
adb shell <command> - Για απομακρυσμένη εντολή shell
adb emu <command> - Εκκινεί την κονσόλα της εικονικής συσκευής
adb logcat [ <filter-spec> ] – Προβολή του log της συσκευής
adb forward <local> <remote> - Προωθεί socket συνδέσεις
```

Η προώθηση μπορεί να αφορά:

```
tcp:<port>
localabstract:<unix domain socket name>
localreserved:<unix domain socket name>
localfilesystem:<unix domain socket name>
dev:<Όνομα συσκευής>
adb jdwp – Εμφανίζει την λίστα των PID ενός JDWP[37]
adb bugreport – Επιστρέφει όλες τις απαραίτητες πληροφορίες που μπορούν να περιληφθούν σε ένα bug report.
adb help – Εμφανίζει την λίστα των εντολών
adb version – Εμφανίζει τον αριθμό της έκδοσης του adb
```

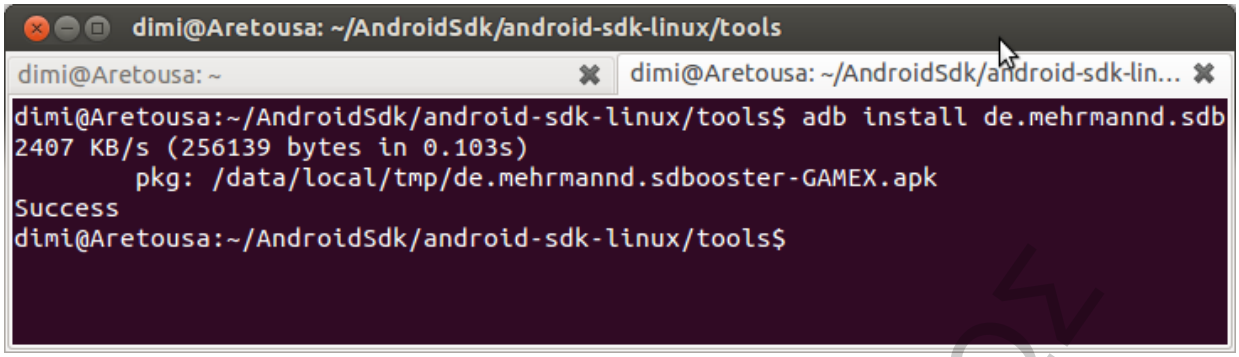
6.5.1.2 Εκτέλεση δειγμάτων στο Adnroid sdk

Για την εκτέλεση μίας εφαρμογής στο sdk χρειάζεται να ξεκινήσει μία εικονική συσκευή (Εικόνα 66) και να εγκατασταθεί (Εικόνα 67) σε αυτήν η ανάλογη εφαρμογή, μετά εκτελείται κανονικά όπως θα την εκτελούσε ένας χρήστης Εικόνες (68-69-70) και μόλις ο αναλυτής συλλέξει τις απαιτούμενες πληροφορίες που θέλει απεγκαθιστά την εφαρμογή (Εικόνα 71).

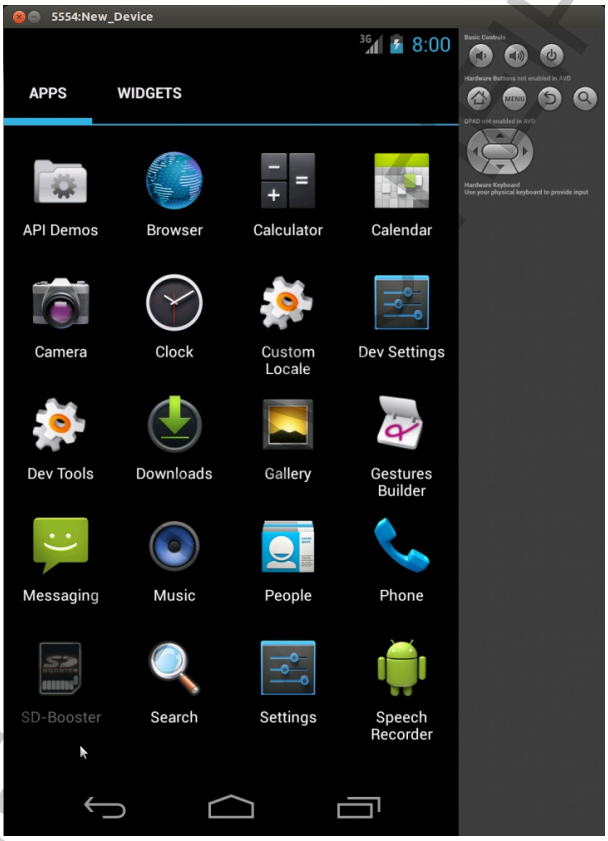
37 JavaTM Debug Wire Protocol http://docs.oracle.com/javase/1.5.0/docs/guide/jpda/jdwp-spec.html

Δείγμα 1 de.mehrmannd.sdbooster-GAMEX.apk

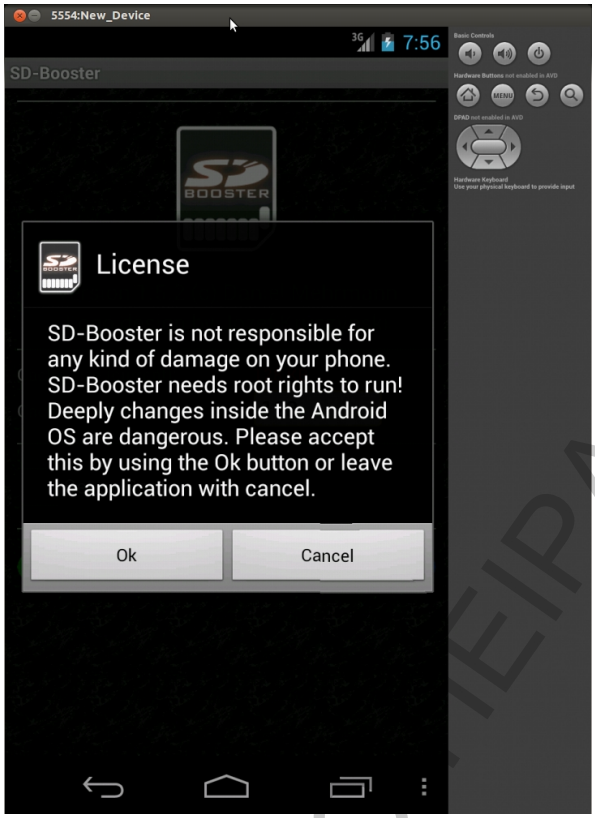
Εικόνα 66, Η εικονική συσκευή New_Device



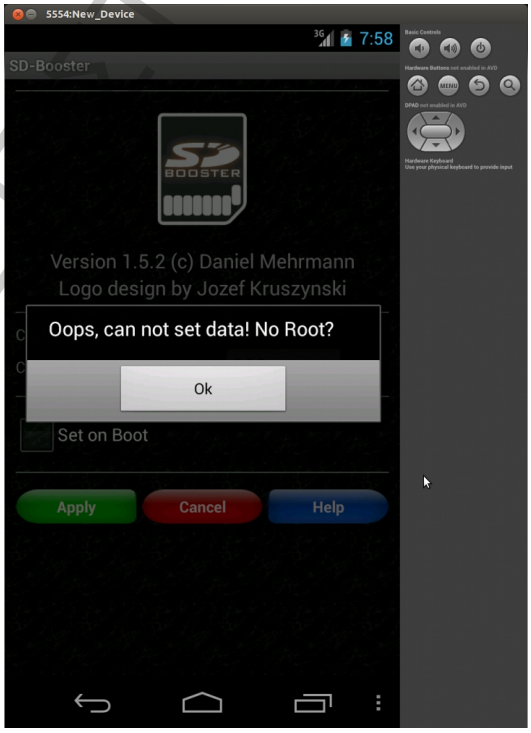
Εικόνα 67. Ενκατάσταση του ιού de.mehrmannd.sdbooster-GAMEX



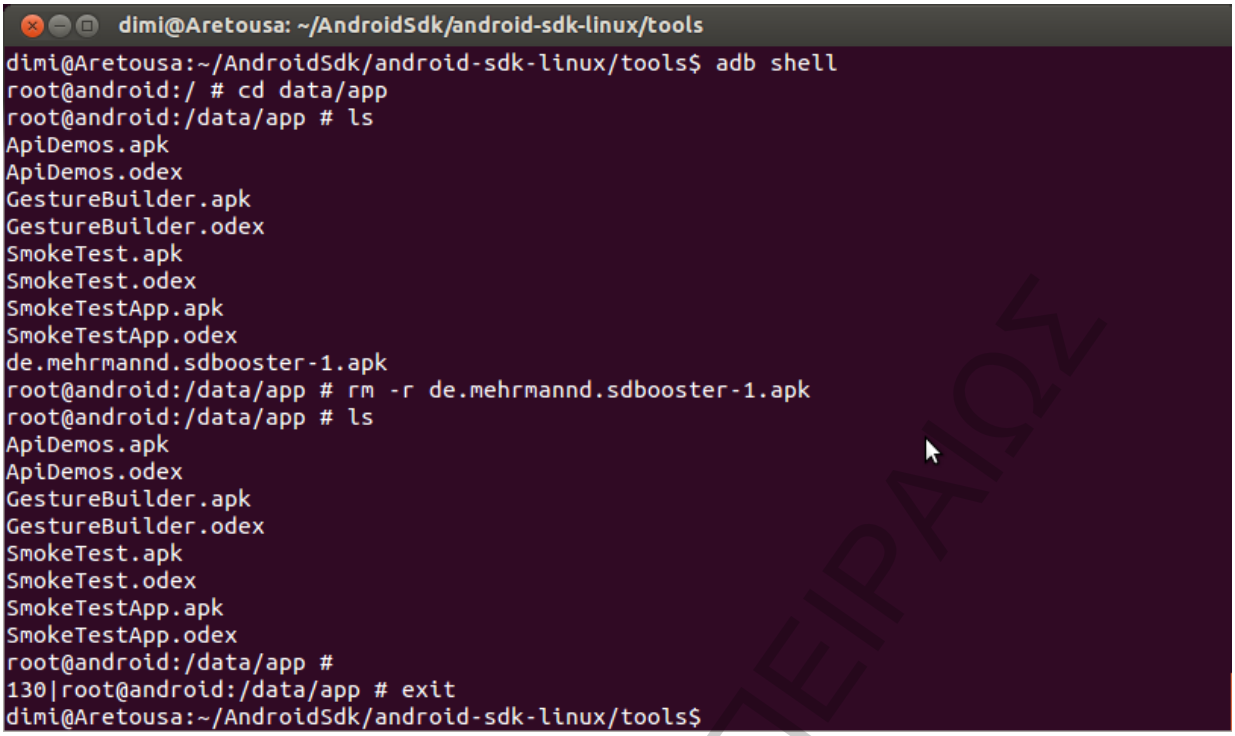
Εικόνα 68, Ο ιός εμφανίζεται σαν μία κανονική εφαρμογή με το όνομα SD-Booster



Εικόνα 69, Το συγκεκριμένο προειδοποιητικό μήνυμα αφορά την χορήγηση root δικαιωμάτων

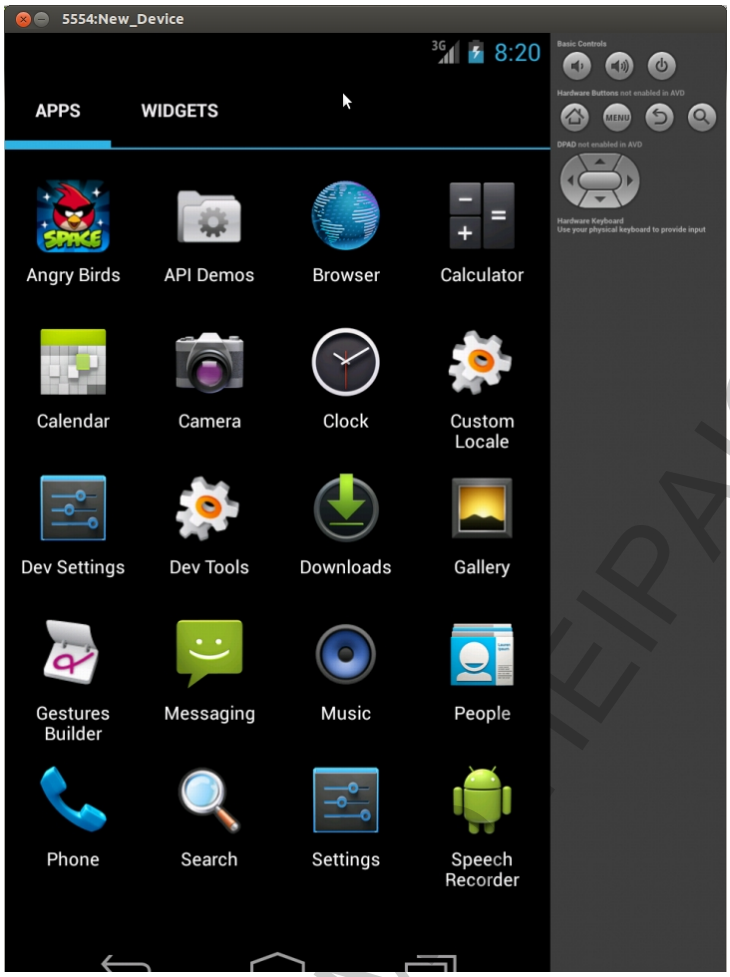


Εικόνα 70, Η εφαρμογή ενημερώνει τον χρήστη ότι δεν μπορεί να εκτελέσει την διαδικασία για την οποία προορίζεται



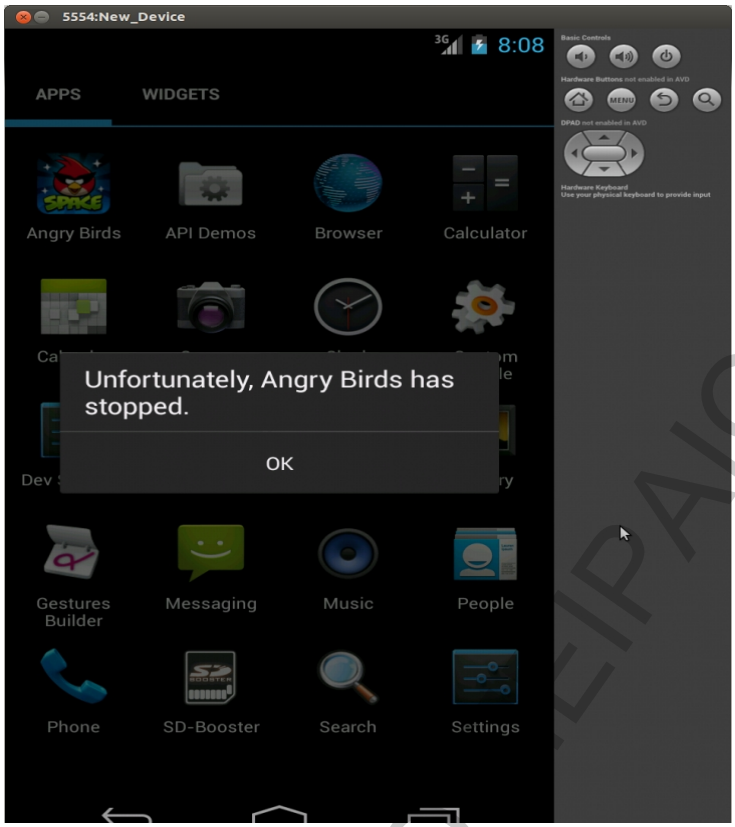
Εικόνα 71, Απεγκατάσταση της εφαρμογής

Δείγμα 2 com.rovio.new.ads-LeNa.c.apk



Εικόνα 72, Το com.rovio.new.ads-LeNa αποτελεί απομίμηση της εφαρμογής Angry Birds

Εικόνα 73, Δοκιμή του δείγματος και στην έκδοση Android 2.2

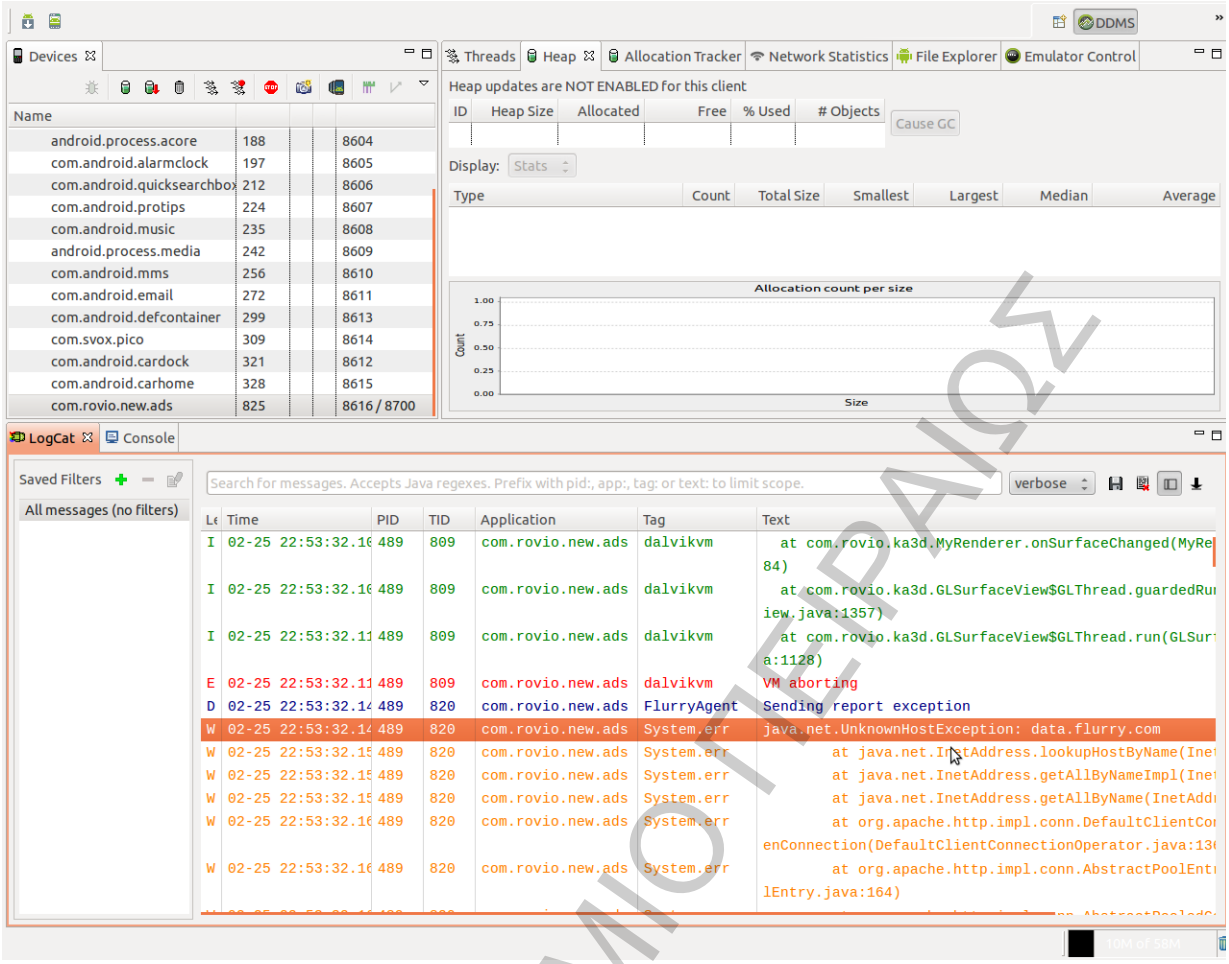


Εικόνα 74, Η εφαρμογή δεν εκτελείται

Το γεγονός ότι η εφαρμογή δεν εκτελείται (Εικόνα 74) μπορεί να σημαίνει ότι :

α) Το malware αναγνώρισε ότι εκτελέστηκε σε εικονικό περιβάλλον και τερματίστηκε για να μην εντοπιστεί.

