ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
Τμήμα Πληροφορικής

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Android Development Worst Practices Analysis

Κωνσταντίνος Κεσόπουλος
13049

Επιβλέπων: Λέκτορας Κ. Πατσάκης

Πειραιάς, *Οκτώβριος 2015*

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)	(υπογραφή)	(υπογραφή)
Κωνσταντίνος Πατσάκης Λέκτορας	Γεώργιος Τσιχριτζής Καθηγητής	Ευθύμιος Αλέπης Επίκουρος Καθηγητής

Επιτελική Σύνοψη

Το αντικείμενο της τρέχουσας διπλωματικής είναι η ανάλυση εφαρμογών κινητών τηλεφώνων (με λειτουργικό android) και η αναζήτηση πρακτικών προγραμματισμού που θέτουν σε κίνδυνο τα προσωπικά δεδομένα του εκάστοτε χρήστη. Αυτά τα κενά ασφαλείας είναι αποτέλεσμα της μη συμμόρφωσης με τις βέλτιστες πρακτικές τις οποίες έχει δημοσιοποιήσει η εταιρία (google) που είναι υπεύθυνη για την ανάπτυξη του λειτουργικού συστήματος και των υπόλοιπων εργαλείων που χρησιμοποιούνται.

Για να γίνει η ανάκτηση του κώδικα των εφαρμογών δημιουργήθηκε ένα εργαλείο το οποίο αυτοματοποιεί τη διαδικασία απόκτησης του πηγαίου κώδικα από τα αρχεία εγκατάστασης και χρησιμοποιώντας εργαλεία ανάκτησης κειμένου αναζητεί τυχόν λανθασμένες πρακτικές προγραμματισμού στον κώδικα. Τα αποτελέσματα αποθηκεύονται σε αρχεία html στον εκάστοτε φάκελο της εφαρμογής.

Ευχαριστίες

Πρώτα από όλους θα ήθελα να ευχαριστήσω την οικογένεια μου η οποία με στήριξε με κάθε δυνατό τρόπο. Επίσης, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή κ. Κ. Πατσάκη για όλες τις συμβουλές και την ενθάρρυνση που μου έδωσε. Τέλος, θα ήθελα να ευχαριστήσω φίλους και συμφοιτητές για την ποικίλη βοήθεια που μου προσέφεραν.

Οκτώβριος 2015

Κωνσταντίνος Κεσόπουλος

Πίνακας περιεχομένων

ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

ΠΙΝΑΚΑΣ ΠΙΝΑΚΩΝ

Κεφάλαιο 1ο

1 Εισαγωγή

Όλο και περισσότερες πτυχές της ζωής μας περνούν πλέον το ψηφιακό κατώφλι. Ο τρόπος με τον οποίο δουλεύουμε, διασκεδάζουμε, κάνουμε τις αγορές μας και επικοινωνούμε έχουν αλλάξει δραστικά. Τα κινητά και τα tablet με τις διάφορες εφαρμογές αντικαθιστούν παλαιότερου τύπου βοηθήματα όπως το ημερολόγιο, η ατζέντα, το ταχυδρομείο. Βέβαια αυτά μπορεί να τα κάνει και ένας υπολογιστής. Η μεγάλη διαφορά όμως είναι ότι μπορούμε να τα έχουμε μαζί μας σε σύγκριση με τον υπολογιστή, που ακόμα και φορητός να είναι, είναι πολύ πιο δύσκολο να μετακινηθεί Παράλληλα, λόγω των πολλών αισθητήρων τους προσφέρουν και άλλα είδη διεπαφής ενώ συγχρόνως είναι σε θέση να γνωρίζουν τη θέση του χρήστη μέσω GPS κτλ. Επιπλέον, έχουμε φτάσει στο σημείο να αποθηκεύουμε πιστωτικές κάρτες, να προσθέτουμε λογαριασμούς τράπεζας και με την χρήση εφαρμογών (ηλεκτρονική αγορά ή με χρήση NFC σε καταστήματα) να κάνουμε τις αγορές ή τις πληρωμές μας. Ουσιαστικά λοιπόν αντικαθιστούμε το πορτοφόλι μας και κυρίως τα μετρητά. Ακόμη, τα κινητά πλέον έχουν την δυνατότητα να ελέγχουν τα δακτυλικά αποτυπώματά μας και να σκανάρουν την ίριδα του ματιού ώστε να κάνουν την αυθεντικοποίηση του χρήστη. Άρα, μπορούμε να πούμε ότι αντικαθιστούμε πλέον και την ταυτότητα για την διαδικασία των αγορών. Προκύπτει το ερώτημα, πόσο ασφαλή είναι όλα αυτά; Εμπιστευόμαστε τους παρόχους αυτών των υπηρεσιών; Ακολούθησαν τις βέλτιστες πρακτικές ώστε να εξασφαλίσουν την εμπιστευτικότητα των ευαίσθητων δεδομένων; Τί δεδομένα αποθηκεύονται; τί αποστέλλεται στις εταιρίες που ανέπτυξαν το λογισμικό και τί σε τρίτους;

Όσο μεγαλύτερο ποσοστό ανθρώπων χρησιμοποιεί αυτές τις εφαρμογές τόσο αυξάνεται η πιθανότητα να υπάρξουν κακόβουλες ενέργειες ώστε να εκμεταλλευτούν την κρισιμότητα των ευαίσθητων αυτών δεδομένων. Αυτές οι ενέργειες δεν αφορούν μόνο τα τοπικά αποθηκευμένα δεδομένα αλλά και αυτά τα οποία μεταδίδονται από την εκάστοτε εφαρμογή στους servers του developer. Για παράδειγμα στις dating που θα εξεταστούν στο πλαίσιο της παρούσας διπλωματικής γινόταν αποστολή (πολλές φορές και χωρίς κρυπτογράφηση) του στίγματος του χρήστη, κάτι το οποίο θέτει σε κίνδυνο την ιδιωτικότητά του. Γι' αυτό τον λόγο με τον καιρό θα γίνεται ακόμη πιο επιτακτική η ανάγκη για ασφάλεια σε όλες τις πτυχές των εφαρμογών, από την τοπική αποθήκευση δεδομένων και την μετάδοση μέχρι και τον τρόπο αποθήκευσης στον server. Επίσης, ακόμη και εν αγνοία των προγραμματιστών, που μπορεί να έχουν ως στόχο την συγγραφή ποιοτικού κώδικα, διάφορα API που χρησιμοποιούν έχουν περίεργη συμπεριφορά (όσον αφορά την ασφάλεια) και μπορούν να προκαλέσουν διαρροή δεδομένων. Χαρακτηριστικό παράδειγμα είναι οι βιβλιοθήκες διαφημίσεων [1] [2] οι οποίες δημιουργούν ποικίλα προβλήματα ασφάλειας. Σε αυτό το σημείο πρέπει να τονιστεί η κατακόρυφη αύξηση των freemium εφαρμογών οπότε και η αύξηση των ενδεχόμενων προβλημάτων.