β) Είτε ότι εκτελέστηκε το κακόβουλο φορτίο και δεν υφίσταται η λειτουργικότητα της εφαρμογής Angry Birds.



Εικόνα 75, Ο λόγος που τερματίστηκε η εφαρμογή

6.5.2 Ανάλυση με DroidBox

Εικόνα 76, DroidBox logo

6.5.2.1 API MONITOR

Το API MONITOR [76],[77] αποτελεί μία συνοδευτική εφαρμογή του DroidBox, η οποία ανασκευάζει τα πακέτα apk ώστε να επιδέχονται ανάλυση ανεξαρτήτως με πιο Android API είναι κατασκευασμένα (έχουν γίνει compile). Ο λόγος που δημιουργήθηκε αυτή η συνοδευτική εφαρμογή είναι ότι η πλατφόρμα Android εξελίσσεται ταχύτατα με αποτέλεσμα να διανέμονται νεώτερες εκδόσεις API ανά τακτά χρονικά διαστήματα. Το αρχικό DroidBox για να μπορεί να εξετάζει την συμπεριφορά του εκάστοτε apk και των API logs ρυθμιζόταν για ανάλυση ενός

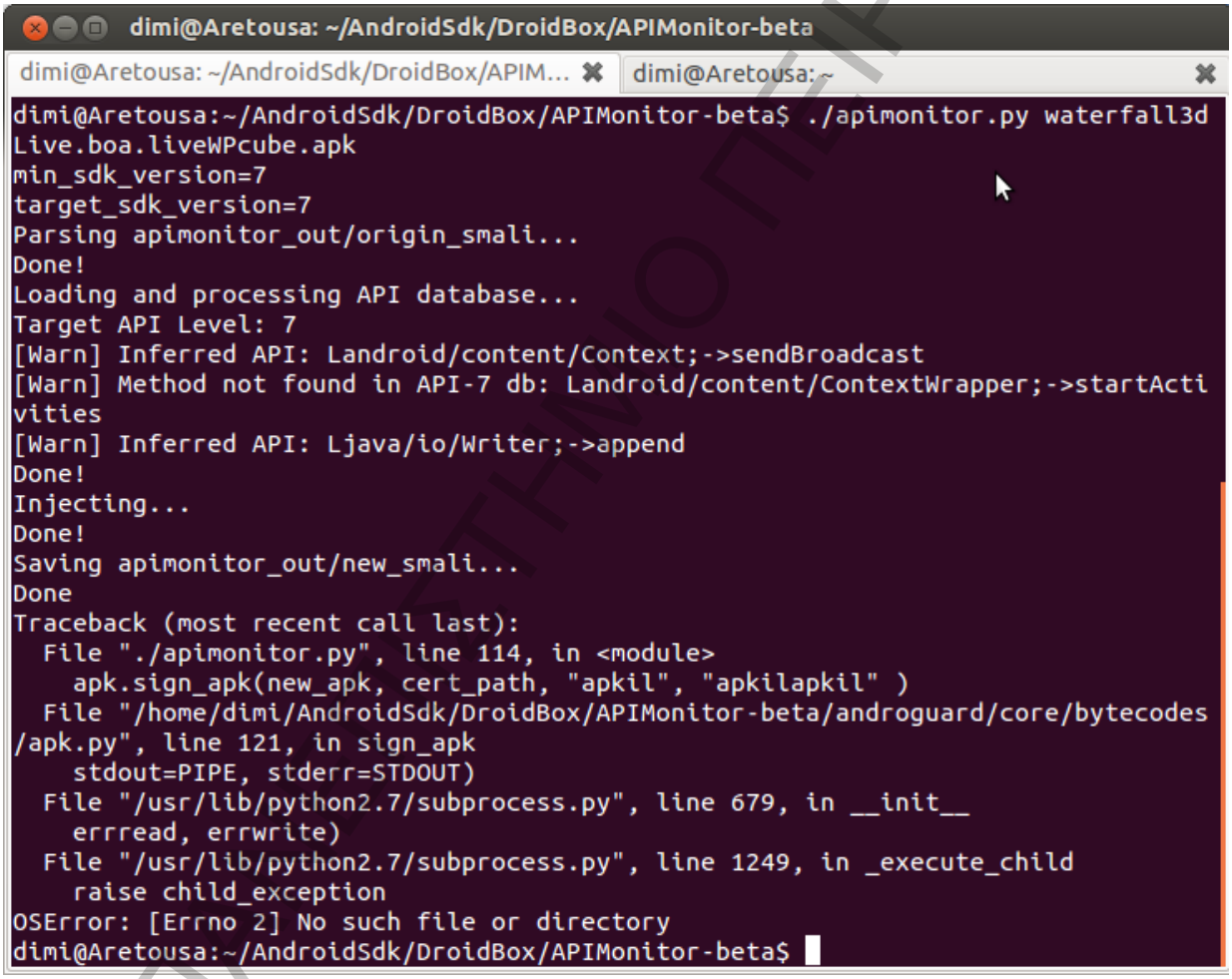
apk ανάλογα με την έκδοση API, διαδικασία αρκετά χρονοβόρα. Για αυτό τον λόγο προτάθηκε και αναπτύχθηκε η λύση του API MONITOR, που ανακατασκευάζει το apk έτσι ώστε να εκτελείται και να αναλύεται απ' ευθείας από το DroidBox.

Χρήση:

$ cd APIMonitor-beta

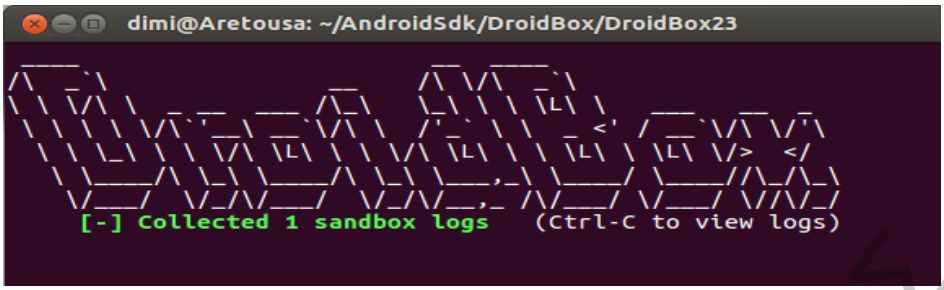
$./apimonitor.py example/APKILTests.apk

και παράγει ως αποτέλεσμα ένα αρχείο με όνομα <δοθέν apk>_new.apk



Εικόνα 77, Η ανασκευή της εφαρμογής waterfall3d από το API MONITOR

6.5.2.2 DroidBox



Εικόνα 78. Η αρχική οθόνη του DroidBox

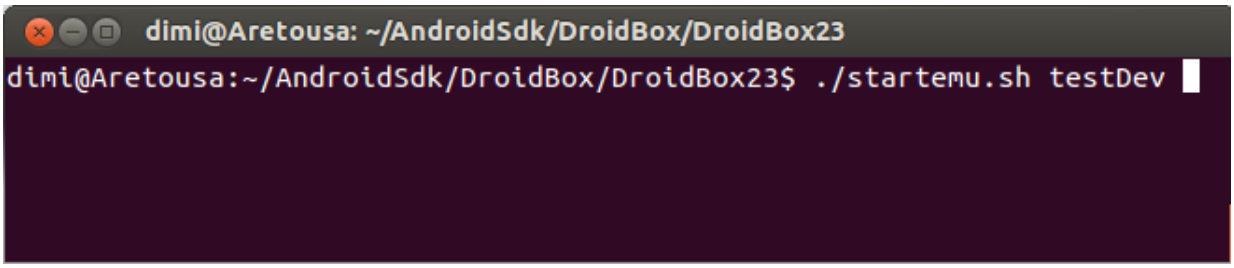
Το DroidBox [58] είναι ένα Sandbox περιβάλλον που αναπτύχθηκε για δυναμική ανάλυση εφαρμογών Android. Οι ακόλουθες πληροφορίες παράγονται όταν ολοκληρωθεί η ανάλυση:

- Τα hashes του πακέτου που αναλύθηκε
- Εισερχόμενη / εξερχόμενη κίνηση δικτύου
- Ενέργειες ανάγνωσης και εγγραφής σε αρχείο
- Τα services και οι κλάσεις που φορτώνονται από τον DexClassLoader
- Διαρροές πληροφορίας από το δίκτυο (αρχεία και SMS)
- Παράκαμψη δικαιωμάτων
- Κρυπτογραφικές διεργασίες που εκτελούνται με τη χρήση του Android API
- Εκχώρηση broadcast receivers
- Αποστολή SMS και κλήσεις σε τηλεφωνικούς αριθμούς.

Επιπλέον δημιουργούνται δύο εικόνες για εικονική παρουσίαση της συμπεριφοράς της εφαρμογής που εκτελέστηκε. Το γράφημα συμπεριφοράς της εφαρμογής που δείχνει τη χρονική σειρά των εργασιών αλλά και ένα δενδροδιάγραμμα (treemap) που μπορεί να χρησιμοποιηθεί για να ελεγχθεί η ομοιότητα μεταξύ πακέτων που αναλύθηκαν. Το treemap συνήθως χρησιμοποιείται για περιπτώσεις σύγκρισης αναδημοσιευμένων εφαρμογών σε εναλλακτικά markets.

Χρήση:

Αφού εγκατασταθεί το DroidBox και ρυθμιστούν οι παράμετροι του Adndoid SDK στον home φάκελο του DroidBox εκκινείται η εικονική συσκευή που θα λειτουργεί ως sandbox περιβάλλον για την εκτέλεση των malware:



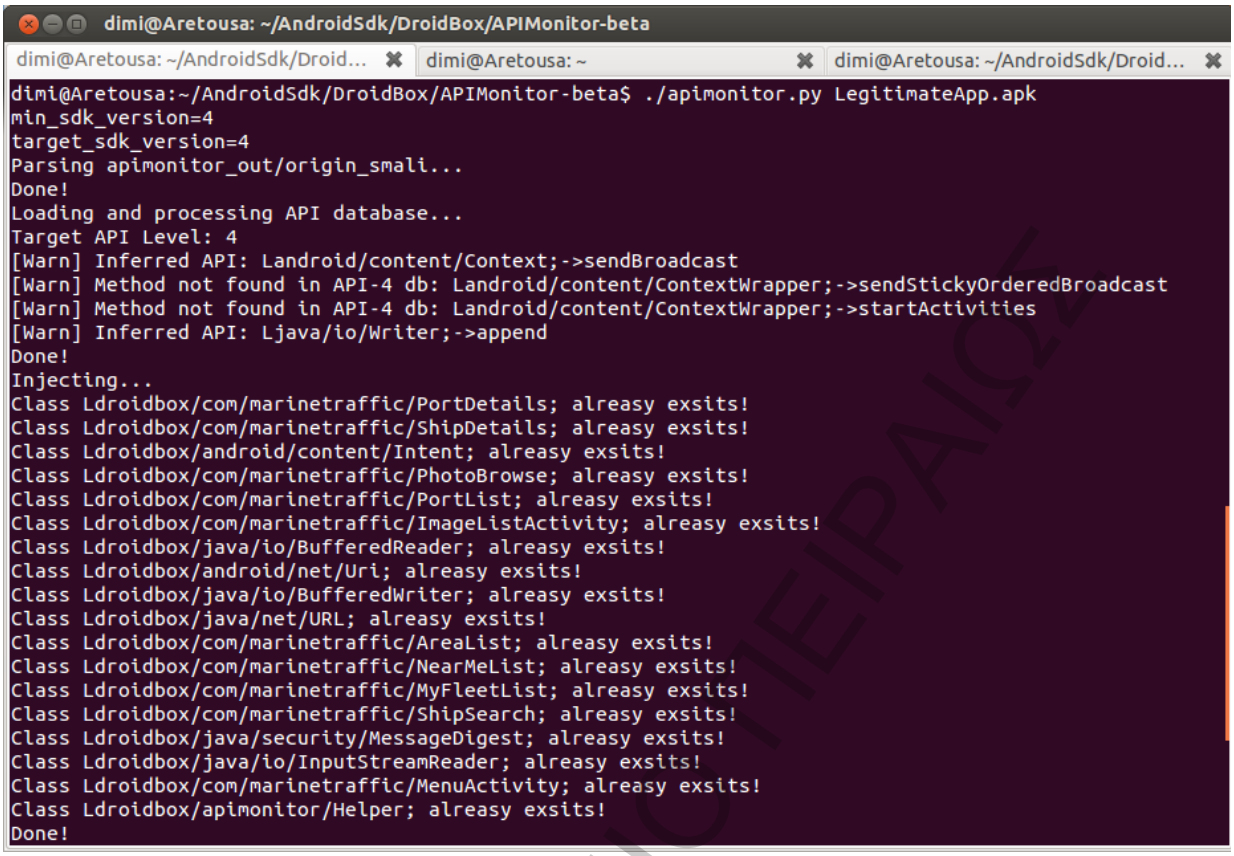
Εικόνα 79, Εκκίνηση της εικονικής συσκευής testDev μέσω του DroidBox

Εικόνα 80, Η εικονική συσκευή ανοίγει κανονικά

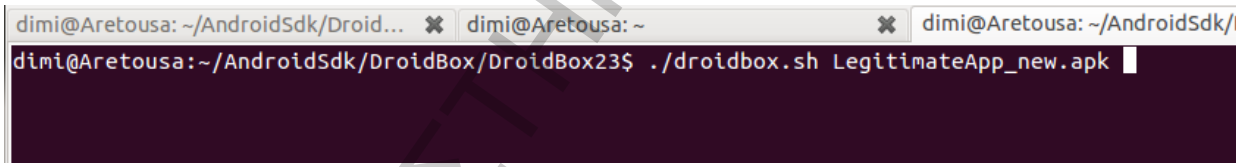
Όταν ολοκληρωθεί η εκκίνηση του Android στην εικονική συσκευή εκτελείται :

./droidbox.sh <ονομα.apk>

Δείγμα 1ο LegitimateApp: Μία κανονική εφαρμογή

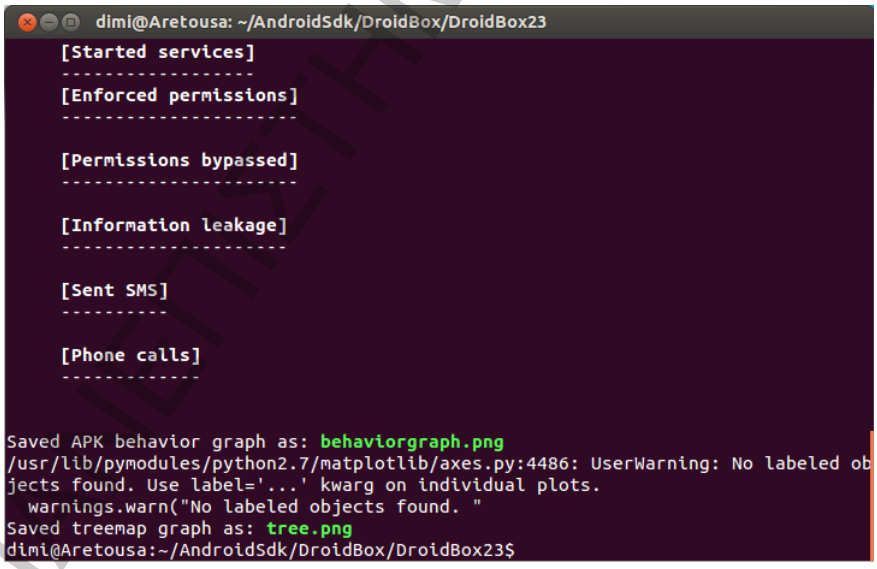


Εικόνα 81, Το APIMONITOR παράγει το LegitimateApp_new.apk

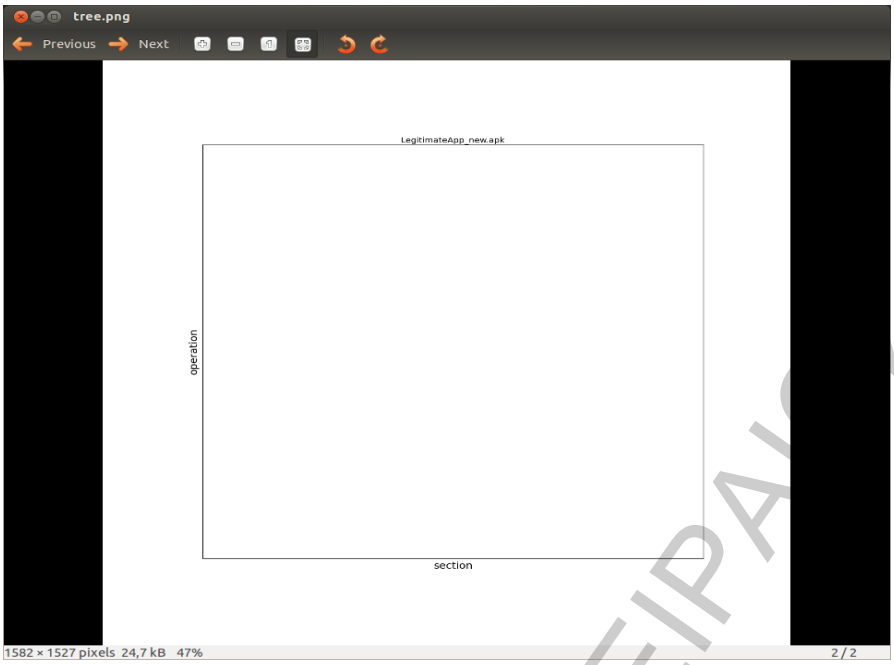


Εικόνα 82, Εκτέλεση στο sandbox

Εικόνα 83, Η αναφορά της εκτέλεσης του LegitimateApp 1/2



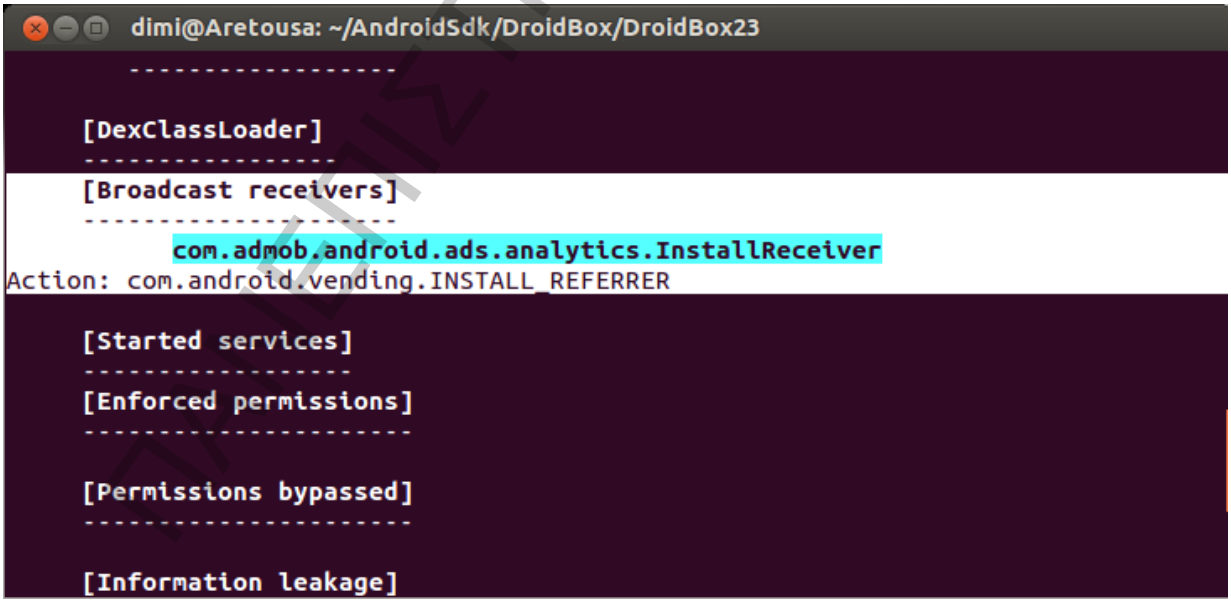
Εικόνα 84, Η αναφορά της εκτέλεσης του LegitimateApp 2/2



Εικόνα 85, Το treemap της εφαρμογής είναι κενό

Συμπεράσματα ανάλυσης δείγματος 1:

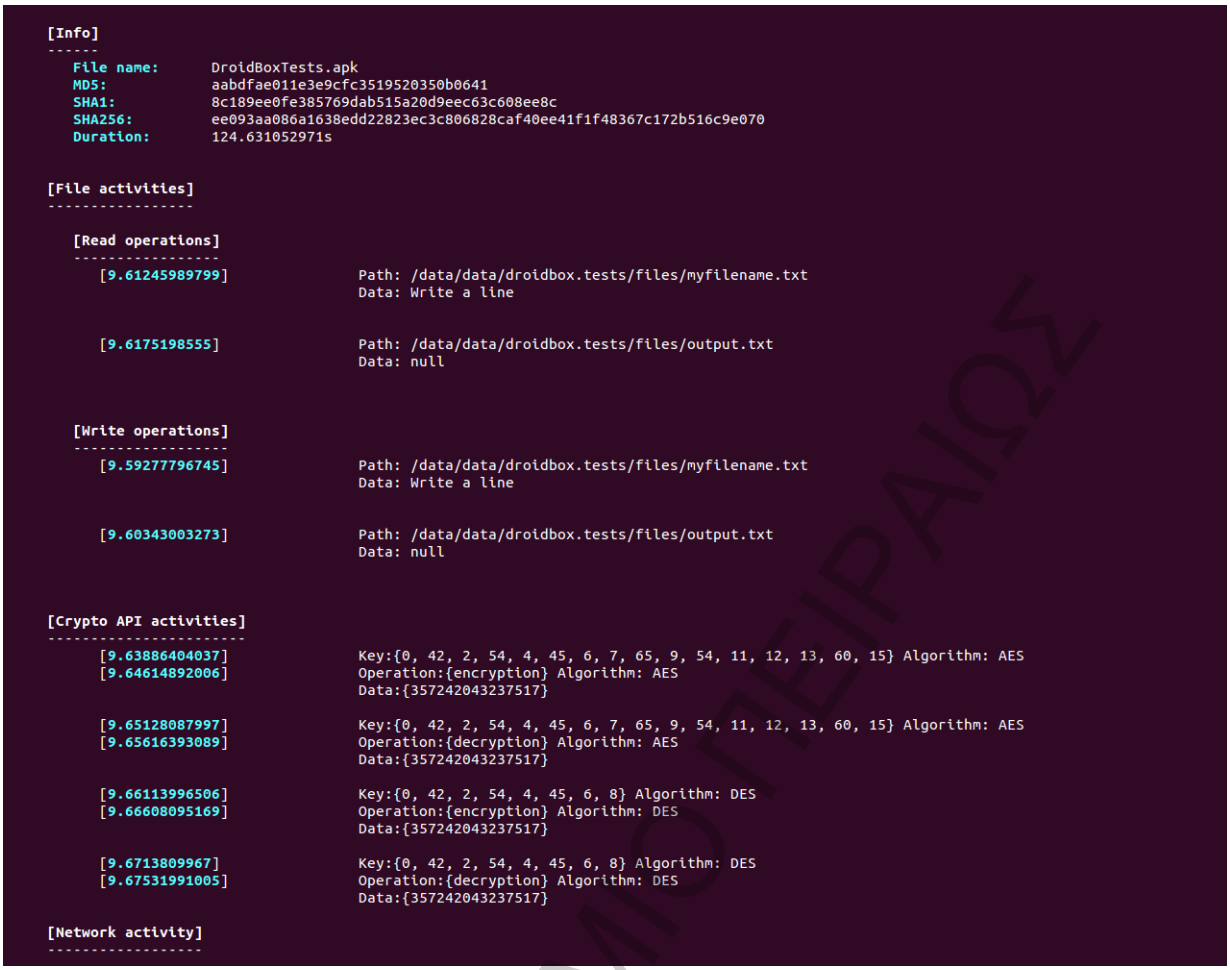
Το γεγονός ότι δεν υπήρχαν ευρήματα δεν είναι αρκετό από μόνο του για να επιβεβαιώσει στον ερευνητή την αθωότητα μίας εφαρμογής. Όπως επίσης μπορεί κάποια ευρήματα να αφορούν την κανονική λειτουργία της εφαρμογής και όχι κάποια κακόβουλη δραστηριότητα. Στο συγκεκριμένο δείγμα εντοπίζεται μία εκχώρηση ενός Broadcast receiver που αφορά τις διαφημίσεις που εμφανίζονται στην εφαρμογή.



Εικόνα 86, Εντοπίστηκε ένας ύποπτος Broadcast receiver

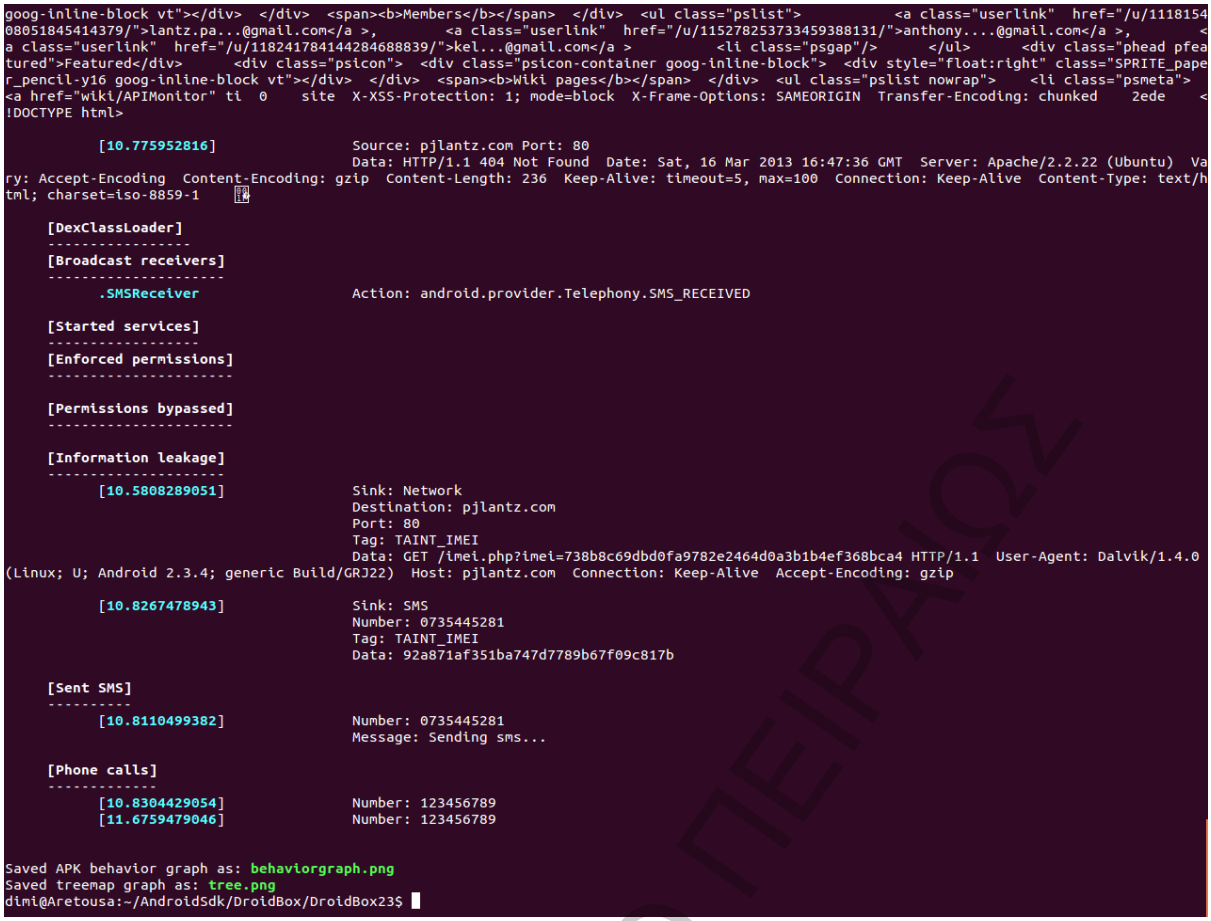
Δείγμα 2 DroidBoxTests: Η demo εφαρμογή του DroidBox

Εικόνα 87, Το DroidBox εντόπισε 22 ευρήματα

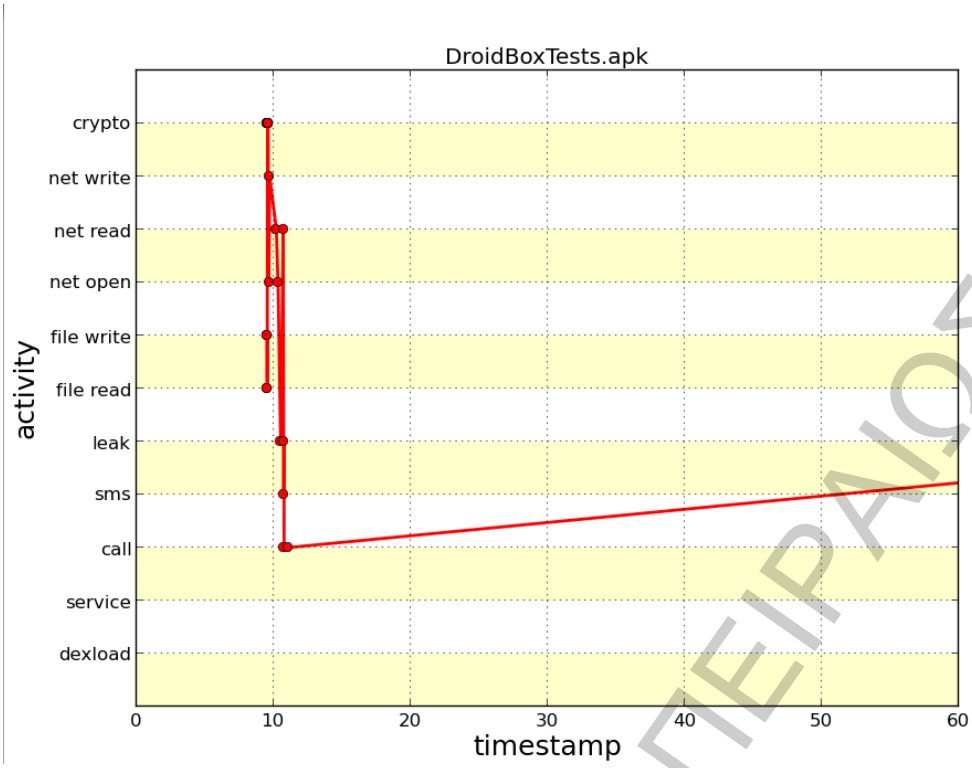


Εικόνα 88. Η αναφορά του DroidBoxTests 1/3

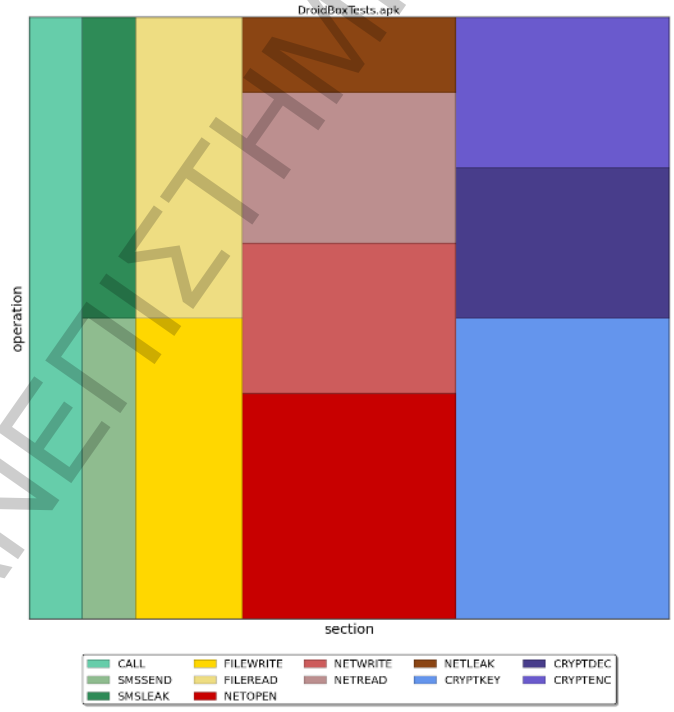
Εικόνα 89, Η αναφορά του DroidBoxTests 2/3



Εικόνα 90, Η αναφορά του DroidBoxTests 3/3



Εικόνα 91, Το γράφημα συμπεριφοράς της demo εφαρμογής



Εικόνα 92, Το treemap της demo εφαρμογής

Συμπεράσματα ανάλυσης δείγματος 2:

Το συγκεκριμένο δείγμα έχει ως σκοπό την επίδειξη της συσκευής. Καταγράφονται δραστηριότητες όπως:

Αρχεία:

Ανάγνωση και εγγραφή του αρχείου: myfilename.txt

Κρυπτογράφηση

με AES και κλειδί Key:{0, 42, 2, 54, 4, 45, 6, 7, 65, 9, 54, 11, 12, 13, 60, 15}

με DES και κλειδί Key:{0, 42, 2, 54, 4, 45, 6, 8}

Δίκτυο:

Σύνδεση στα code.google.com Port: 80 και pjlantz.com Port: 80

Και λήψη των δεδομένων αυτών των ιστοτόπων.

Broadcast receivers:

Εκχώρηση του receiver.SMSReceiver (που λαμβάνει τα sms που έρχονται στην συσκευή)

Διαρροή πληροφοριών:

Αποστέλλει στο pjlantz.com (μη υπαρκτός ιστότοπος) το IMEI της συσκευής και κάποια άλλα στοιχεία του χρήστη.