Άλλος ένας λόγος ανησυχίας, όσον αφορά την ασφάλεια, είναι η δημιουργία καταστημάτων εφαρμογών πλην του επίσημου (3rd party app stores) ή η εγκατάσταση των εφαρμογών μόνο με την χρήση του αρχείου εγκατάστασης (apk). Σε αυτή την περίπτωση εγγυάται ο ιστότοπος ή η υπηρεσία που παρέχει τα apk ότι δεν περιέχεται κακόβουλο λογισμικό. Όμως, σύμφωνα με

δοκιμές [3] που πραγματοποιήθηκαν σε εφαρμογές οι οποίες βρίσκονται σε 3rd party app stores, αυτές κατά 5-13% αποτελούν repackaged εφαρμογές της "γνήσιας" αρχικής. Δηλαδή ακολουθήθηκε η διαδικασία του decompile, έγιναν αλλαγές στον πηγαίο κώδικα και έπειτα έγινε ξανά compile ώστε να διατεθεί μέσω του καταστήματος. Αυτό μπορεί να οδηγήσει στην, ανώδυνη για τον χρήστη, αλλαγή του provider των διαφημίσεων (με στόχο την μεταφορά των κερδών σε κάποιον άλλο από αυτόν που είχε οριστεί αρχικά) μέχρι και στην εμφύτευση backdoor ή malicious payload.

Η παρούσα διπλωματική έχει ως αντικείμενο την μελέτη εφαρμογών android και τον εντοπισμό κάποιων συνηθισμένων αλλά σημαντικών λαθών κατά την ανάπτυξη του λογισμικού. Αποτελεί συνέχεια προηγούμενων δημοσιευμένων εργασιών που είχαν αντικείμενο την ασφάλεια και ιδιωτικότητα των δεδομένων σε mobile dating εφαρμογές [4] [5].

Με την μορφή reports η εφαρμογή που υλοποιήθηκε μπορεί να ειδοποιεί κάποιον ενδιαφερόμενο για πιθανά προβλήματα που υπάρχουν στον κώδικα της εφαρμογής. Για να δημιουργήσουμε τα reports της κάθε προς έλεγχο εφαρμογής έπρεπε να ακολουθήσουμε την αντίστροφη πορεία σε σύγκριση με την ανάπτυξη λογισμικού. Τα αρχεία που είχαμε στην διάθεσή μας ήταν τα αρχεία εγκατάστασης (apk) των εφαρμογών. Αυτά, είτε ήταν από διάφορους mirrors είτε τα κάναμε extract από το κινητό μας στο οποίο είχαν εγκατασταθεί μέσω του επίσημου αποθετηρίου (Google Play Store). Με την χρήση του dex2jar και του jd (Java Decompiler) αποκτήσαμε ό,τι πιο κοντινό γινόταν στον πηγαίο κώδικα της εφαρμογής.

Μία εναλλακτική λύση [6] [7] η οποία χρησιμοποιείται για την ανάλυση των λειτουργιών μίας εφαρμογής είναι η καταγραφή, εκτός των αδειών που είναι δηλωμένες στο AndroidManifest, των κλήσεων API που γίνονται και των αδειών που χρειάζονται για να εκτελεστούν, των πληροφοριών (ID's, location data, WiFi data, κτλ) που μπορεί να αποστέλλονται μέσω διαύλων επικοινωνίας. Οι άδειες στο manifest δεν είναι αρκετές για να αναλυθεί η συμπεριφορά μίας εφαρμογής καθώς δεν ξέρει ο χρήστης πως αξιοποιείται το κάθε resource. Για παράδειγμα μία εφαρμογή χαρτών ζητάει πρόσβαση στο internet και στο GPS μέσω του AndroidManifest. Η έγκριση του χρήστη ώστε να δοθεί η δυνατότητα στην εφαρμογή να έχει πρόσβαση στα κατάλληλα API δεν σημαίνει ότι γνωρίζει αν π.χ το στίγμα το οποίο χρησιμοποιεί η εφαρμογή αποστέλλεται μέσω των αντίστοιχων βιβλιοθηκών σε εταιρίες με στόχο τις στοχευμένες διαφημίσεις.

1.1 Στόχοι της εργασίας

Ο στόχος της εργασίας είναι να δημιουργηθεί λογισμικό το οποίο θα μπορεί να κάνει τους παρακάτω ελέγχους αυτοματοποιημένα, ώστε να μειώνεται ο χρόνος που ο κάθε ερευνητής πρέπει να δαπανήσει για να ελέγξει την εκάστοτε εφαρμογή.

Οι στόχοι της εργασίας περιλαμβάνουν τα ακόλουθα:

1. Έλεγχος για την ύπαρξη RawQuery.
2. Έλεγχος για αποθήκευση σε SharedPreferences.
3. Αποθήκευση σε αρχεία.
4. Χρήση του DeviceID.

5. Χρήση αδύναμων αλγορίθμων κρυπτογράφησης ή και συνδυασμών αυτών (AES/CBC/NoPadding, AES/ECB, MD5, DES).
6. Εσφαλμένη χρήση secure random.
7. Λάθος flags ως προς την πρόσβαση σε αρχεία.
8. Προσθήκη πιστοποιητικών πέραν του συστήματος.

1.2 Βασικοί ορισμοί

Υπάρχουν διάφοροι ορισμοί οι οποίοι καλό είναι να αναλυθούν πριν γίνει εκτενέστερη παρουσίαση των επί μέρους στοιχείων της διπλωματικής.

1.2.1 Decompile

Το decompile των εφαρμογών γίνεται σε δύο στάδια. Στο τέλος έχουμε έναν φάκελο με όλα τα αρχεία java της εφαρμογής. Ουσιαστικά έχουμε το λειτουργικό κομμάτι της εφαρμογής και είναι αυτό το οποίο ελέγχουμε για πιθανά σφάλματα.

1.2.1.1 Dex2jar

Το dex2jar είναι το πρώτο εργαλείο το οποίο χρησιμοποιούμε ώστε να παράξουμε, από το apk της εφαρμογής, το jar αρχείο.