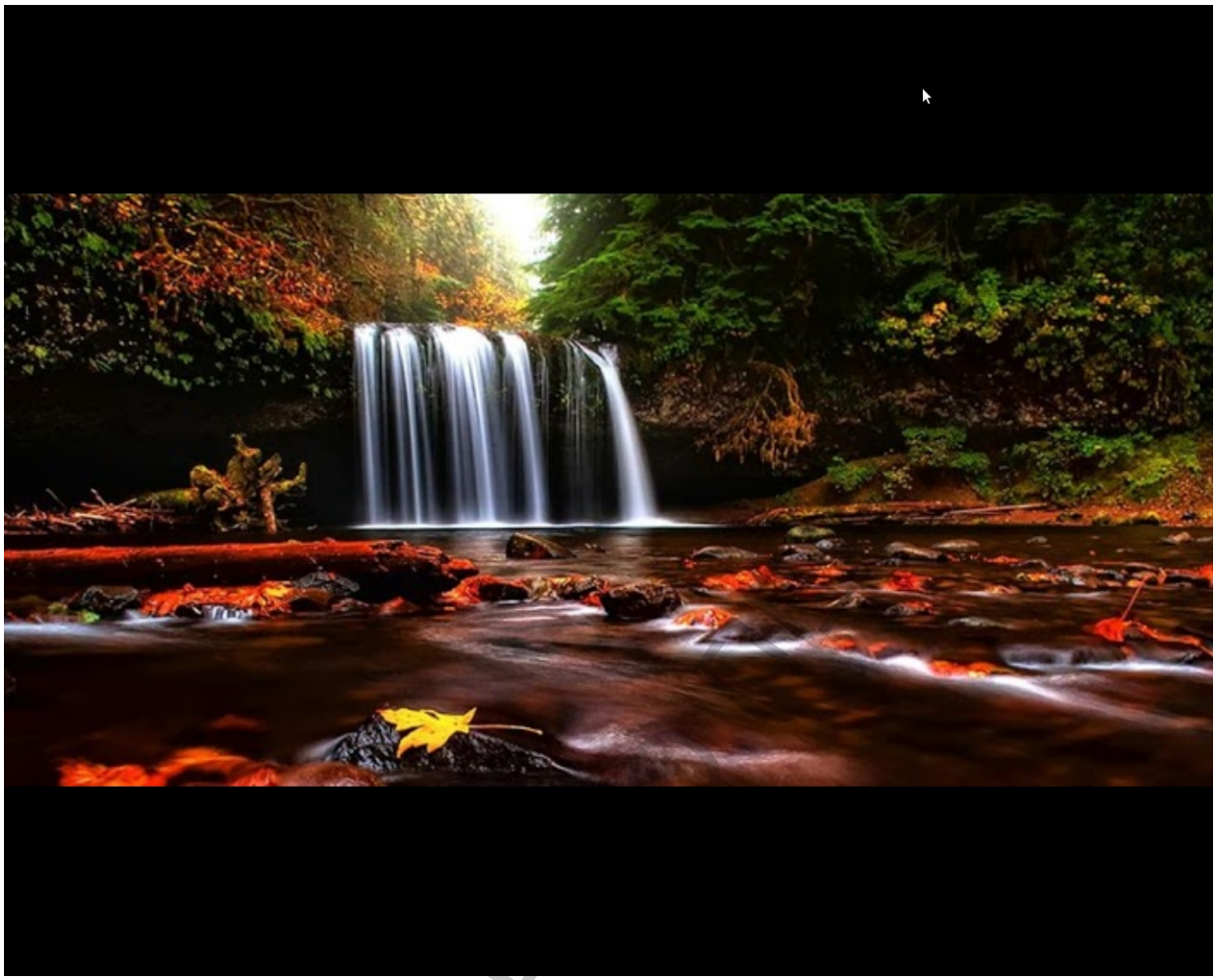
Διαρροή πληροφοριών + Αποστολή sms:

Επίσης αποστέλλει με sms το IMEI στον αριθμό 0735445281

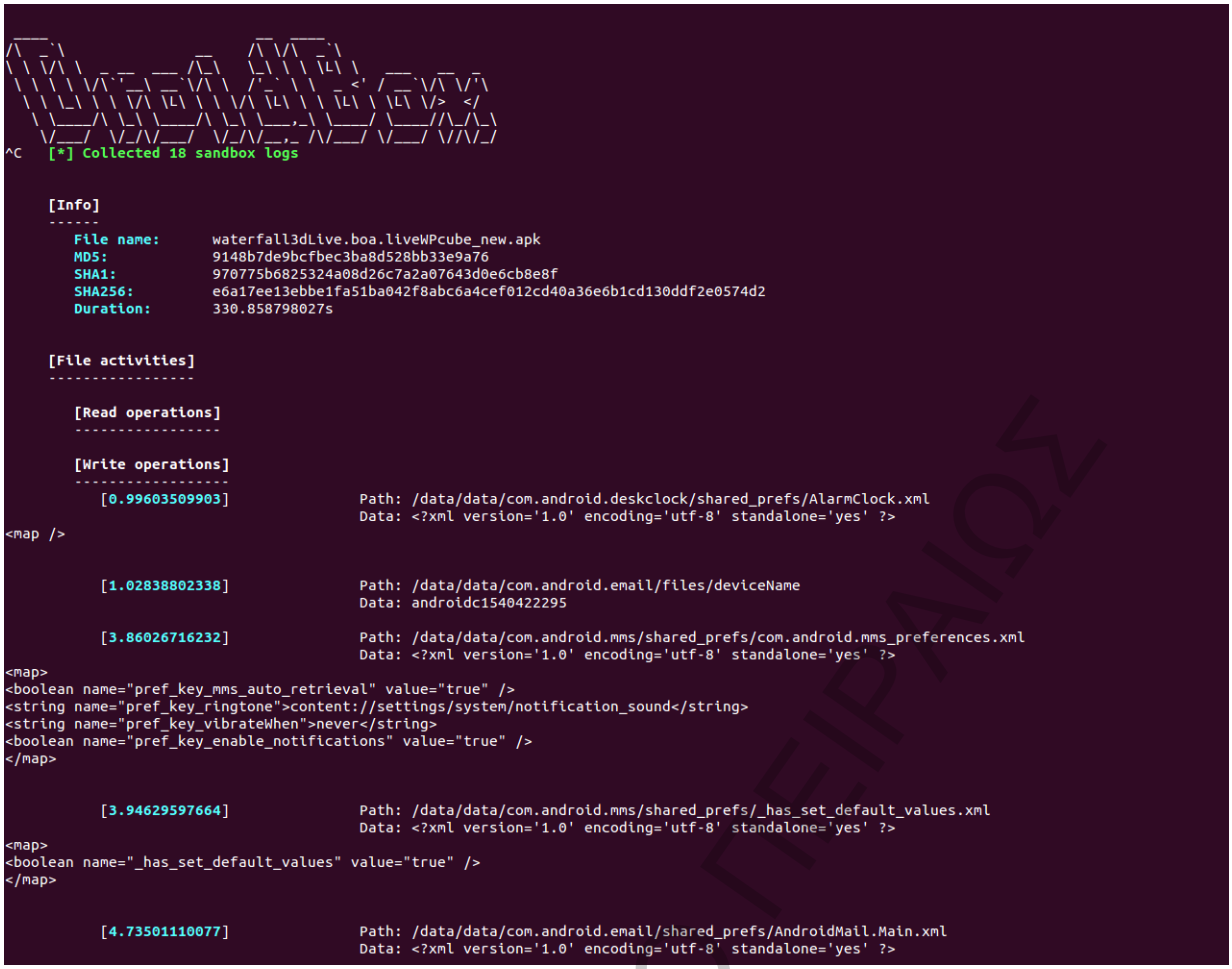
Κλήσεις:

Καλεί τον αριθμό 123456789. Αν ο χρήστης τερματίσει την κλήση, η εφαρμογή επαναλαμβάνει την κλήση.

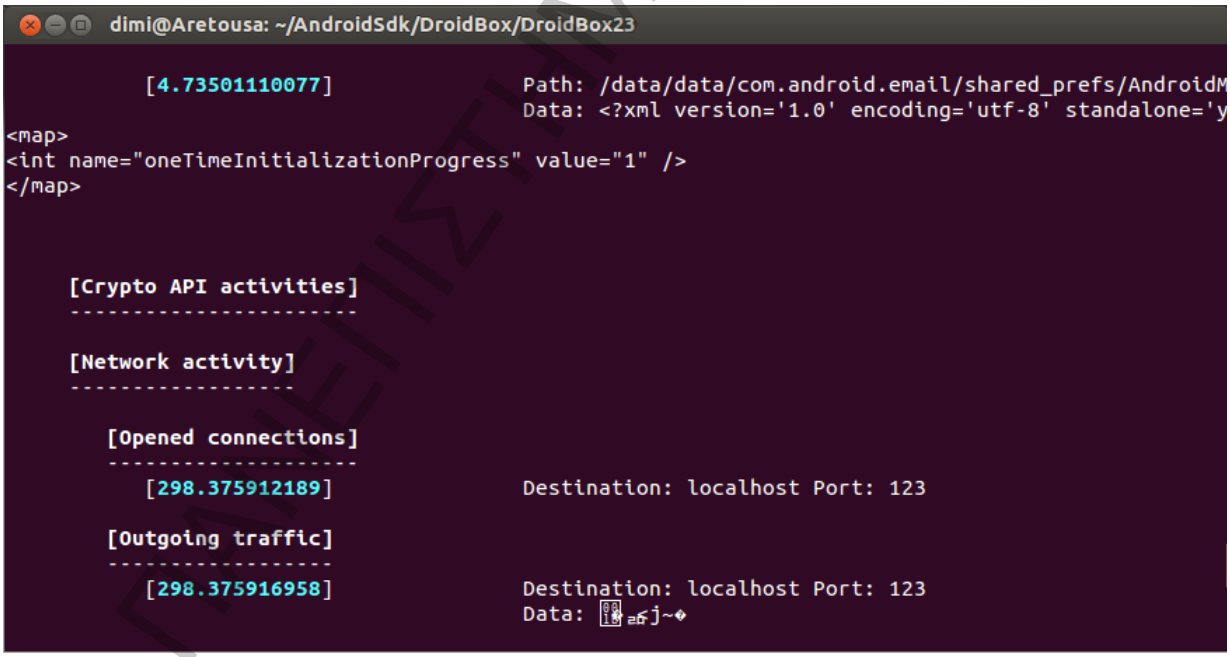
Δείγμα 3 waterfall3dLive.boa.liveWPcube : malware



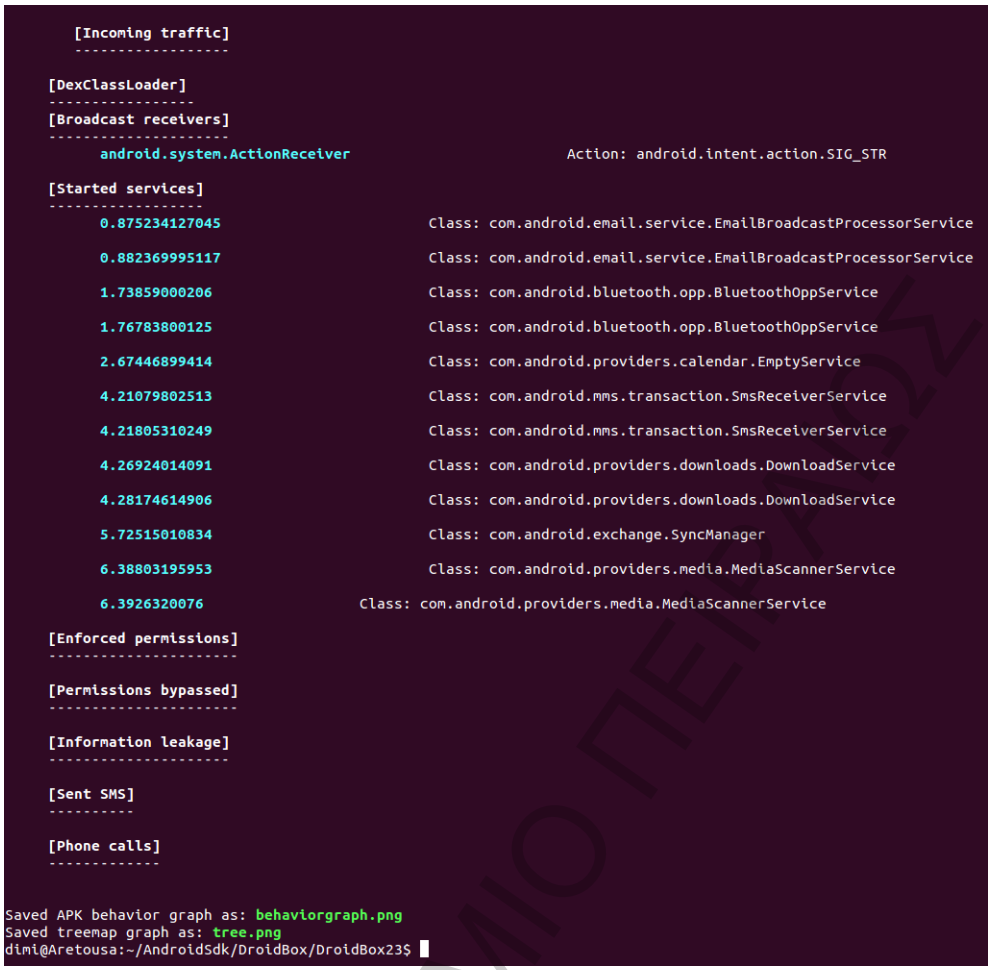
Εικόνα 93. Η εφαρμογή waterfall3d



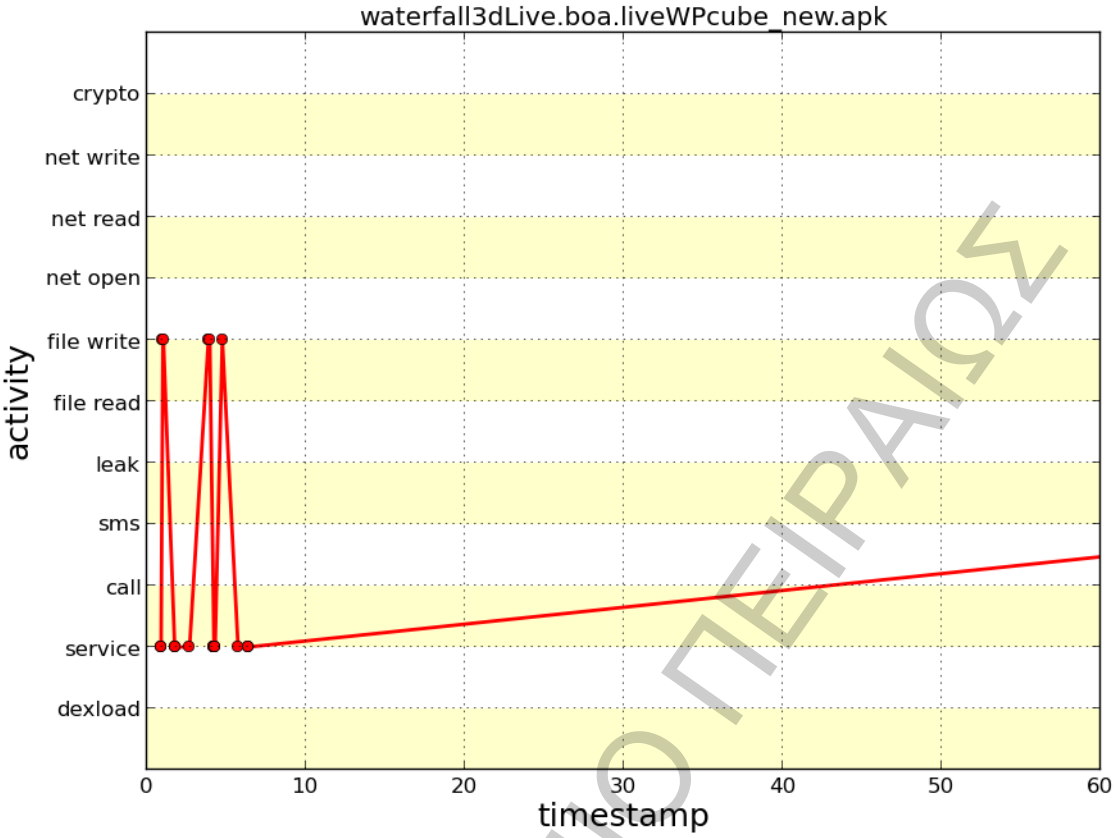
Εικόνα 94, Το DroidBox εντόπισε 18 ευρήματα, αναφορά 1/3



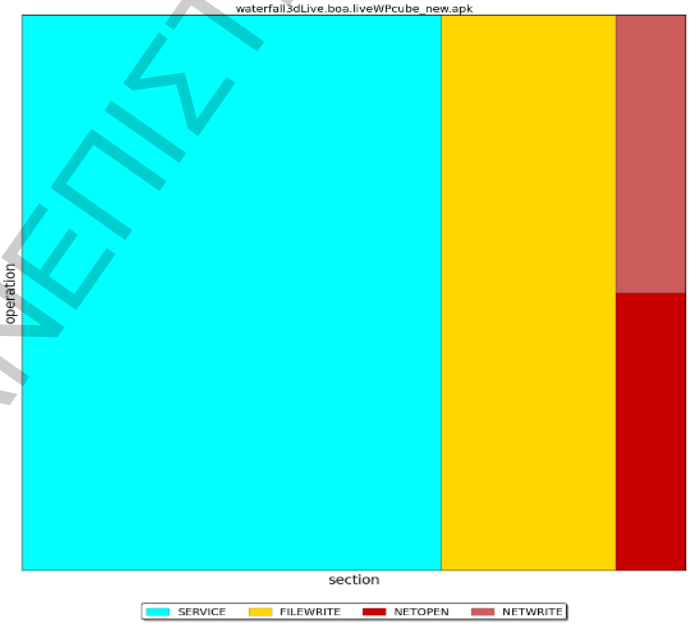
Εικόνα 95, Αναφορά 2/3



Εικόνα 96, αναφορά 3/3



Εικόνα 97, Το γράφημα συμπεριφοράς του waterfal3d



Εικόνα 98, Το treemap του waterfall3d

Συμπεράσματα ανάλυσης δείγματος 3:

Στο συγκεκριμένο δείγμα εντοπίστηκαν 18 sandbox logs πιο συγκεκριμένα:

Αρχεία:

Τροποποιεί τα αρχεία:

/data/data/com.android.deskclock/shared_prefs/AlarmClock.xml

/data/data/com.android.email/files/deviceName

/data/data/com.android.mms/shared_prefs/com.android.mms_preferences.xml

/data/data/com.android.mms/shared_prefs/_has_set_default_values.xml

/data/data/com.android.email/shared_prefs/AndroidMail.Main.xml

Δίκτυο:

Συνδέεται σε κάποιον localhost στην θύρα 123. (πιθανότατα σε κάποιο από τα αρχεία που τροποποιήθηκαν άλλαξε και η τιμή του localhost, από 127.0.0.1 σε μία static ip που ανήκει σε κάποιο απομακρυσμένο server που ελέγχει ο συγγραφέας του malware).

Broadcast receivers:

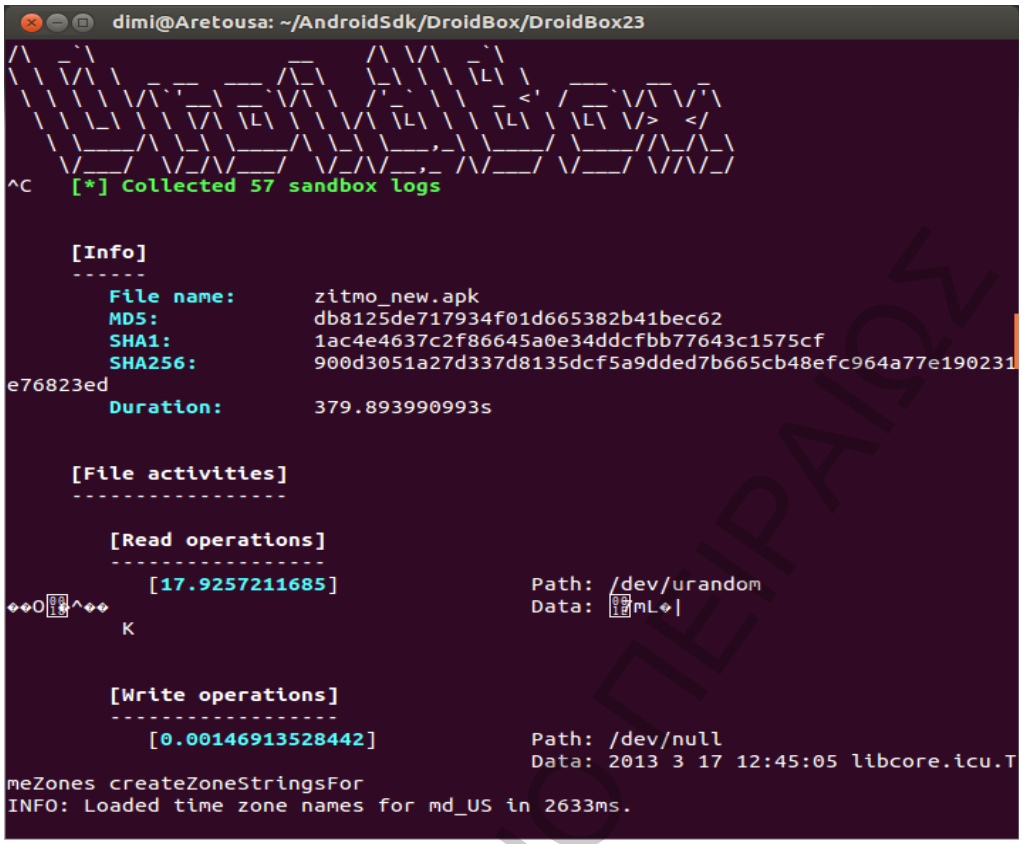
Εκχώρηση του receiver Action: android.intent.action.SIG_STR (πρόκειται για προσαρμοσμένο receiver, τον έχει κατασκευάσει ο συγγραφέας του malware)

Services:

Με την εκκίνηση της εφαρμογής ξεκινάνε 12 Services εκ των οποίων σχεδόν όλα υπό συνθήκες μπορούν να είναι κακόβουλα. Όταν η αυθεντική εφαρμογή [78] αιτείται 2 τα οποία δεν ξεκινάνε κατά την εκκίνηση.

Σημειωση: Ο αναλυτής πριν εκτελέσει το malware στο sandbox οφείλει να έχει ολοκληρώσει την στατιστική ανάλυση ώστε να γνωρίζει περισσότερες λεπτομέρειες για το τι κάνει η εφαρμογή.

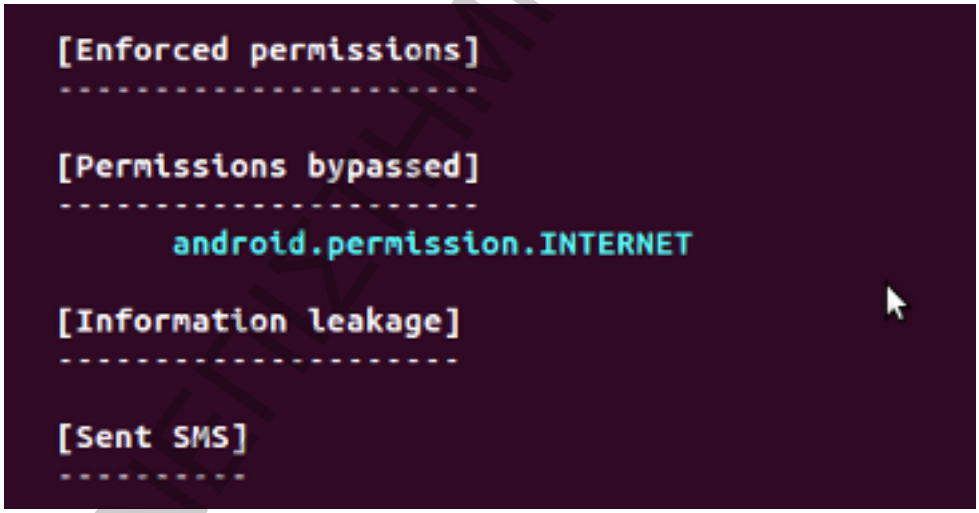
Δείγμα 4 Zitmo : malware



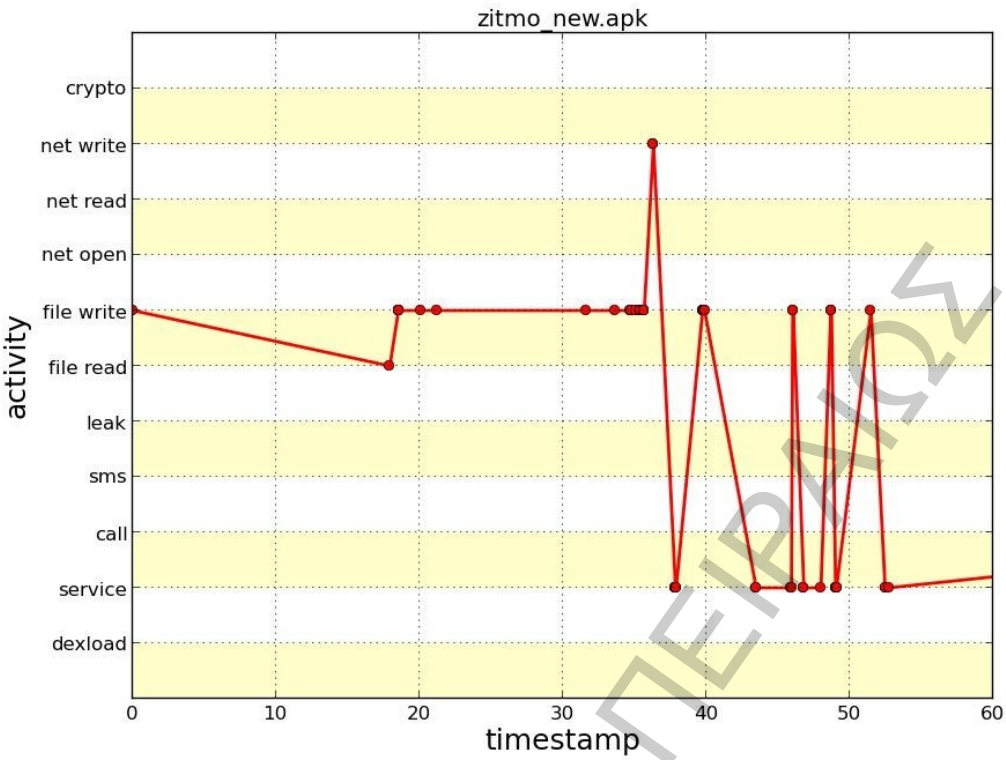
Εικόνα 99, Το DroidBox εντόπισε 57 ύποπτα ευρήματα , αναφορά 1/4

Εικόνα 100, Αναφορά της εκτέλεσης του Zitmo 2/4

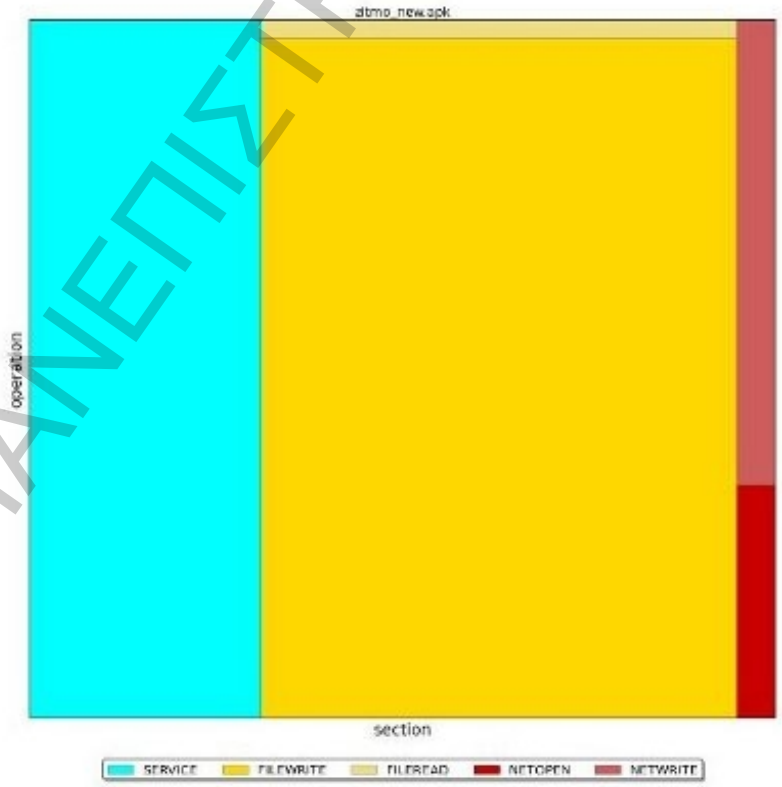
Εικόνα 101, Αναφορά της εκτέλεσης του Zitmo 3/4



Εικόνα 102, Αναφορά της εκτέλεσης του Zitmo 4/4



Εικόνα 103,Το γράφημα συμπεριφοράς του zitmo



Εικόνα 104, Το treemap του zitmo

Συμπεράσματα ανάλυσης δείγματος 4:

Στο συγκεκριμένο δείγμα εντοπίστηκαν 18 sandbox logs πιο συγκεκριμένα:

Αρχεία:

Διαβάζει από το /dev/urandom

Data: #7mL  |

 O#^

Τροποποιεί τα αρχεία:

/data/backup/pending/journal-1057936222.tmp

/data/data/com.android.providers.calendar/shared_prefs/CalendarUpgradeReceiver.xml

/data/data/com.android.providers.contacts/shared_prefs/ContactsUpgradeReceiver.xml

/data/data/com.android.phone/shared_prefs/_has_set_default_values.xml

/data/com.android.inputmethod.latin/shared_prefs/com.android.inputmethod.latin_preferences.xml

/data/data/com.android.providers.telephony/shared_prefs/preferred-apn.xml

/data/data/com.android.providers.telephony/shared_prefs/preferred-apn.xml

/data/data/com.android.providers.contacts/shared_prefs/com.android.providers.contacts_preferences.xml

/data/data/com.android.mms/shared_prefs/com.android.mms_preferences.xml

Δίκτυο:

Συνδέεται σε κάποιον localhost στην θύρα 123.

Destination: localhost Port: 123

και αποστέλλει κάποια δεδομένα: #  9  ; 

Broadcast receivers:

Εκχωρούνται οι receivers:

com.security.service.receiver.SmsReceiver

Action: android.provider.Telephony.SMS_RECEIVED

com.security.service.receiver.ActionReceiver

Action: android.intent.action.USER_PRESENT

com.security.service.receiver.RebootReceiver

Action: android.intent.action.ACTION_SHUTDOWN

Services:

com.android.providers.downloads.DownloadService

com.android.providers.media.MediaScannerService

com.android.email.service.EmailBroadcastProcessorService

com.android.bluetooth.opp.BluetoothOppService

com.android.providers.calendar.EmptyService

com.android.mms.transaction.SmsReceiverService

com.android.providers.downloads.DownloadService

com.android.providers.media.MediaScannerService

com.android.exchange.SyncManage**r**

Δικαιώματα που παρακάμφθηκαν:

android.permission.INTERNET

Το δείγμα 4 έχει αρκετά στοιχεία που προδίδουν την ενοχή του ως malware.

Τα services και ο συνδυασμός actions που ξεκινάει η εφαρμογή υποδεικνύουν ότι σκοπό έχει την υποκλοπή sms. Όπως επίσης παρακάμπτεται ο έλεγχος του δικαιώματος χρήσης Internet.

Σημείωση:το συγκεκριμένο malware αποτελεί υλοποίηση του γνωστού malware Zeus για τα Android. [79]

6.7 Δυναμική ανάλυση Βέλτιστη πρακτική

1) Το πρώτο βήμα της δυναμικής ανάλυσης αποτελεί η συγκέντρωση όλων των αποτελεσμάτων της στατικής ανάλυσης. Η οποία αποτελεί το κλειδί της οργάνωσης κατά την έρευνα μίας εφαρμογής. Τα ελάχιστα στοιχεία που οφείλει να έχει ένας αναλυτής μετά το πέρας της στατικής ανάλυσης είναι:

1. Λίστα των δικαιωμάτων
2. Τμήματα του ύποπτου κώδικα
3. Σημειώσεις που περιλαμβάνουν κάθε άλλο σχετικό εύρημα.
4. Την υπόθεση της κατάστασης που έγινε στο τέλος της Στατικής ανάλυσης.

Ο αναλυτής, αρχικά οφείλει να μελετήσει τα αποτελέσματα της στατικής ανάλυσης. Τι ύποπτα δικαιώματα εντοπίστηκαν, με τι μεθόδους συνδέονται, τι δεδομένα αποθηκεύονται και που.

2) Με βάσει τα δεδομένα αυτά, ο αναλυτής πρέπει να μελετήσει και να αποφασίσει το πως θα εκτελεστεί η εφαρμογή. Η εξέταση του κώδικα μίας εφαρμογής αποτελεί την καλύτερη λύση για να εκτιμηθεί σε ικανοποιητικό βαθμό ο τρόπος με τον οποίο θα συμπεριφερθεί η εφαρμογή.

3) Μόλις βγει το πόρισμα για το πως περίπου θα συμπεριφερθεί η εφαρμογή, το επόμενο βήμα είναι να σχεδιαστεί το σενάριο ελέγχου (test scenario). Ώστε να επιβεβαιωθούν τα ευρήματα της στατικής ανάλυσης.

Ο σχεδιασμός του test scenario αποτελείται από τον καθορισμό

- Της πλατφόρμας ανάλυσης
- Των συμμετεχόντων συσκευών
- Την ή τις έκδοση/εις του λειτουργικού συστήματος.
- Των εργαλείων που θα χρησιμοποιούν

4) Μετά την ολοκλήρωση της εκτίμησης της συμπεριφοράς και τον σχεδιασμό του test scenario, το επόμενο βήμα είναι η σύσταση του test scenario για την επιβεβαίωση της υπόθεσης. Για παράδειγμα, στην περίπτωση που μία εφαρμογή είναι ύποπτη για υποκλοπή sms

σε μία συσκευή, στο test scenario οφείλουν να είναι δύο εικονικές συσκευές και να μελετάται η αλληλεπίδραση τους [80].

Όπως για παράδειγμα, η Trusteer Rapport case[38] όπου εξετάστηκε η υπόθεση στέλνοντας sms από μία καθαρή συσκευή σε μία μολυσμένη συσκευή. Τα μηνύματα αυτά ποτέ δεν παρελήφθησαν από την μολυσμένη συσκευή. Γεγονός που επιβεβαίωσε την υπόθεση αναγνωρίζοντας πλέον το malware ως Severe κίνδυνο.

Το test scenario πρέπει να εκτελείται με ιδιαίτερη προσοχή στα αποτελέσματα τα οποία αναμένονται από τον σχεδιασμό του ίδιου του test scenario. Αφού συσταθεί και μελετηθεί το test scenario, είναι η ώρα για την εκτέλεση της εφαρμογής σε ένα sandboxed περιβάλλον ώστε να επιβεβαιωθεί η υπόθεση. Είναι πολύ σημαντικό να καταγράφεται το καθετί που συμβαίνει κατά την διάρκεια του test. Πολλές φορές αρκετά ενδιαφέροντα ευρήματα για την υπόθεση, βρίσκονται σε άσχετα μέρη. Όπως για παράδειγμα σε μία εφαρμογή παρόλο που δεν αναμένεται δικτυακή κίνηση από την εφαρμογή, είναι υποχρεωτικό να εκτελείται το Wireshark καθ' όλη την διάρκεια του test. Και φυσικά οφείλει ο αναλυτής να καταγράφει (με κείμενο ή εικόνα ή Wireshark κτλ...) όλα τα ευρήματα είτε φαινομενικά σημαντικά είτε ασήμαντα.

Το test θεωρείται ότι έφτασε στο τέλος του, όταν ο αναλυτής έχει συγκεντρώσει όσο το δυνατόν περισσότερα στοιχεία ώστε να καταρρίπτουν ή να επιβεβαιώνουν την υπόθεση. Τέλος συντάσσεται η αναφορά. Όπου σημάνει και το τέλος της έρευνας.

38 http://osaf-community.org/images/threatindeximages/tr/trreport.pdf

Συμπεράσματα 6ου κεφαλαίου

Οι συγγραφείς κακόβουλου λογισμικού γίνονται όλο και πιο έμπειροι και αρχίζουν να ενσωματώνουν ρουτίνες που ελέγχουν εάν τα malware τους εκτελούνται μέσα σε ένα εικονικό περιβάλλον. Αν αυτό επιβεβαιωθεί το κακόβουλο λογισμικό μπορεί να αλλάξει τον τρόπο λειτουργίας του ή ακόμα και να απενεργοποιηθεί. Γιαυτό τον λόγο πρέπει να λαμβάνονται μέτρα ώστε το malware να νομίζει ότι εκτελείται σε πραγματικό περιβάλλον και όχι σε εικονικό. Παράλληλα ο αναλυτής οφείλει να διαφυλάσσει τον εαυτό του από τυχών επιθέσεις που μπορεί να ξεκινήσει ένα malware κατά την διάρκεια της ανάλυσης του. Η ανάλυση πρέπει να σχεδιάζεται καλά και να εκτελείται με βάση τις βέλτιστες πρακτικές στατικής[39] και δυναμικής[40]. Τα δείγματα malware που εξετάστηκαν αποδεικνύουν ότι η δομή τους έχει ευάλωτα σημεία που προδίδουν την κακόβουλη διάθεση τους (παράδειγμα το waterfall.. όπου ο αναλυτής αντιλαμβάνεται ότι μία εφαρμογή wallpaper δεν χρειάζεται να έχει αυτά τα δικαιώματα) Το android sdk έδειξε πολύ σημαντικά ευρήματα (πχ εικόνα 75). Όπως επίσης το droidbox αποδείχθηκε ιδιαίτερα ικανό sandbox αποκαλύπτοντας την κακόβουλη δραστηριότητά σε κάθε δείγμα.

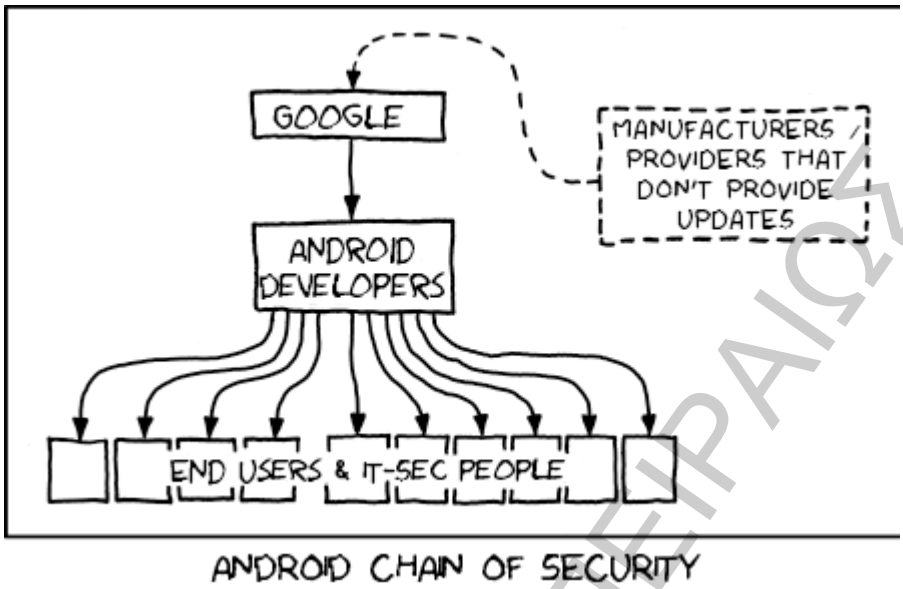
(Πηγές ανάλυσης: [89], [90], [91], [92], [93], [94], [95], [96], [97], [98], [99])

39 Κεφάλαιο 6.4

40 Κεφάλαιο 6.7

Κεφάλαιο 7ο Ανάλυση απειλών στα Android

Don't assume the sky is falling.... assume that it already fell



Εικόνα 105, xckcd για την ασφάλεια του Android

Για να προσδιοριστεί το επίπεδο ασφάλειας στο Android πρέπει να οριστεί το πεδίο εφαρμογής της - security scope [83]. Γιαυτό το λόγο γίνεται και η μοντελοποίηση των απειλών που με βάση το πως ορίζεται η ασφαλεία, που εφαρμόζεται, ποιες είναι οι παράμετροι που την επηρεάζουν, ποια είναι τα όρια του λειτουργικού, πότε ένα σύστημα Android θεωρείται ασφαλές και κατά πόσο μπορεί να θεωρείται ασφαλές.