1.2.1.2 Apktool

Το apktool χρησιμοποιείται για να παράξουμε το AndroidManifest. Το αρχείο αυτό περιέχει κρίσιμες πληροφορίες όπως τα δικαιώματα της εφαρμογής.

1.2.1.3 Jd-cmd

Το jd-cmd είναι το commandline εργαλείο του java decompiler το οποίο μετατρέπει το jar αρχείο σε Java Sources.

1.3 Παραδοτέα της εργασίας

1. Το έντυπο κείμενο της πτυχιακής εργασίας.
2. Το λογισμικό το οποίο αναπτύχθηκε.
3. Η συλλογή πηγών και σχετικής βιβλιογραφίας για τη δημιουργία μίας βάσης γνώσης σχετικά με το θέμα.

1.4 Δομή της εργασίας

Τα επόμενα κεφάλαια θα μπουν σε περισσότερες τεχνικές λεπτομέρειες όσον αφορά την αρχιτεκτονική του android, τις βέλτιστες πρακτικές προγραμματισμού, την ανάλυση η οποία έγινε από το πρόγραμμά μας και τέλος τα αποτελέσματα τα οποία εξήχθησαν για την κάθε εφαρμογή.

Κεφάλαιο 2ο

2 Αρχιτεκτονική ασφαλείας στο Android

Τα κομμάτια της αρχιτεκτονικής ασφαλείας του Android τα οποία άπτονται των ελέγχων που κάναμε αφορούν αποκλειστικά την ασφάλεια του λειτουργικού συστήματος και του kernel. Το Android βασίζεται στον linux kernel οπότε θα μπορούσαμε να πούμε ότι αυτή η πλατφόρμα μας παρέχει τις λειτουργίες του πυρήνα του linux καθώς επίσης και έναν ασφαλή δίαυλο επικοινωνίας για να συνομιλούν οι εφαρμογές μεταξύ τους. Θα μπορούσαμε, για να αναλύσουμε ευκολότερα κάποιους τομείς, να χωρίσουμε την ασφάλεια στις εξής κατηγορίες: linux security, application sandbox, System partition security, filesystem permissions, cryptography, rooting of devices and filesystem encryption.

Εικόνα 1 Το framework του Android

2.1 Linux security

Κάποια από τα χαρακτηριστικά ασφαλείας που δανείζει ο linux kernel στο Android είναι τα εξής:

- User-based μοντέλο αδειών.
- Διαχωρισμός των διεργασιών.
- Επεκτάσιμος μηχανισμός για ασφαλές IPC.
- Δυνατότητα αφαίρεσης άχρηστων ή εν δυνάμει μη ασφαλών μερών του πυρήνα.

Το Android ως ένα λειτουργικό σύστημα πολλαπλών χρηστών, θα πρέπει να έχει την δυνατότητα να διαχωρίζει τα resources του εκάστοτε χρήστη ακριβώς όπως συμβαίνει και στο linux. Έτσι επιτυγχάνεται:

- Η απαγόρευση στον χρήστη Α να διαβάσει τα αρχεία του χρήστη Β.
- Ο χρήστης Α να μην επεμβαίνει στην μνήμη του χρήστη Β.
- Ο χρήστης Α να μην επεμβαίνει στην χρήση των CPU resources του χρήστη Β.
- Ο χρήστης Α να μην επεμβαίνει στις συσκευές (τηλέφωνο, GPS, Bluetooth) του χρήστη Β.

2.2 Application Sandbox

Όπως αναφέραμε και πριν το android, που βασίζεται στον πυρήνα του linux, αναγνωρίζει και διαχωρίζει τα resources μεταξύ των εφαρμογών. Το λειτουργικό σύστημα δίνει ένα μοναδικό user ID (UID) σε κάθε android εφαρμογή και την εκτελεί (με εκείνο το UID) σε ξεχωριστό process. Αυτή η προσέγγιση είναι διαφορετική από άλλα λειτουργικά συστήματα, ακόμη και από linux όπου πολλαπλές εφαρμογές τρέχουν με τις ίδιες άδειες χρήσης. Αυτό αποτελεί ένα sandbox εφαρμογών σε επίπεδο kernel. Ο πυρήνας επιβάλει την ασφάλεια μεταξύ των εφαρμογών και του συστήματος στο επίπεδο διεργασιών. Αφού το sandbox γίνεται σε επίπεδο kernel, το μοντέλο ασφάλειας επεκτείνεται στον native κώδικα και στις εφαρμογές του ίδιου του λειτουργικού συστήματος.

2.3 System Partition Security

Το partition του συστήματος περιέχει τον πυρήνα του android, βιβλιοθήκες του λειτουργικού, το runtime των εφαρμογών, το framework των εφαρμογών και τις εφαρμογές. Αυτό το partition είναι read-only.

2.4 Filesystem Permissions

Σε ένα UNIX-like περιβάλλον, οι άδειες του filesystem διασφαλίζουν ότι ο ένας χρήστης δεν μπορεί να πειράξει ή να διαβάσει τα αρχεία ενός άλλου. Στη περίπτωση του android, κάθε εφαρμογή τρέχει ως ξεχωριστός χρήστης. Τα αρχεία που δημιουργούνται δεν μπορούν να διαβαστούν ή να μετατραπούν, εκτός κι αν ο προγραμματιστής επιλέξει διαφορετικά.

2.5 Cryptography

Το android προσφέρει ένα σύνολο κρυπτογραφικών API για να χρησιμοποιούνται από τις εφαρμογές. Σε αυτά συμπεριλαμβάνονται υλοποιήσεις γνωστών standard και συχνά χρησιμοποιούμενων αλγορίθμων όπως οι AES, RSA, DSA, και SHA. Επιπλέον, προσφέρονται API για πρωτόκολλα υψηλότερου επιπέδου όπως το SSL και το HTTPS.

Από το Android 4.0 και μετά υπάρχει το KeyChain το οποίο επιτρέπει στις εφαρμογές να αποθηκεύουν ιδιωτικά κλειδιά και certificate chains στον χώρο του συστήματος για την φύλαξη των διαπιστευτηρίων.

2.6 Rooting of Devices

Εξ αρχής στο android μόνο ο πυρήνας και ένα μικρό μέρος των core εφαρμογών τρέχουν με root δικαιώματα. Γενικά ο root χρήστης έχει πλήρη πρόσβαση σε όλες τις εφαρμογές και τα δεδομένα τους. Οι χρήστες οι οποίοι αλλάζουν τις άδειες στην android συσκευή τους για να αποκτήσουν root πρόσβαση οι εφαρμογές τους, αντιμετωπίζουν αυξημένο ρίσκο ασφαλείας καθώς κακόβουλες εφαρμογές μπορεί να προσπαθήσουν να εκμεταλλευτούν ατέλειες εφαρμογών ή και του συστήματος.