7.1 Οι Top 10 κίνδυνοι στις κινητές συσκευές

Οι σύγχρονες κινητές συσκευές διαθέτουν τη λειτουργική ικανότητα ενός desktop ή ενός laptop με ένα λειτουργικό σύστημα γενικού σκοπού. Από αυτή την άποψη πολλές από τις απειλές είναι παρόμοιες με εκείνες των παραδοσιακών spyware, Trojan και των μη ασφαλώς σχεδιασμένων εφαρμογών. Ωστόσο, οι κινητές συσκευές δεν είναι μόνο μικροί υπολογιστές. Οι κινητές συσκευές έχουν σχεδιαστεί με βάση την λειτουργικότητα της προσωπικής επικοινωνίας γεγονός που διαφοροποιεί τις απειλές των κινητών εφαρμογών από τις παραδοσιακές απειλές των υπολογιστών. Απάντηση σε αυτό το θέμα δίνουν ο OWASP και η Veracode με την μοντελοποίηση των απειλών υπό την μορφή της λίστας Top 10. [84]

1) Το Top 10 Mobile Application Risks είναι ένα έργο του OWASP με σκοπό να ενημερώσει και να κατευθύνει τους προγραμματιστές και τους υπεύθυνους ασφαλείας για την συμπεριφορά των ιών σε κινητές συσκευές, που εκτίθενται οι χρήστες. Η συμπεριφορά αυτή μπορεί να σχεδιαστεί είτε κακόβουλα ή ακούσια.

2) Το Top Mobile App 10 μπορεί να χρησιμοποιηθεί για τον καθορισμό της κάλυψης μιας λύσης ασφαλείας η οποία μπορεί να προστατεύσει από αυτές τις απειλές. Μια λύση ασφαλείας μπορεί να δηλώσει αυτή την κάλυψη του κινητού του με βάση το Top Mobile App 10 έτσι ώστε οι πελάτες μπορούν να αντιληφθούν τι απειλές μετριάζονται σε αυτή τη λύση.

Οι Top Mobile app λύσεις ασφάλειας μπορούν να χρησιμοποιηθούν κατά την ανάπτυξη μιας εφαρμογής, ως μέρος της ένα market, ως διαδικασία έγκρισης, ως δοκιμή αποδοχής μίας εφαρμογής ή ως λογισμικό ασφαλείας που εκτελείται σε μια φορητή συσκευή [84].

7.2 Το Mobile App Top 10 της VERACODE

Σύμφωνα με την VERACODE υπάρχουν 2 κύριες κατηγορίες κινδύνου στις κινητές συσκευές.

Α. Κακόβουλη λειτουργικότητα

Η κατηγορία της κακόβουλης λειτουργίας αποτελείται από μία λίστα ανεπιθύμητων και επικίνδυνων συμπεριφορών που τοποθετούνται κακόβουλα σε μία εφαρμογή με στόχο να παρασυρθεί ο χρήστης και να την εγκαταστήσει.[41]

1. Παρακολούθηση της δραστηριότητας και ανάκτηση δεδομένων
2. Η μη εξουσιοδοτημένες κλήσεις, SMS και χρεώσεις
3. Μη εξουσιοδοτημένη σύνδεση δικτύου (Command & Control, bot)
4. Απομίμηση UI
5. Τροποποίηση συστήματος (rootkit, APN, τροποποίηση του proxy)
6. Λογική ή ωρολογιακή βόμβα

Β. Ευπάθειες

Η κατηγορία των ευπειθειών οφείλεται σε λάθη στο σχεδιασμό ή την υλοποίηση που εκθέτουν τα δεδομένα της συσκευής σε κίνδυνο υποκλοπής και ανάκτησης από επιτιθέμενους. Επίσης οι αδυναμίες αυτές μπορεί να εκθέσουν την συσκευή (και τις πιθανές εφαρμογές cloud που χρησιμοποιεί) σε μη εξουσιοδοτημένη πρόσβαση.

7. Διαρροή ευαίσθητων δεδομένων (ακούσια ή παράπλευρα)
8. Επισφαλής αποθήκευση ευαίσθητων δεδομένων
9. Επισφαλής μετάδοση ευαίσθητων δεδομένων
10. Ενσωμάτωση κωδικών / κλειδιών στον κώδικα

7.2.1 Κατηγορία κακόβουλης λειτουργικότητας

1. Παρακολούθηση δραστηριότητας και ανάκτηση δεδομένων

Η παρακολούθηση της δραστηριότητας και η ανάκτηση δεδομένων είναι η βασική λειτουργικότητα για κάθε spyware. Τα δεδομένα μπορούν να υποκλαπούν σε πραγματικό χρόνο από την στιγμή που δημιουργούνται στην συσκευή. Παραδείγματα αποτελούν: Η αποστολή κάθε εξερχόμενου email σε επιπλέον ένα κρυφό τρίτο παραλήπτη, Η δυνατότητα σε έναν εισβολέα να ακούσει τις τηλεφωνικές κλήσεις ή να διατηρεί ανοιχτό το μικρόφωνο καταγραφής. Αποθηκευμένα δεδομένα, όπως η λίστα επαφών ή αποθηκευμένα email μπορούν επίσης να ανακτηθούν από τον επιτιθέμενο.

Τα ακόλουθα είναι παραδείγματα από δεδομένα μίας συσκευής που μπορεί ένας επιτιθέμενος να παρακολουθήσει ή να υποκλέψει:

41 Κεφάλαιο 3.1.1 Κακόβουλο Λογισμικό

a. Μηνύματα (SMS και e-mail)
b. Ήχο (κλήσεις και ανοιχτό μικρόφωνο καταγραφής)
c. Βίντεο (ακόμα και με πλήρη κίνηση)
d. Τοποθεσία
e. Λίστα επαφών
f. Ιστορικό κλήσεών
g. Ιστορικό περιήγησης
h. Είσοδος (πληκτρολόγιο, φωνή, κτλ..)
i. Αρχεία δεδομένων

2. Η μη εξουσιοδοτημένες κλήσεις, SMS και χρεώσεις

Οι επιτιθέμενοι που εκμεταλλευόμενοι τις αδυναμίες της ανθρώπινης φύσης και το μοντέλο διανομής των εφαρμογών στις κινητές συσκευές μπορούν να μετατρέψουν τις απλές τηλεφωνικές κλήσεις και τα sms σε κλήσεις και sms πρόσθετου τέλους. Συμπεριλαμβάνοντας την λειτουργικότητα πριμοδοτημένης κλήσης ως εφαρμογή Trojan ο επιτιθέμενος μπορεί να υποχρεώσει το λογαριασμό του τηλεφώνου του θύματος και να πληρωθεί από τις εταιρείες κινητής τηλεφωνίας. Επιπρόσθετος στόχος για τους επιτιθέμενους αποτελεί το γεγονός ότι οι κινητές συσκευές μπορούν να χρησιμοποιηθούν για την αγορά προϊόντων, πραγματικών και εικονικών, με το κόστος αυτό να χρεώνεται στο λογαριασμό του πελάτη. Μια άλλη χρήση μη εξουσιοδοτημένης αποστολής μηνυμάτων SMS είναι ως φορέας διάδοσης worms. Όταν μια συσκευή έχει μολυνθεί από έναν ιό τύπου worm μπορεί να στείλει μηνύματα sms σε όλες τις επαφές της με περιεχόμενο ένα σύνδεσμο ώστε να ξεγελαστούν οι παραλήπτες και να μεταβούν στον κακόβουλο ιστοχώρο και να εγκαταστήσουν το **worm.**

3. Μη εξουσιοδοτημένη σύνδεση δικτύου (Command & Control, bot)

Τα Spyware όπως πολλές άλλες κακόβουλες λειτουργικότητες συνήθως απαιτεί διαφυγή (=ικανότητα να μπορεί να επικοινωνήσει με τον επιτιθέμενο) ώστε να είναι επωφελής για τον εισβολέα. Δεδομένου ότι οι κινητές συσκευές έχουν σχεδιαστεί για επικοινωνία, υπάρχουν πολλοί πιθανοί φορείς που μία κακόβουλη εφαρμογή μπορεί να χρησιμοποιήσει ώστε να στείλει τα δεδομένα στον εισβολέα. Ένα πλήρης λειτουργικό malware συχνά επιτρέπει στον επιτιθέμενο να δίνει εντολές στο spyware. Εντολές όπως, να ανοίξει το μικρόφωνο ή να υποκλέψει ένα αρχείο δεδομένων σε μια συγκεκριμένη χρονική στιγμή.

Τα ακόλουθα αποτελούν παραδείγματα καναλιών επικοινωνίας που οι επιτιθέμενοι μπορούν να χρησιμοποιήσουν για τη διαφυγή και για **επιθέσεις command and control:**
a. E-mail
b. SMS
c. HTTP GET / POST
d. TCP socket
e. UDP socket
f. DNS exfiltration
g. Bluetooth
h. Blackberry Messenger

4. Απομίμηση UI

Οι επιθέσεις phishing σε υπολογιστές λειτουργούν ξεγελώντας τον χρήστη να κάνει κλικ σε ένα σύνδεσμο στο brwoser που τους πλοηγεί σε μια ψευδή ιστοσελίδα που μιμείται το User Interface (UI) κάποιας υπηρεσίας (πχ τράπεζας). Το ψεύτικο UI ζητά από το χρήστη να πληκτρολογήσει τα διαπιστευτήρια του. Έτσι ο επιτιθέμενος συλλέγει τα διαπιστευτήρια και τα χρησιμοποιεί για να παραστήσει το θύμα. Στις κινητές συσκευές υπάρχουν νέες ευκαιρίες για τους επιτιθέμενους να κατασκευάσουν ένα ψεύτικο UI. Αυτό συμβαίνει όταν μία υπηρεσία προσφέρεται και σε mobile εφαρμογή, παράλληλα με την web. Ο επιτιθέμενος συχνά δημιουργεί μία απομίμηση της πραγματικής και την δημοσιεύει ανάλογα. Ο χρήστης έχει την εντύπωση ότι κάνει λήψη της νόμιμης εφαρμογής ενώ στην πραγματικότητα λαμβάνει μία ψεύτικη εφαρμογή που κατά πάσα πιθανότητα θα του υποκλέψει τα διαπιστευτήρια, καθώς αυτός θα νομίζει ότι συνδέεται στην πραγματική υπηρεσία. Αυτό συνήθως επιτυγχάνεται με την ψεύτικη εφαρμογή να έχει το ίδιο UI με την αυθεντική. Παράδειγμα (Proxy/MITM 09Droid Banking apps)

5. Τροποποίηση συστήματος (rootkit, APN, τροποποποήση proxy)

Οι κακόβουλες εφαρμογές συχνά προσπαθούν να τροποποιήσουν την τρέχουσα κατάσταση του συστήματος για να αποκρύψουν την παρουσία τους. Αυτή η προσπάθεια καλείται “συμπεριφορά rootkit”.

Επίσης αυτές οι αλλαγές καθιστούν το σύστημα ευάλωτο σε συγκεκριμένες επιθέσεις. Όπως για παράδειγμα, η τροποποίηση της ρύθμισης παραμέτρων διακομιστή μεσολάβησης (proxy) ή η αλλαγή του APN (Access Point Name).

6. Λογική ή ωρολογιακή βόμβα[42]

Είναι οι κλασικές τεχνικές κερκόπορτας που εγείρουν κάποια κακόβουλη δραστηριότητα βασισμένες σε ένα συγκεκριμένο γεγονός ή σε κάποια συγκεκριμένη κατάσταση της συσκευής ή σε κάποιο προκαθορισμένο χρόνο.

7.2.2 Κατηγορία ευπαθειών

7. Διαρροή ευαίσθητων δεδομένων (ακούσια ή παράπλευρα)[43]

Μια διαρροή ευαίσθητων δεδομένων μπορεί να είναι είτε ακούσια ή παράπλευρη. Σε μία νόμιμη και όχι κακόβουλη εφαρμογή που διαχειρίζεται διαπιστευτήρια, υπάρχει η πιθανότητα κατά την κατασκευή της να έχουν εφαρμοστεί ελλείπως τα μέτρα προστασίας, με αποτέλεσμα τα ευαίσθητα δεδομένα που πιθανώς χειρίζεται να εκτίθενται σε κίνδυνο.

Ευαίσθητα δεδομένα όπως:

1. a. Τοποθεσία
   b. Πληροφορίες του χρήστη: όνομα, αριθμός, ID της συσκευής
   c. Διαπιστευτήρια ελέγχου ταυτότητας
   d. Token εξουσιοδότησης.

42 CWE-511: Logic/Time Bomb

43 CWE-200: Information Exposure

8. Επισφαλής αποθήκευση ευαίσθητων δεδομένων[44]

Οι εφαρμογές στις κινητές συσκευές συχνά αποθηκεύουν ευαίσθητα δεδομένα όπως αριθμούς τραπεζικών λογαριασμών, διάφορα PIN πληρωμών, αριθμούς πιστωτικών καρτών και κωδικούς πρόσβασης online υπηρεσιών.

Τα ευαίσθητα δεδομένα πρέπει πάντα να αποθηκεύονται κρυπτογραφημένα έτσι ώστε οι επιτιθέμενοι να μην μπορούν να ανακτήσουν απλά αυτά τα δεδομένα από το σύστημα αρχείων της συσκευής. Τέλος οποιαδήποτε αποθήκευση ευαίσθητων δεδομένων χωρίς κρυπτογράφηση σε αφαιρούμενα μέσα, όπως μια κάρτα micro SD είναι ιδιαίτερα επικίνδυνη.

9. Επισφαλής μετάδοση ευαίσθητων δεδομένων[45]

Είναι σημαντικό τα ευαίσθητα δεδομένα να κρυπτογραφούνται κατά τη μετάδοση προς αποφυγή υποκλοπής από επιτιθέμενους. Οι κινητές συσκευές είναι ιδιαίτερα ευπαθής, λόγω της αποκλειστικής χρήσης ασύρματης επικοινωνίας και συχνά δημόσιων Wi-Fi, το οποίο είναι γνωστό ως ανασφαλής. Το SSL είναι ένας από τους καλύτερους τρόπους για να διασφαλιστούν τα ευαίσθητα δεδομένα υπό μεταφορά σε ένα δίκτυο. Όμως ακόμα και αν η εφαρμογή υλοποιεί SSL υπάρχει περίπτωση να πέσει θύμα επίθεσης υποβάθμισης (downgrade attack-[46]) εάν επιτρέπει εναλλαγή του HTTPS σε HTTP, όταν το HTTPS δεν υποστηρίζεται. Ένας άλλος τρόπος που το SSL αποτυγχάνει είναι η περίπτωση που η εφαρμογή δεν τερματίζει όταν τα πιστοποιητικά δεν είναι έγκυρα. Όπου επιτρέπει επιθέσεις man-in-the-middle.

10. Ενσωμάτωση κωδικών / κλειδιών στον κώδικα [47]

Η ενσωμάτωση των κωδικών πρόσβασης ή των κλειδιών μέσα στον πηγαίο κώδικα της εφαρμογής πραγματοποιείται κάποιες φορές από τους προγραμματιστές για να κάνουν την εφαρμογή πιο εύκολη στην υλοποίηση, την υποστήριξη ή και το debug. Στην περίπτωση του reverse engineering ο κωδικός μπορεί να ανακαλυφθεί καθιστώντας αναποτελεσματική την ασφάλεια της εφαρμογής και των συστημάτων που επικυρώνει να με αυτόν τον κωδικό.

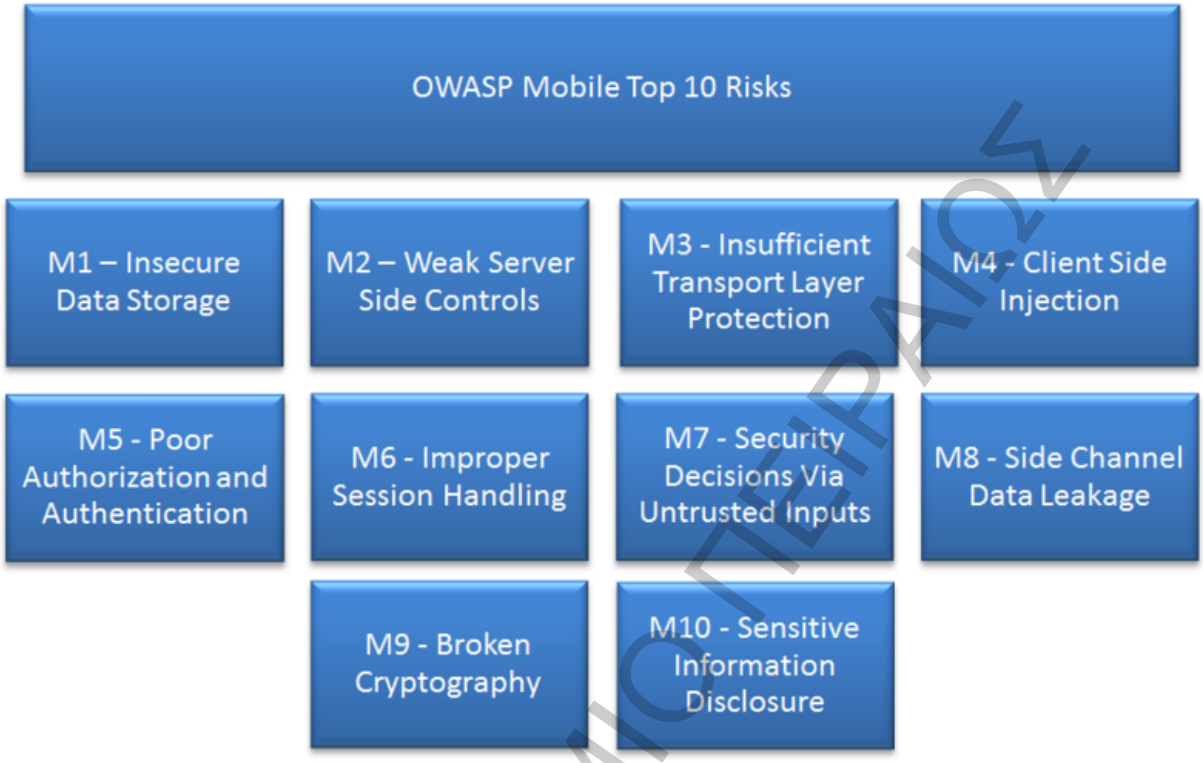
44 CWE-312: Cleartext Storage of Sensitive Information

45 CWE-319: Cleartext Transmission of Sensitive Information

46 Είναι η επίθεση που στοχεύει στην υποβάθμιση ενός κρυπτογραφικού μηχανισμού έτσι ώστε να μην γίνεται η κρυπτογράφηση και τα μηνύματα να μεταφέρονται ως απλό κείμενο.