Όσον αφορά στην κρυπτογράφηση των δεδομένων, με κλειδί που αποθηκεύεται στην συσκευή, δεν υπάρχει πλέον προστασία σε μία root συσκευή. Αυτό συμβαίνει επειδή ο root χρήστης έχει πρόσβαση σε όλο το σύστημα και τα αρχεία όπως προαναφέραμε. Στην περίπτωση της κρυπτογράφησης με κλειδί εκτός της συσκευής (π.χ σε έναν server ή με χρήση user password) υπάρχει μία σχετική ασφάλεια, μέχρις ότου χρειαστεί να δοθεί το password ή γενικότερα το κλειδί με οποιονδήποτε τρόπο. Σε αυτή την περίπτωση ο root χρήστης πάλι θα είναι σε θέση να αποκτήσει πρόσβαση στο κλειδί.

Στην περίπτωση της απώλειας της συσκευής, η πλήρης κρυπτογράφηση του filesystem στην συσκευή χρησιμοποιεί το password του χρήστη για να προστατέψει το κλειδί κρυπτογράφησης. Έτσι, ακόμη και με αλλοίωση ή αλλαγή του bootloader ή του λειτουργικού συστήματος δεν είναι δυνατή η ανάκτηση των δεδομένων του χρήστη χωρίς το password του.

2.7 Filesystem Encryption

Από το android 3 και μετά προσφέρεται πλήρης κρυπτογράφηση του filesystem οπότε όλα τα δεδομένα του χρήστη κρυπτογραφούνται με την χρήση AES128 με CBC και ESSIV:SHA256. Το κλειδί κρυπτογράφησης προστατεύεται από τον AES128 χρησιμοποιώντας ένα κλειδί το οποίο προέρχεται από το password του χρήστη, διασφαλίζοντας έτσι την πρόσβαση στα αποθηκευμένα δεδομένα χωρίς το password του χρήστη. Για να αποφευχθούν οι επιθέσεις με χρήση rainbow tables ή brute force, το password συνδυάζεται με ένα τυχαίο salt και γίνεται hash επανειλημμένα με SHA1 σε συνδυασμό με τον PBKDF2 αλγόριθμο πριν χρησιμοποιηθεί για να γίνει αποκρυπτογράφηση του filesystem κλειδιού. Για να αποφευχθούν επιθέσεις με την χρήση λεξικού το android θέτει κανόνες πολυπλοκότητας των password. Οι κανόνες αυτοί ορίζονται από τον admin της συσκευής και επιβάλλονται από το λειτουργικό σύστημα.

Κεφάλαιο 3ο

3 Βέλτιστες πρακτικές προγραμματισμού

Για την αρτιότερη υλοποίηση εφαρμογών η εταιρία που αναπτύσσει το λειτουργικό σύστημα, τα διάφορα API και το SDK έχει τεκμηριώσει τις πιο βασικές βέλτιστες πρακτικές όσον αφορά λάθη στον προγραμματισμό τα οποία μπορούν να οδηγήσουν σε πρόβλημα εμπιστευτικότητας δεδομένων. Οι πρακτικές αυτές δεν αποτελούν απαραίτητη προϋπόθεση για την αποδοχή της εφαρμογής στο κατάστημα εφαρμογών αλλά αποτελούν απλά συμβουλές για την ασφάλειά της. Οι βέλτιστες πρακτικές που επιλέξαμε για να ελέγχουμε τις εφαρμογές μπορούν να χωριστούν σε δύο γενικές κατηγορίες. Η πρώτη έχει να κάνει με τον τρόπο αποθήκευσης των δεδομένων της εφαρμογής μας και η δεύτερη με την κρυπτογράφηση των δεδομένων. Υπάρχουν και μερικοί άλλοι έλεγχοι που δεν έχουν συνάφεια με τις παραπάνω κατηγορίες. Θα εξετάσουμε μία μία τις πρακτικές αυτές παρακάτω.

3.1 Αποθήκευση δεδομένων στις android εφαρμογές

Μία εφαρμογή android μπορεί να αποθηκεύσει δεδομένα με τρεις τρόπους. Ο πρώτος είναι η αποθήκευση στην SQLite που είναι ενσωματωμένη στο android. Ο δεύτερος τρόπος είναι τα sharedpreferences τα οποία είναι αρχεία που αποθηκεύονται στο internal storage και τα δεδομένα μέσα σε αυτά βρίσκονται σε μορφή ζευγών (όνομα-τιμή) και τέλος σε αρχεία τα οποία μπορούν να αποθηκευτούν είτε στην internal μνήμη είτε στην κάρτα sd. Πρόβλημα εμπιστευτικότητας μπορεί να υπάρξει σε οποιονδήποτε από τους τρεις παραπάνω τρόπους.

3.1.1 RawQuery

Τα rawQuery είναι query της SQLite που υπάρχει στο android. Αποτελούν query που δεν χρησιμοποιούν prepared statements αλλά το query είναι σε plain text. Για να γίνει εισαγωγή δεδομένων από μεταβλητές γίνεται concatenate στο sql string και ένωση με την τιμή όποιας μεταβλητής θέλουμε.

Για να αποφύγουμε τα προβλήματα με τα rawquery (το πιο σύνηθες θα μπορούσαμε να πούμε ότι είναι τα SQL Injections) προτείνεται να χρησιμοποιήσουμε prepared statements.