47 **CWE-798: Use of Hard-coded Credentials**

7.3 Top 10 των κινδύνων από τον OWASP



Εικόνα 106, OWASP TOP 10

OWASP:

1. Επισφαλής αποθήκευση ευαίσθητων δεδομένων
2. Αδυναμίες ασφάλειας από την πλευρά του Server
3. Επισφαλής μετάδοση δεδομένων
4. Επιθέσεις από τον Client
5. Ανεπαρκείς μηχανισμοί αυθεντικοποίησης και ταυτοποίησης
6. Ακατάλληλη διαχείριση συνόδου.
7. Αποφάσεις Ασφαλείας από αναξιόπιστες εισόδους
8. Διαρροή δεδομένων
9. Ευάλωτη κρυπτογραφία
10. Ενσωμάτωση ευαίσθητων δεδομένων

([87], [88])

1. Επισφαλής αποθήκευση ευαίσθητων δεδομένων

Αφορά τα ευαίσθητα δεδομένα που διατηρούνται απροστάτευτα. Ισχύει για δεδομένα που αποθηκεύονται τοπικά ή συγχρονισμένα σε ένα Cloud. Γενικά αυτό είναι αποτέλεσμα, μη κρυπτογράφησης των δεδομένων που αποθηκεύονται στην κρυφή μνήμη όταν αυτά δεν προορίζονται για μακροχρόνια αποθήκευση. Μη ορθός σχεδιασμός και διανομή δικαιωμάτων ή χρήση global δικαιωμάτων. Επίσης της μη αξιοποίησης των βέλτιστων πρακτικών ασφάλειας της πλατφόρμας

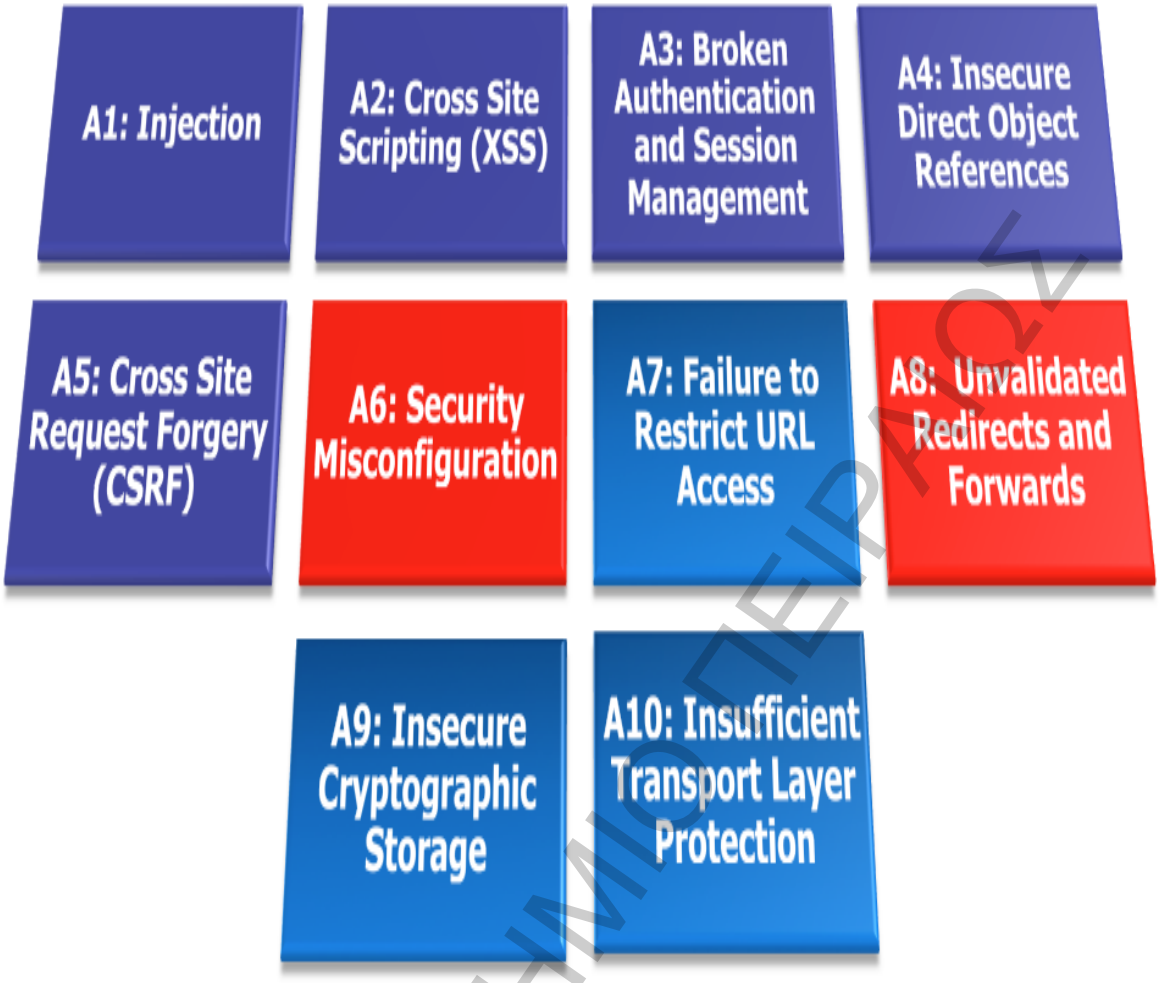
Παρενέργειες	Υποδείξεις πρόληψης – αντίμετρα
Απώλεια εμπιστευτικότητας των δεδομένων που χάνονται Αποκάλυψη διαπιστευτηρίων Παραβίαση ιδιωτικότητας Μη συμμόρφωση	-Στην συσκευή αποθηκεύεται μόνο ό, τι είναι απολύτως απαραίτητο -Ποτέ δεν πρέπει να χρησιμοποιείται κοινόχρηστους χώρους για αποθήκευση (πχ-SD κάρτα) -Πρέπει να γίνεται αξιοποίηση των μηχανισμών ασφαλείας της πλατφόρμας και τα παρεχόμενα APIs κρυπτογράφησης. -Μην χορηγείται στα αρχεία δικαιώματα τύπου : world readable ή world writeable. -Τα ευαίσθητα δεδομένα πρέπει να εντοπίζονται και να προστατεύονται στη φορητή συσκευή. -Τα διαπιστευτήρια πρέπει να διαχειρίζονται με ασφάλεια στη συσκευή

Πίνακας 6, Επισφαλής αποθήκευση ευαίσθητων δεδομένων

2. Αδυναμίες ασφάλειας από την πλευρά του Server

Η παράγραφος αυτή αφορά τις υπηρεσίες backend, δεν έχει να κάνει με τις συσκευές. Αντιθέτως έχει να κάνει με την ασφάλεια του Server και το γεγονός ότι ακόμα και ένας διαπιστευμένος client δεν μπορεί να είναι απολύτως αξιόπιστος. Τα ήδη υπάρχοντα Controls, web app top 10 και Cloud top 10 του OWASP έχουν εφαρμογή σε αυτό την παράγραφο.

web app top 10:



Εικόνα 107, OWASP Web app Top 10

Εικόνα 108, OWASP Cloud top 10

Παρενέργειες	Υποδείξεις πρόληψης – αντίμετρα
Απώλεια εμπιστευτικότητας των δεδομένων που χάνονται Απώλεια της *ακεραιότητας των* δεδομένων	Κατανόηση των πρόσθετων κινδύνων κινητών εφαρμογών που ενσωματώνονται στις υπάρχουσες αρχιτεκτονικές. Αξιοποίηση της γνώσης που υπάρχει ήδη: Web OWASP Top 10, Top Cloud 10, Web Services Cheat sheets, development guides, ESAPI, κτλ.. Οι backend APIs (υπηρεσίες) και η πλατφόρμα (ο server) πρέπει να διατηρούνται όσο το δυνατόν πιο ασφαλή.

Πίνακας 7, Αδυναμίες ασφάλειας από την πλευρά του Server

3. Επισφαλής μετάδοση δεδομένων

Σε κάποιες περιπτώσεις υπάρχει ακόμα και πλήρης έλλειψη κρυπτογράφησης για τα δεδομένα που μεταδίδονται. Ή ακόμα όταν υπάρχει κρυπτογράφηση που γίνεται με αδύναμους αλγορίθμους είτε με λάθος υλοποίηση, αγνοώντας τις προειδοποιήσεις ασφαλείας και αγνοώντας τα σφάλματα επικύρωσης του πιστοποιητικού. Κάποιοι αλγόριθμοι όταν αποτυγχάνουν επιστρέφουν απλό κείμενο.

Παράδειγμα : Google Client Login Authentication Protocol Authorization header sent over HTTP.

Όταν οι χρήστες συνδέονται μέσω Wi-Fi, οι εφαρμογές αυτόματα αποστέλλουν ένα token σε μια προσπάθεια να συγχρονιστούν αυτόματα τα δεδομένα από το διακομιστή. Στο παρελθόν (μέχρι 06/2011) αν κάποιος υπέκλεπτε αυτή την τιμή μπορούσε να μιμηθεί τον χρήστη [86].

Παρενέργειες	Υποδείξεις πρόληψης – αντίμετρα
Επιθέσεις Man-in-the-middle Αλλοίωση δεδομένων κατά τη μεταφορά Απώλεια εμπιστευτικότητας των δεδομένων	Όλα τα ευαίσθητα δεδομένα που φεύγουν από τη συσκευή πρέπει να είναι κρυπτογραφημένα. Αυτό περιλαμβάνει δεδομένα μέσω δικτύων μεταφοράς, WiFi, κτλ.. Όταν εγείρονται εξαιρέσεις ή προβλήματα ασφάλειας δεν πρέπει να αγνοούνται.

Πίνακας 8, Επισφαλής μετάδοση δεδομένων

4. Επιθέσεις από τον Client

Σε μία εφαρμογή ένας διαπιστευμένος client, μπορεί να:

-Χρησιμοποιήσει τις libraries σε web apps ή σε συνδυασμό

web/native εφαρμογές.

-Εκτελέση

XSS και HTML Injection

SQL Injection

όπως και παγίδευση της συσκευής με phone dialer + SMS

παγίδευση της συσκευής με in-app payments

Παρενέργειες	Υποδείξεις πρόληψης – αντίμετρα
Περιορισμός ασφάλειας της συσκευής Χρηματική απάτη κλιμάκωση προνομίων (privilege escalation)	Τα δεδομένα που προέρχονται από τον client οφείλουν να ελέγχονται και να απορρίπτονται ανάλογα, πριν αποθηκευτούν ή καταστούν κάποιας επεξεργασίας.

	Για τις κλήσεις της βάσεις συνιστώνται τα prepared statements για την αποφυγή sql injection. Ελαχιστοποίηση των ευαίσθητων φυσικών δυνατοτήτων που συνδέονται με υβριδική λειτουργικότητα web-mobile app.

Πίνακας 9, Control. Επιθέσεις από τον Client

5. Ανεπαρκείς μηχανισμοί αυθεντικοποίησης και ταυτοποίησης

Ορισμένες εφαρμογές βασίζονται αποκλειστικά στις αμετάβλητες τιμές (IMEI, IMSI, UUID). Το ότι είναι αμετάβλητες όμως δεν σημαίνει ότι δεν έχουν υποκλαπεί. Στις περιπτώσεις service ή αντικατάστασης της συσκευής μπορεί να αντικατασταθούν, καθιστώντας την εφαρμογή μη προσβάσιμη στον χρήστη. Η προσθήκη εξειδικευμένης πληροφορίας για αυθεντικοποίηση είναι χρήσιμη, αλλά δεν είναι αλάνθαστη.

Παρενέργειες	Υποδείξεις πρόληψης – αντίμετρα
Κλιμάκωση προνομίων (privilege escalation) Μη εξουσιοδοτημένη πρόσβαση	Αυτές οι εξειδικευμένες πληροφορίες ενίσχυσης της αυθεντικοποίησης μπορεί να βελτιώσουν τα πράγματα, αλλά μόνο ως μέρος μίας υλοποίησης με πολλούς παράγοντες. Ποτέ μόνο με έναν παράγοντα (πχ το subscriber ID)

Πίνακας 10, Ανεπαρκείς μηχανισμοί αυθεντικοποίησης και ταυτοποίησης

6. Ακατάλληλη διαχείριση συνόδου

Τα Mobile app sessions, για λόγους ευκολίας και ευχρηστίας, γενικά κρατάνε για μεγαλύτερο χρονικό διάστημα

οι εφαρμογές διατηρούν sessions για :

HTTP cookies

OAuth tokens

SSO authentication services

Η χρήση του sid της συσκευής ως session token είναι πολύ επικίνδυνη τακτική που δυστυχώς εφαρμόζεται συχνά.

Παρενέργειες	Υποδείξεις προλήψεις – αντίμετρα
κλιμάκωση προνομίων (privilege escalation) Μη εξουσιοδοτημένη πρόσβαση Παράκαμψη αδειοδότησης και πληρωμών	Μία καλή τακτική αποτελεί η ανα-αυθεντικοποίηση σε τακτά χρονικά διαστήματα. Τα tokens πρέπει να μπορούν να ανακληθούν γρήγορα σε περίπτωση απώλειας / κλοπής της συσκευής. Συνιστάται η χρήση υψηλής εντροπίας στους

	μηχανισμούς δημιουργίας token.

Πίνακας 11, Ακατάλληλη διαχείριση συνόδου

7. Αποφάσεις ασφαλείας από αναξιόπιστες εισόδους

Μπορούν να χρησιμοποιηθούν για την παράκαμψη των δικαιωμάτων του μοντέλου ασφάλειας:

Android- Abusing Intents

Κακόβουλες εφαρμογές

Επιθέσεις από τον Client

Παρενέργειες	Υποδείξεις προλήψεις – αντίμετρα
Κατανάλωση χρηματικών πόρων Consuming paid resources Διήθηση πληροφορίας (εισαγωγή - υποκλοπή) κλιμάκωση προνομίων (privilege escalation)	Πριν τις ανάλογες κλήσεις είναι καλό να ελέγχονται ξανά τα δικαιώματα του χρήστη που εκτελεί την κλήση. Στις περιπτώσεις που οι έλεγχοι δικαιωμάτων δεν μπορούν να γίνουν, πρέπει να εξασφαλίζονται πρόσθετα μέτρα που να είναι προαπαιτούμενα για την έναρξη ευαίσθητων ενεργειών.

Πίνακας 12, Αποφάσεις ασφαλείας από αναξιόπιστες εισόδους

8. Διαρροή δεδομένων

Ο συνδυασμός απενεργοποίησης χαρακτηριστικών της πλατφόρμας με προγραμματιστικά λάθη μπορεί να προκαλέσει την αποθήκευση ευαίσθητων πληροφοριών σε μη επιθυμητά μέρη:

Web cache

Keystroke logging

Screenshots (ie- iOS backgrounding)

Logs (system, crash)

Κατάλογοι Temp

```
try {
    userInfo = client.validateCredentials(userName, password);
    if (userInfo.get("success").equals("true"))
        launchHome(v);
    else {
        Log.w("Failed login", userName + " " + password);
    }

} catch (Exception e) {
    Log.w("Failed login", userName + " " + password);
}
```

Εικόνα 109, Κακή πρακτική. Καταγραφή των άστοχων username, pasword

Παρενέργειες	Υποδείξεις πρόληψης – αντίμετρα
Τα δεδομένα διατηρούνται επ' αόριστον Παραβίαση ιδιωτικότητας	Διαπιστευτήρια ή άλλα ευαίσθητα δεδομένα δεν πρέπει ποτέ να καταγράφονται στα αρχεία καταγραφής του συστήματος(logs). Κατά τη λήψη screenshots είναι καλό να αφαιρούνται τα ευαίσθητα δεδομένα. Σε περιπτώσεις υπηρεσιών καταγραφής εισόδου πρέπει να αναγνωρίζεται το σημείο εισαγωγής ευαίσθητων δεδομένων και να παύει προσωρινά η υπηρεσία. Όλα τα αρχεία που παράγονται κατά την διάρκεια εκτέλεσης μίας εφαρμογής, οφείλουν να έχουν μελετηθεί για τυχών παραλείψεις.

Πίνακας 13, Διαρροή δεδομένων

9. Ευάλωτη κρυπτογραφία

Αυτή η ενότητα χωρίζεται σε δύο κύριες κατηγορίες:

α) Ελαττωματικές ή εσφαλμένες υλοποιήσεις δυνατών αλγορίθμων.

β) Παραμετροποιημένες εφαρμογές ανίσχυρων κρυπτογραφικών λύσεων.

Encoding != encryption

Obfuscation != encryption

Serialization != encryption

Παρενέργειες	Υποδείξεις_πρόληψης– αντίμετρα
Απώλεια εμπιστευτικότητας των δεδομένων κλιμάκωση προνομίων (privilege escalation) Παράκαμψη επιχειρηματικής λογικής	-Η αποθήκευση του κλειδιού μαζί με τα κρυπτογραφημένα δεδομένα αποτελεί σοβαρό σφάλμα υλοποίησης, στην ουσία αναιρεί την κρυπτογράφηση. -Η δημιουργία ενός νέου κρυπτογραφικού αλγορίθμου προσαρμοσμένου στις ανάγκες τις εφαρμογής δεν μπορεί να συγκριθεί με τους αλγόριθμους που ήδη υφίστανται και έχουν ανταπεξέλθει σε εξαντλητικούς ελέγχους και επιθέσεις. - Συνίσταται η χρήση των βιβλιοθηκών και των πακέτων που προσφέρονται από την πλατφόρμα, όπου αυτό είναι δυνατό. -Δεν συνίσταται “η ανακάλυψη του τροχού”.

Πίνακας 14, Ευάλωτη κρυπτογραφία

10. Ενσωμάτωση ευαίσθητων δεδομένων

Στο owasp mobile top 10 διαφοροποιείται το M1 με το M10 το M1 αναφέρεται στην αποθήκευση ενώ το M10 στην ενσωμάτωση στον κώδικα. Οι Android εφαρμογές υπόκεινται σε reverse enginiering με σχετική ευκολία. Οι λύσεις Code obfuscation μπορούν να ανεβάσουν λίγο το επίπεδο προστασίας αλλά όχι ικανοποιητικά.