3.1.2 SharedPreferences

Τα sharedpreferences [8] είναι αρχεία τα οποία αποθηκεύονται στην internal μνήμη στην οποία όπως προαναφέραμε θεωρητικά έχει πρόσβαση μόνο η εκάστοτε εφαρμογή. Τα αρχεία αυτά είναι xml και τα δεδομένα μέσα σε αυτά είναι αποθηκευμένα σε μορφή ζευγών (όνομα-τιμή). Παρόλη την προστασία των δεδομένων της εφαρμογής από τρίτους, που θα προσπαθήσουν να διαβάσουν τα αρχεία, υπάρχουν τρόποι ανάκτησης αυτών των αρχείων. Σε περίπτωση που ο προγραμματιστής έχει επιλέξει την αποθήκευση ευαίσθητων δεδομένων όπως π.χ. id, κλειδιά κρυπτογράφησης, gps στίγμα, email, αριθμός τηλεφώνου κτλ, αυτά τα δεδομένα θα μπορούσαν να καταστούν ευάλωτα σε περίπτωση που αλλοιωθεί το λειτουργικό από τον χρήστη. Η αλλοίωση αυτή μπορεί να είναι είτε εσκεμμένη είτε όχι. Το κύριο παράδειγμα εσκεμμένης

αλλοίωσης είναι το "rootάρισμα" του κινητού, η αλλαγή του kernel κτλ ενώ το πιο συχνό αίτιο μη εσκεμμένης αλλοίωσης είναι όταν κάποιος ιός εκμεταλλεύεται ένα κενό ασφαλείας και αποκτά με αυτό τον τρόπο root δικαιώματα. Με οποιονδήποτε από αυτούς τους τρόπους είναι δυνατό να ανακτηθούν από τον κακόβουλο χρήστη τα δεδομένα τα οποία είναι αποθηκευμένα στα sharedpreferences και στην περίπτωση που αυτά είναι ευαίσθητα τότε το κόστος είναι μεγαλύτερο.

Πρόταση σε αυτή την περίπτωση είναι να μην αποθηκεύουμε ευαίσθητα δεδομένα ή κατά το δυνατόν τα λιγότερα.

3.1.3 Files

Τέλος, η αποθήκευση δεδομένων μπορεί να γίνει και σε απλά αρχεία [8]. Αυτά τα αρχεία μπορούν να αποθηκευτούν είτε στην internal μνήμη είτε στην κάρτα SD. Σημειώνουμε ότι η δεύτερη είναι προσβάσιμη από όλους τους χρήστες και τα προγράμματα. Στην περίπτωση αυτή, και εφόσον έχουν αποθηκευτεί ευαίσθητα δεδομένα σε αυτά τα αρχεία, οποιοσδήποτε μπορεί να προσπελάσει το περιεχόμενό τους αρκεί να έχει το κατάλληλο δικαίωμα στο AndroidManifest. Βέβαια, και στην εσωτερική μνήμη να είναι αποθηκευμένα, υπάρχει η δυνατότητα όπως εξηγήσαμε και παραπάνω να γίνει προσπέλαση των αρχείων.

Για να αποφύγουμε προβλήματα εμπιστευτικότητας καλό είναι να μην αποθηκεύονται ευαίσθητα δεδομένα στα αρχεία και ειδικότερα αν βρίσκονται στην κάρτα SD.

3.2 Κρυπτογράφηση στις android εφαρμογές

Για την ασφαλέστερη αποθήκευση δεδομένων, με όποιον από τους τρεις τρόπους και αν επιλέξουμε, προτείνεται να γίνεται κρυπτογράφηση των ευαίσθητων δεδομένων [9] και αυτό γιατί στην εσωτερική μνήμη, στον φάκελο της εφαρμογής, υπάρχει και το αρχείο της βάσης δεδομένων. Αυτό θα μπορούσε να διαβαστεί από όποιον καταφέρει να έχει πρόσβαση στην εσωτερική μνήμη. Συνήθης πρακτική είναι να γίνεται hash κάποια τιμή της οποία δεν μας ενδιαφέρει να ξέρουμε το αρχικό περιεχόμενο. Χαρακτηριστικό παράδειγμα είναι το password. Όχι μόνο δεν χρειάζεται να έχουμε αποθηκευμένη την αρχική τιμή αλλά αντενδείκνυται. Αν υποθέσουμε ότι μιλάμε για μία αποκλειστικά offline εφαρμογή (άρα χρειάζεται password αποθηκευμένο στην βάση δεδομένων), κάνοντας hash με κάποιον ισχυρό αλγόριθμο το password του χρήστη κατά την εγγραφή του χρήστη, θα αποθηκεύσουμε στην βάση αυτή την hashed τιμή. Κάθε φορά που ο χρήστης θα εισάγει το password για να κάνει την αυθεντικοποίηση τότε θα γίνεται hash αυτό το password και αν συμπίπτει με το hash στην βάση τότε θα επιτρέπεται η είσοδος.

Τα hashes όμως δεν βολεύουν στην περίπτωση που θέλουμε να κρυπτογραφήσουμε κάτι το οποίο θα πρέπει να υπάρχει η δυνατότητα να διαβαστεί στην αρχική του τιμή πριν την κρυπτογράφηση.

Και στην περίπτωση των hash και της κρυπτογράφησης θα πρέπει να επιλεχθούν συνδυασμοί αλγορίθμων που δεν θα καθιστούν ευάλωτη αυτή την κρυπτογράφηση [10].

3.2.1 Αλγόριθμοι κρυπτογράφησης

Στην μοντέρνα κρυπτογραφία ασφαλής θεωρείται ένας αλγόριθμος που είναι ακριβός υπολογιστικά για να σπάσει. Αλγόριθμοι κλειστού λογισμικού θα πρέπει να αποφεύγονται καθώς στηρίζονται στην ασφάλεια, λόγω της έλλειψης γνώσης του τρόπου λειτουργίας του και όχι στην ποιότητα τους.

Παρακάτω αναλύονται κάποιοι συνδυασμοί οι οποίοι είναι ευάλωτοι [11] [12] καθώς είναι σχετικά εύκολο να βρεθεί η αρχική τιμή του κρυπτογραφήματος.

3.2.1.1 MD5

Ο MD5 hash αλγόριθμος αποδείχθηκε ότι ποτέ δεν ήταν τόσο ασφαλής από όσο πιστευόταν. Παρόλο που θεωρείται ασφαλής για εφαρμογές που για παράδειγμα υπολογίζουν τις hash τιμές για τα binaries τα οποία γίνονται διαθέσιμα στο internet, οι εφαρμογές που διαχειρίζονται ευαίσθητα δεδομένα θα πρέπει να αρχίσουν να αποφεύγουν την χρήση του και να χρησιμοποιούν άλλους ασφαλέστερους.

3.2.1.2 SHA-0

Ο SHA-0 έχει επανειλημμένα αποδειχθεί ότι δεν είναι ασφαλής πλέον για χρήση σε εφαρμογές που περιέχουν ευαίσθητα δεδομένα.

3.2.1.3 SHA-1

Ο SHA-1 έχει χάσει μέρος της “δύναμής” του και πλέον δεν προτείνεται η χρήση του.

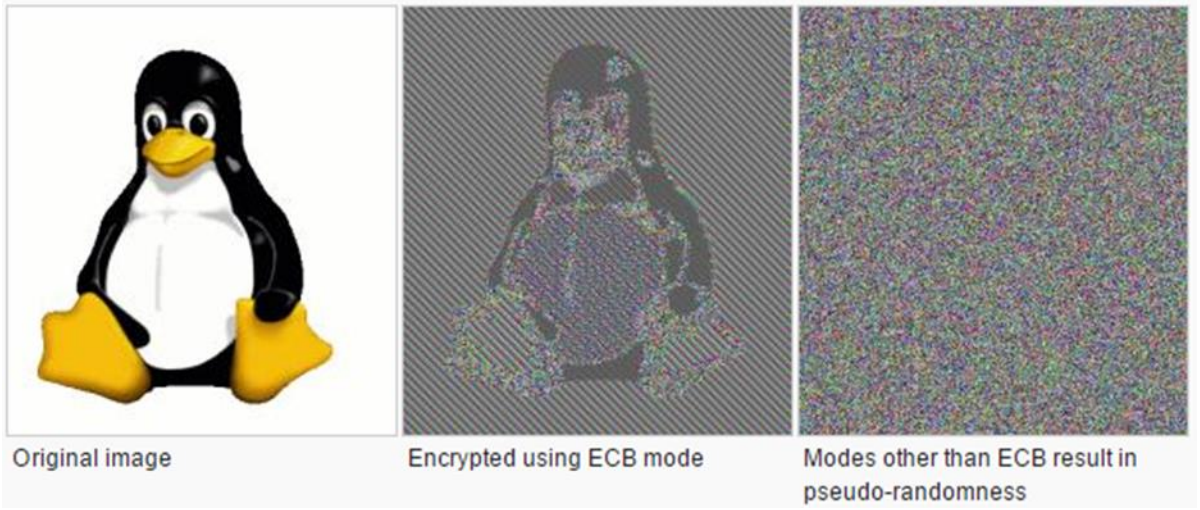
Για την αντικατάσταση των προηγούμενων τριών αλγορίθμων (MD5, SHA-0, SHA-1) προτείνεται η χρήση του SHA-256.

3.2.1.4 DES

Ο DES θεωρείται ανασφαλής για τα σημερινά δεδομένα και δεν θα πρέπει να προτείνεται η χρήση του παρόλο που εθεωρείτο για πολλά χρόνια το στάνταρ. Πλέον, ακόμη και ένα desktop μηχάνημα μπορεί να σπάσει το κρυπτογράφημα. Αντ’αυτού προτείνεται η χρήση του επίσης συμμετρικού αλγορίθμου AES.

3.2.1.5 AES/ECB

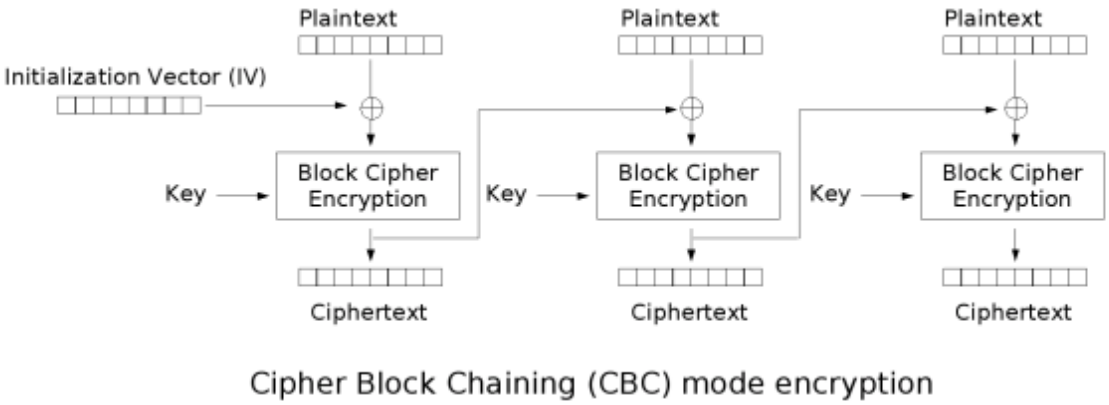
Ο συνδυασμός AES/ECB θεωρείται ευάλωτος και θα πρέπει να αποφεύγεται η χρήση του. Αυτό συμβαίνει διότι ο αλγόριθμος αυτός κρυπτογραφεί το κάθε μπλοκ με τον ίδιο τρόπο. Για παράδειγμα ας υποθέσουμε ότι κρυπτογραφούμε τους κωδικούς για κάθε site. Στην περίπτωση που δύο κωδικοί είναι ακριβώς ίδιοι το αποτέλεσμα μετά την κρυπτογράφηση είναι κι αυτό ίδιο. Οπότε μπορεί κάποιος να καταλάβει σε ποιες υπηρεσίες/sites ο χρήστης έχει τον ίδιο κωδικό. Γι’αυτόν τον λόγο ο ECB αλγόριθμος είναι επιρρεπής στην ανάλυση και στις επαναληπτικές επιθέσεις.



Εικόνα 2 Γραφική απεικόνιση της κρυπτογράφησης ECB

3.2.1.6 AES/CBC

Το CBC mode σε αντίθεση με το ECB διασφαλίζει ότι δύο μπλοκ ακόμη κα αν το plaintext είναι το ίδιο και στα δύο δεν θα βγάλουν το ίδιο κρυπτογράφημα. Αυτό γίνεται με την προσθήκη (με την χρήση XOR) στο plaintext του κρυπτογραφήματος του προηγούμενου γύρου. Η χρήση του συνδυασμού αυτού προτείνεται καθώς είναι ισχυρός. Θα πρέπει όμως να αποφεύγεται η χρήση του χωρίς padding.



Εικόνα 3 Ο τρόπος λειτουργίας του Cipher Block Chaining (CBC)

3.2.2 Padding

Ένας block cipher [13] λειτουργεί σε δεδομένα που βρίσκονται σε σταθερού μεγέθους block. Κρυπτογραφεί το κάθε μπλοκ και περνά στο επόμενο μέχρι να φτάσει και στο τελευταίο. Επίσης, με τον ίδιο τρόπο γίνεται και η αποκρυπτογράφηση. Ξεκινάει από το πρώτο μπλοκ και πηγαίνει στο επόμενο μέχρι να αποκρυπτογραφήσει και το τελευταίο.