Κοινά στόχοι ευρήματα αποτελούν:

Κλειδιά API

Κωδικοί

Ευαίσθητη επιχειρηματική λογική

Παρενέργειες	Υποδείξεις πρόληψης – αντίμετρα
Διαρροή διαπιστευτηρίων Γνωστοποίηση σημαντικών παραμέτρων - χαρακτηριστικών	-Τα Private κλειδιά δεν πρέπει σε καμία περίπτωση να βρίσκονται στον client. -Η εσωτερική και η επιχειρηματική λογική πρέπει επίσης να είναι στον server. - Δεν υπάρχει κανένας λόγος να αποθηκεύονται οι κωδικοί στον κώδικα.

Πίνακας 15, Ενσωμάτωση ευαίσθητων δεδομένων

7.4 Σύγκριση OWASP και VERACODE

Vera	OWASP
1. Παρακολούθηση της δραστηριότητας και ανάκτηση δεδομένων	
2. Η μη εξουσιοδοτημένες κλήσεις, SMS και χρεώσεις	
3. Μη εξουσιοδοτημένη σύνδεση δικτύου (Command & Control, bot)	
4. Απομίμηση UI	
5. Τροποποίηση συστήματος	
6. Λογική ή ωρολογιακή βόμβα	
7. Διαρροή ευαίσθητων δεδομένων	8. Διαρροή δεδομένων
8. Επισφαλής αποθήκευση ευαίσθητων δεδομένων	1. Επισφαλής αποθήκευση ευαίσθητων δεδομένων
9. Επισφαλής μετάδοση ευαίσθητων δεδομένων	3. Επισφαλής μετάδοση δεδομένων
10. Ενσωμάτωση κωδικών / κλειδιών στον κώδικα	10. Ενσωμάτωση ευαίσθητων δεδομένων

Πίνακας 16, Veracode - OWASP

Το veracode Top 10 έχει βασιστεί στο OWASP Top 10. Η κυριότερη διαφορά που έχουν είναι ότι το veracode συμπεριλαμβάνει την κατηγορία της κακόβουλης λειτουργικότητας (τα έξι πρώτα controls), ενώ το OWASP δεν εξετάζει αυτή την σκοπιά. Επίσης το contol vera 7 αποτελεί την συγχώνευση των OWASP 1 και 8. Από την στιγμή που στο OWASP το 1. “Επισφαλής αποθήκευση ευαίσθητων δεδομένων” καλύπτει και το πλαίσιο της απώλειας ευαίσθητων δεδομένων. Και τα δύο πρότυπα αποτελούν ολοκληρωμένες λύσεις. Με το OWASP Top10 να είναι ελαφρώς πιο αφηρημένο όπως τα περισσότερα πρότυπα που προτείνει ο OWASP ενώ το Vera Top 10 να είναι πιο εφαρμοσμένο. Σαφώς σε περιπτώσεις ανάπτυξής εφαρμογών που χειρίζονται ευαίσθητα δεδομένα, η χρήση προτύπων για την ανάπτυξη αποτελεί επιτακτική ανάγκη.

Επίλογος

Το Android αναπτύχθηκε ώριμα από άποψη ασφάλειας υιοθετώντας τις βέλτιστες υπάρχουσες πρακτικές από το Linux και την Java. Φυσικά το νέο αυτό λειτουργικό αντιμετώπισε καινούρια θέματα ασφάλειας τα οποία καλύπτονται με την πάροδο του χρόνου με τις καινούριες εκδόσεις του λειτουργικού. Επίσης από την έρευνα προέκυψε ότι οι εφαρμογές αποτελούν το κύριο μέσο μετάδοσης των ιών, και μεγάλο μερίδιο ευθύνης έχουν οι προγραμματιστές. Οι όποιοι οφείλουν να ορίζουν την αρχιτεκτονική ασφάλειας της εφαρμογής τους ώστε αυτή να είναι όσο το δυνατόν πιο απλή και σαφής. Εν συνεχεία, καταδείχθηκε ότι στο Android η πιο συχνή μορφή malware είναι οι πλαστές εφαρμογές όπου ο δημιουργός του ιού εισάγει τον κακόβουλο κώδικα σε κάποια σημεία της αυθεντικής δημοφιλούς εφαρμογής και την αναδημοσιεύει ως μία νέα δωρεάν έκδοση της αυθεντικής σε εναλλακτικά market. Τα malware έχουν την τάση να εξελίσσονται σε πιο ευφυή τόσο στον τρόπο επίθεσης τους αλλά και στον τρόπο απόκρυψης τους. Παραδόξως δεν βασίζονται ιδιαίτερα στην εκμετάλλευση root exploits αλλά το μεγαλύτερο ποσοστό βασίζεται στις αδυναμίες της ανθρώπινης φύσης σε συνδυασμό με την άγνοια και την αφέλεια των χρηστών. Όμως και τα malware έχουν τις δικές τους αδυναμίες από την στιγμή που συχνά χρειάζεται να εγγράφονται σε ένα μεγάλο αριθμό γεγονότων για να έχουν την δυνατότητα να ξεκινάνε αξιόπιστα και γρήγορα τα κακόβουλα φορτία τους. Επίσης σημειώνεται ότι ο μέσος όρος των δικαιωμάτων που αιτούνται οι mlware εφαρμογές είναι 11 ενώ ο μέσος όρων των νόμιμων εφαρμογών είναι 4. Από την έρευνα προέκυψε ακόμα πως τα δύο κυρίαρχα χαρακτηριστικά που ουσιαστικά είναι αδυναμίες των malware στο Android και πως προδιαθέτουν για την κακόβουλη διάθεση τους. Σαν αντίμετρο οι συγγραφείς malware αρχίζουν να ενσωματώνουν ρουτίνες για ελέγχουν εάν τα malware τους εκτελούνται μέσα σε ένα περιβάλλον ανάλυσης. Οι αναλυτές έχουν ένα δύσκολο έργο που καθημερινά προκύπτουν νέες προκλήσεις, καθώς εισάγονται νέες επιθέσεις μηχανισμοί κρυπτογραφίας και απόκρυψης. Τα εργαλεία της ανάλυσης είναι ικανοποιητικά, αξιόλογα και εύχρηστα. Να σημειωθεί πως η πλατφόρμα SDK προσφέρεται στις υπηρεσίες αυτής της αντιμετώπισης των απειλών. Τέλος τα πρότυπα της veracode ή του OWASP αποτελούν αξιόλογες λύσεις για ανάπτυξη ασφαλών εφαρμογών ειδικά όταν πρόκειται για εφαρμογές που χειρίζονται ευαίσθητα δεδομένα η υιοθέτηση τους ως πρότυπα είναι απαραίτητη.

Βιβλιογραφία

[1] Android

www.android.com/

[2] Android Σύστημα αρχείων

πηγή http://www.vogella.de/articles/AndroidFileSystem/article.html

[3] About Android memory

http://android.modaco.com/content/htc-hero-hero-modaco-com/295688/free-ram-on-htc-hero-sorry-bit-of-a-newbie/#entry1110783

[4] Android memory types

http://androidforums.com/htc-hero-sprint/13022-ram-rom-what.html

[5] Dalvik VM

David Ehringer, March, 2010 The dalvik virtual machine architecture

[6] How to Access Android Bootloader or System Recovery Mode

http://www.mydigitallife.info/how-to-access-android-bootloader-or-system-recovery-mode/

[7] Google Play - Android Market

http://en.wikipedia.org/wiki/Android_Market

[8] Android Developers

developer.android.com/

[9] Android developers sdk.

http://developer.android.com/sdk/android-1.1.html.

[10] Mobile Application Security on Android Black Hat 2009

http://www.blackhat.com/presentations/bh-usa-09/BURNS/BHUSA09-Burns-AndroidSurgery-PAPER.pdf

[11] Notepad Tutorial

http://code.google.com/android/intro/tutorial.html

[12] David Ehringer "THE DALVIK VIRTUAL MACHINE ARCHITECTURE" March, 2010

[13] AndroidManifestPermission protectionLevel

http://code.google.com/android/reference/android/R.styleable.html#AndroidManifestPermission_protectionLevel

[14] Intent

http://developer.android.com/reference/android/content/Intent.html

[15] Activities

http://developer.android.com/guide/components/activities.html

[16] BroadcastReceiver | Android Developers

http://developer.android.com/reference/android/content/BroadcastReceiver.html

[17] Service | Android Developers

http://developer.android.com/reference/android/app/Service.html

[18] Content Providers | Android Developers

http://developer.android.com/guide/topics/providers/content-providers.html

[19] Binder | Android Developers

http://developer.android.com/reference/android/os/Binder.html

[20] Android Security : Files and Preferences

http://programming4.us/mobile/1302.aspx

[21] A Survey of Mobile Malware in the Wild

Adrienne Porter Felt, Matthew Finifter, Erika Chin, Steven Hanna, and David Wagner

University of California, Berkeley

http://www.cs.berkeley.edu/~afelt/mobilemalware.pdf

[22] Yajin Zhou Xuxian Jiang "Dissecting Android Malware: Characterization and Evolution", 2011

[23] Android.Bgserv Found on Fake Google Security Patch.

http://www.symantec.com/connect/blogs/androidbgserv-

found-fake-google-security-pat

[24] GGTracker Technical Tear Down.

http://blog.mylookout.

com/wp-content/uploads/2011/06/GGTracker-Teardown

Lookout-Mobile-Security.pdf.

[25] Malicious QR Codes Pushing Android Malware.

https://www.securelist.com/en/blog/208193145/Its time for malicious QR codes.

[26] First SpyEye Attack on Android Mobile Platform now in the Wild.

https://www.trusteer.com/blog/first-spyeye-attack-android-mobile-platform-now-wild.

[27] ZeuS-in-the-Mobile - Facts and Theories.

http://www.securelist.com/en/analysis/204792194/ZeuS in the Mobile Facts and Theories.

[28] Using QR tags to Attack SmartPhones (Attaging).

http://kaoticoneutral.blogspot.com/2011/09/using-qr-tags-to-attack-smartphones 10.html.

[29] Security Alert: New Sophisticated Android Malware Droid-

KungFu Found in Alternative Chinese App Markets.

http://www.csc.ncsu.edu/faculty/jiang/DroidKungFu.html.

[30] LeNa (Legacy Native) Teardown. http://blog.mylookout.

com/wp-content/uploads/2011/10/LeNa-Legacy-Native-

Teardown Lookout-Mobile-Security1.pdf

[31] DroidKungFu Utilizes an Update Attack.

http://www.f-secure.com/weblog/archives/00002259.html.

[32] An Analysis of the AnserverBot Trojan.

http://www.csc.ncsu.edu/faculty/jiang/pubs/AnserverBot Analysis.pdf.

[33] What are the exact mechanisms/flaws exploited by the "rage against the cage" and "z4root" Android exploits?

http://www.quora.com/What-are-the-exact-mechanisms-flaws-exploited-by-the-rage-against-the-cage-and-z4root-Android-exploits

[34] AVG Antivirus App for Smartphones

www.avg.com/antivirus-for-android

[35] Lookout Mobile Security.

https://www.lookout.com/.

[36] Norton Mobile Security

mobilesecurity.norton.com/

[37] TrendMicro.

www.trendmicro.com/

[38] SANS Institute InfoSec "Reading Room Dissecting Andro Malware" 2011

[39] Rajdeep Chakraborty, "Detailed analysis of the continuously evolving threat of Malwares", Retrieved

From: http://www.malwareinfo.org/library/whitepapers/MalwareAnalysisHow2.pdf

[40] Malware Analysis Basics

http://www.porcupine.org/forensics/forensic-discovery/chapter6.html

[41] Dennis Distler, "Malware Analysis: An Introduction",

http://www.sans.org/reading_room/whitepapers/malicious/malware analysis-introduction_2103,

[42] Performing Android malware analysis

http://blog.secfence.com/2012/05/performing-android-malware-analysis/

[43] Notepad++ Home

notepad-plus-plus.org/

[44] Extract RAR Files with WinZip®

www.winzip.com/lanrar.htm

[45] VirusTotal

https://www.virustotal.com/

[46] Apktool

http://code.google.com/p/android-apktool/

[47] Axmlprinter

http://code.google.com/p/meinvpic/downloads/list?q=label:AXMLPrinter

[48] Ded

http://siis.cse.psu.edu/ded/

[49] Jad

http://en.wikipedia.org/wiki/JAD_%28JAva_Decompiler%29

[50] Smali/Baksmali

http://code.google.com/p/smali/

[51] Mobile Sandbox

http://www.mobile-sandbox.com/

[52] IDA pro

http://www.hex-rays.com/products/ida/6.1/index.shtml

[53] Dex2jar

http://code.google.com/p/dex2jar/

[54] Dexdump

http://code.google.com/p/dex-decomplier/

[55] JD-GUI

http://java.decompiler.free.fr/?q=jdgui

[56] Androguard

http://code.google.com/p/androguard/

[57] APKinspector

https://github.com/honeynet/apkinspector/

[58] Droidbox

http://code.google.com/p/droidbox/

[59] Android SDK

http://developer.android.com/sdk/index.html

[60] Wireshark · Go deep.

www.wireshark.org/

[61] Android Reverse Engineering (A.R.E.)

https://redmine.honeynet.org/projects/are/wiki

[62] OSAF Virtual Machine

http://osaf-community.org/wiki/tiki-index.php?page=Installation+of+OSAF+VM+File

+

http://sourceforge.net/projects/osaftoolkit/files/latest/download?utm_expid=6384-3&utm_referrer=http%3A%2F%2Fosaf-community.org%2F

[63] Building a Malware Analysis Lab

http://www.windowsecurity.com/articles/Building-Malware-Analysis-Lab.html

[64] Best Practices: Organization

http://osaf-community.org/wiki/tiki-index.php?page=Best+Practice%3A+Organization

[65] Android Malware Analysis - Static Analysis of HolyF******Bible

http://malwarecrypt.blogspot.gr/2011/12/android-malware-analysis-static.html

[66] Contagio mobile- mobile malware mini dump

http://contagiominidump.blogspot.com/

[67] LeNa

http://www.mediafire.com/?qicgt2bqa7g1bzb

(password infected)

[68] Liveprints

http://www.mediafire.com/?qicgt2bqa7g1bzb

(password infected)

[69] sbooster

http://www.mediafire.com/?zgjks6kow79so8g

(password infected)

[70] waterfall3dLive.boa.liveWPcube

http://www.mediafire.com/?gcb0bxiicamh350

(password infected)

[71] Zitmo

http://www.mediafire.com/?jvds225lczxrv7o

(password infected)

[72] The AndroidManifest.xml File

http://developer.android.com/guide/topics/manifest/manifest-intro.html

[73] Use APKTool to Decompile, Edit, Translate and Recompile an APK

http://www.miui-au.com/add-ons/apktool/

[74] Android debug bridge.

http://developer.android.com/guide/developing/tools/adb.html

[75] [HOW-TO] Install & Use ADB tool | Android Debug Bridge | Drivers - Videos - Tutorial

http://forum.xda-developers.com/showthread.php?t=1474956

[76] APIMonitor -Installation and usage of DroidBox APIMonitor

http://code.google.com/p/droidbox/wiki/APIMonitor

[77] Beta Release of DroidBox for Android 2.3 and APIMonitor

http://www.honeynet.org/node/940

[78] 3D Waterfall Live Wallpaper- handySoft

https://play.google.com/store/apps/details?id=waterfall3dLive.liveWPcube&feature=search_result#?t=W251bGwsMSwxLDEsIndhdGVyZmFsbDNkTGl2ZS5saXZlV1BjdWJlIl0

[79] Zitmo Growing More Sophisticated, Prevalent in Android

http://threatpost.com/en_us/blogs/zitmo-growing-more-sophisticated-prevalent-android-100912

[80] Open Source Android Forensics Threat Index

http://osaf-community.org/threatindex.html

[81] Android Virtual Device Basics

http://osaf-community.org/wiki/tiki-index.php?page=Android+Virtual+Device+Basics

[82] Android SDK | Android Developers

developer.android.com/sdk/

[83] Android Security Overview -Introduction

http://source.android.com/tech/security/

[84] Veracode -Mobile App Top 10 List

http://www.veracode.com/blog/2010/12/mobile-app-top-10-list/

http://www.veracode.com/directory/mobileapp-top-10.html

[85] Projects/OWASP Mobile Security Project - Top Ten Mobile Risks

https://www.owasp.org/index.php/Projects/OWASP_Mobile_Security_Project_-_Top_Ten_Mobile_Risks

[86] Catching AuthTokens in the Wild The Insecurity of Google's ClientLogin Protocol

http://www.uni-ulm.de/in/mi/mitarbeiter/koenings/catching-authtokens.html

[87] OWASP Mobile Security Project -Contols

https://www.owasp.org/index.php/OWASP_Mobile_Security_Project?goback=.gde_3819235_member_54681631#tab%3DTop_Ten_Mobile_Controls

[88] OWASP Mobile Security Project - Model

https://www.owasp.org/index.php/OWASP_Mobile_Security_Project?goback=.gde_3819235_member_54681631#OWASP_Mobile_Threat_Model_Project

[89] J.F. DiMarzio "Android A Programmer's Guide" , 2008

[90] William Enck and Patrick McDaniel "Understanding Android's Security Framework", 2008

[91] Sheng -Fu Chen "Understanding Android Security" 2009

[92] Stephen A. Ridley "Hello Moto! Android Malware Reverse Engineering", 2011

[93] Anthony Desnos, Geoffroy Gueguen "Android: From Reversing to Decompilation", 2011

[94] David Barrera, H. Güne¸ Kayacık, P.C. van Oorschot, Anil Somayaji "A Methodology for Empirical Analysis of Permission-Based Security Models", 2011

[95] Leonid Batyuk, Markus Herpich, Seyit Ahmet Camtepe, Karsten Raddatz, Aubrey-Derrick Schmidt, Sahin Albayrak "Using Static Analysis for Automatic Assessment and Mitigation of Unwanted and Malicious Activities Within Android Applications", 2012

[96] Lok Kwong Yan, Heng Yin "DroidScope: Seamlessly Reconstructing the OS and Dalvik Semantic Views for Dynamic Android Malware Analysis", 2012

[97] Timothy Vidas, Daniel Votipka, Nicolas Christin "All Your Droid Are Belong To Us: A Survey of Current Android Attacks", 2012

[98] Kris Kendall "PRACTICAL MALWARE ANALYSIS"

[99] Dinesh Shetty "Demystifying the Android Malware"