Σε αυτό το σημείο δημιουργούνται δύο μεγάλα ερωτήματα:

- Τί συμβαίνει εάν το μήκος των δεδομένων δεν είναι πολλαπλάσιο του μεγέθους του μπλοκ;
- Τί συμβαίνει όταν παραπάνω από ένα μπλοκ είναι ίδια και κρυπτογραφούνται με ακριβώς το ίδιο αποτέλεσμα;

Για να λύσουμε το πρώτο “πρόβλημα” χρησιμοποιούμε padding. Κάθε γκρουπ κρυπτογραφικών στάνταρ χρησιμοποιεί διαφορετικό μοτίβο padding. Ο πιο απλός τρόπος να δημιουργήσει κάποιος σωστό padding είναι να προσθέσει 0 ή 1 στο τέλος κάθε μπλοκ ώστε το μήκος των δεδομένων να φτάσει να γίνει πολλαπλάσιο του μεγέθους του μπλοκ δεδομένων.

Οι περισσότεροι block ciphers χρησιμοποιούν το PKCS7 για padding. Το PKCS7 ορίζει ότι στο τέλος του τελευταίου μπλοκ δεδομένων προστίθεται ο αριθμός των byte που χρειάζονται για να συμπληρωθεί το απαραίτητο μήκος δεδομένων. Έτσι, αν το μέγεθος του μπλοκ είναι 8 byte και το string είναι το "ABC", τότε θα πρέπει να συμπληρωθεί με padding \x05 ώστε το μπλοκ να φτάσει να έχει τιμή "ABC\x05\x05\x05\x05\x05" (Το ABC έχει μέγεθος 3 byte. Άρα 8-3= 5 byte. Γι’ αυτό γεμίζει με την τιμή x05 και την τοποθετεί πέντε φορές στο μπλοκ. Αν το string ήταν "ABCDEFG", με το padding θα έπρεπε να γίνει "ABCDEFG\x01".

Επιπρόσθετα αν το μπλοκ του string είναι πολλαπλάσιο του μεγέθους του μπλοκ τότε θα πρέπει να προστεθει ένα άδειο μπλοκ επιπλέον. Αυτό μπορεί να ακούγεται περίεργο αλλά είναι απαραίτητο ώστε να μην υπάρξει πρόβλημα στην όλη διαδικασία. Για να γίνει κατανοητό ας υποθέσουμε ότι έχει το παρακάτω string "ABCDEFG\x01". Το \x01 φαίνεται σαν να είναι padding αλλά δεν είναι τίποτα άλλο παρά η τιμή “1” και όχι padding. Γι αυτό το λόγο αν έχουμε ένα string του οποίου το μήκος είναι ακριβές πολλαπλάσιο του μεγέθους του μπλοκ προσθέτουμε άλλο ένα μπλοκ με padding. Για παράδειγμα ας υποθέσουμε ότι έχουμε το string "ABCDEFG\x01". Για να μην υπάρξει σύγχυση με το \x01 το οποίο είναι μέρος της τιμής και όχι padding προσθέτουμε μετά από αυτό 8byte για padding. Έτσι το “ABCDEFG\x01" γίνεται "ABCDEFGH\x08\x08\x08\x08\x08\x08\x08\x08".

3.2.1 Λάθος IV

Κατά την χρησιμοποίηση του CBC mode ενός αλγορίθμου κρυπτογράφησης χρειάζεται ένα τυχαίο IV. Το IV πρέπει να είναι ένα τυχαία αρχικοποιημένο string (όπως είδαμε και στο AES/CBC) το οποίο χρησιμοποιείται για να λύσουμε το δεύτερο πρόβλημα που αναφέραμε προηγουμένως. Το IV χρησιμοποιείται ώστε να αλλάζει το προς κρυπτογράφηση plaintext. Έπειτα αυτό γίνεται κρυπτογράφηση με το κλειδί. Σε περίπτωση που έχουμε πολλαπλά μπλοκ τότε το κάθε ένα παίρνει για IV κομμάτι του cyphertext του προηγούμενου μπλοκ εκτός από το πρώτο που χρησιμοποιεί το IV.

Όταν χρησιμοποιείται αλγόριθμος που απαιτεί IV υπάρχει το πρόβλημα της αποθήκευσης του IV ώστε να βρίσκεται διαθέσιμο για την αποκρυπτογράφηση. Αυτό αποτελεί πρόβλημα καθώς τις περισσότερες φορές χρησιμοποιείται ένα fixed. Επίσης, το android έχει παράξενη συμπεριφορά όταν κάποιος δεν προσδιορίσει IV.

Στην kitkat, το android δημιουργεί ένα φαινομενικά τυχαίο IV κατά την διάρκεια της κρυπτογράφησης, το οποίο είναι διαφορετικό σε κάθε εκτέλεση οπότε δεν θα αποκρυπτοφραφήσει με τον ίδιο τρόπο δεύτερη φορά.

Στην 4.3 η εφαρμογή χρησιμοποιούσε IV γεμάτο μηδενικά τόσο στην κρυπτογράφηση όσο και στην αποκρυπτογράφηση το οποίο οδηγούσε στον ίδιο τρόπο encrypt-decrypt.

3.2.1 Seeded Secure Random και SetSeed()

Η χρησιμοποίηση του seeded constructor ή του setSeed(byte[]) [14] μπορεί να αντικαταστήσει το κρυπτογραφικά ισχυρό seed που προηγουμένως είχε δημιουργηθεί με μία προβλέψιμη σειρά αριθμών που δεν είναι κατάλληλη για ασφαλή χρήση. Καλό είναι να αποφεύγεται η χρήση του seed.

3.3 Χρήση του DeviceId για αναγνωριστικό εγκατάστασης μίας εφαρμογής

Είναι σύνηθες, και δικαιολογημένο μέχρι πρόσφατα, ένας developer θέλοντας να παρακολουθεί τις μοναδικές εγκαταστάσεις της εφαρμογής του να καλεί το TelephonyManager.getDeviceId() ώστε να λάβει το id της εκάστοτε συσκευής και να μπορεί να ταυτοποιεί την κάθε συσκευή. Υπάρχουν όμως δύο προβλήματα [15]. Καταρχάς δεν δουλεύει πάντα διότι δεν είναι όλες οι συσκευές ίδιες και δεύτερον ακόμη και αν δουλέψει, η τιμή αυτή “επιζεί” ακόμη και μετά από factory reset, οπότε ακόμη και αν ο κάτοχος του κινητού έχει αλλάξει, τα δεδομένα του developer δεν θα αλλάζουν.

Αυτό που προτείνεται, για να γίνεται καταγραφή των εγκαταστάσεων, είναι η χρήση του UUID ως αναγνωριστικού.

3.4 Προσθήκη πιστοποιητικών με την χρήση του Certificate Manager

Από την ίδια την πλατφόρμα του λογισμικού δίνεται η δυνατότητα εφόσον κάποιος το θελήσει να εισάγει ένα δικό του πιστοποιητικό και να είναι αποδεκτό το λειτουργικό. Αυτό εγκυμονεί κινδύνους καθώς ο κάθε developer μπορεί να εισάγει οποιοδήποτε πιστοποιητικό θέλει το οποίο ενδεχομένως να είναι προβληματικό αν όχι εξαρχής κακόβουλο.

Κεφάλαιο 4ο

4 Η εφαρμογή μας και η ανάλυση που πραγματοποιεί

Οι έλεγχοι οι οποίοι υλοποιήθηκαν βασίζονται πάνω στις βέλτιστες πρακτικές που αναφέρθηκαν προηγουμένως. Ελέγχουμε για πρακτικές οι οποίες είναι ακριβώς αντίθετες με αυτές που προτείνει η εταιρία που υποστηρίζει το λειτουργικό σύστημα. Σε περίπτωση που βρεθούν αποτελέσματα που συνάδουν με τα κριτήρια της αναζήτησης τότε βγαίνει το αντίστοιχο warning μαζί με το όνομα του αρχείου στο οποίο βρέθηκε και τη γραμμή όπου εντοπίστηκε.

4.1 Η δομή της εφαρμογής

Το project περιέχει τους φακέλους apk, jar, Sources και Tools. Στον φάκελο apk τοποθετούνται τα apk των εφαρμογών που θέλουμε να εξετάσουμε. Στον φάκελο jar αποθηκεύονται αυτόματα τα jar αρχεία των apk και στον Sources ο πηγαίος κώδικας της εκάστοτε εφαρμογής. Εάν τοποθετήσουμε παραπάνω από ένα apk αρχείο τότε σειριακά θα γίνει το decompile και έπειτα η ανάλυση σε όλα. Στον φάκελο Tools περιέχονται όλα τα εργαλεία τα οποία θα χρησιμοποιηθούν από τα script μας.

Εφόσον κάποιος έχει αντιγράψει τα apk αρχεία που θέλει στον κατάλληλο φάκελο, τρέχει το script(decser.py) που βρίσκεται στον root φάκελο του project μας. Θα αρχίσει το decompile μίας μίας των εφαρμογών και όταν ολοκληρωθεί η διαδικασία για όλες τότε θα ξεκινήσει το δεύτερο script (search.py) το οποίο θα κάνει τον έλεγχο στον κώδικα των εφαρμογών που βρίσκονται στον φάκελο Sources. Η διάσπαση των δύο script έγινε ώστε να μπορούμε να εκτελέσουμε το search χωρίς να χρειάζεται κάθε φορά να εκτελούμε και την διαδικασία του decompile.

Για το decompile χρησιμοποιήθηκαν τα apktool για την εξαγωγή του androidmanifest, το dex2jar για την εξαγωγή του jar αρχείου από το apk και τέλος το jd-cmd για να πάρουμε τον κώδικα από το Jar αρχείο. Ειδικά για το κομμάτι του decompiling δοκιμάστηκαν διάφοροι decompilers όπως ο Procyon, ο cfr και ο jd και κατέληξα στον τελευταίο. Ο jd παρουσίασε τους καλύτερους χρόνους και την μικρότερη πιθανότητα να μην καταφέρει να φέρει εις πέρας το έργο του. Παρόλα αυτά, ο Procyon και ο cfr όποτε ολοκλήρωναν το decompile ενδεχομένως να είχαν καλύτερο αποτέλεσμα έναντι του jd.

4.2 Έλεγχος για προβληματικό κώδικα

Για το search χρησιμοποιήθηκε η grep και ως όρισμα για την αναζήτηση λέξεις κλειδιά όπως ο constructor, ονόματα μεθόδων, αλγορίθμων κτλ. Η grep πραγματοποιεί recursive αναζήτηση από τον root φάκελο της κάθε εφαρμογής και αν βρει κάποιο αποτέλεσμα τότε προστίθεται στο

αρχείο αποτελεσμάτων ο κατάλληλος τίτλος που περιγράφει το πρόβλημα και από κάτω τα ευρήματα.

4.2.1 Έλεγχος για την ύπαρξη RawQuery

Για να ελέγξουμε την ύπαρξη RawQuery στον κώδικα των εφαρμογών κάνουμε αναζήτηση έχοντας ως κλειδί το “RawQuery”. Στην περίπτωση που εμφανιστούν αποτελέσματα τότε ειδοποιούμε τον χρήστη να κοιτάξει τα σημεία στα οποία χρησιμοποιεί αυτού του είδους τα query ώστε να επιβεβαιώσει ότι δεν γίνεται εισαγωγή δεδομένων χωρίς έλεγχο. Στην περίπτωση που ο προγραμματιστής δεν έχει δημιουργήσει τις κατάλληλες δικλείδες ασφαλείας υπάρχει κίνδυνος για επίθεση SQL injection. Το μήνυμα στον χρήστη αποτελεί προειδοποίηση διότι στις περισσότερες περιπτώσεις τα RawQuery χρησιμοποιούνται για να δημιουργηθούν λίστες με περιεχόμενα (listviews). Συνήθως τα Listviews χρησιμοποιούνται για να υλοποιηθούν τα μενού ή γενικότερα scrollable λίστες. Στην περίπτωση των λιστών, κατά κανόνα, δεν υπάρχει είσοδος δεδομένων στο query αλλά αυτά είναι fixed μέσα στην εφαρμογή (με αυτό τον τρόπο δεν υπάρχει κίνδυνος SQL Injection).

4.2.2 Έλεγχος για αποθήκευση σε SharedPreferences

Για να εντοπιστούν οι περιοχές του κώδικα στις οποίες υπάρχει χρήση των SharedPreferences χρησιμοποιούμε το string “getSharedPreferences”. Η χρήση τους δεν είναι προβληματική εξ ορισμού αλλά μόνο στην περίπτωση που τα αρχεία αυτά περιέχουν ευαίσθητα δεδομένα. Γι’ αυτό τον λόγο και σε αυτή την περίπτωση ο χρήστης της εφαρμογής δέχεται προειδοποίηση ώστε να ελέγξει τον κώδικά του για τυχόν αποθήκευση δεδομένων που θα μπορούσαν να επιφέρουν πρόβλημα εμπιστευτικότητας.

4.2.3 Αποθήκευση σε αρχεία

Η αναζήτηση για αποθήκευση σε αρχεία στην sd κάρτα γίνεται με κλειδί το “getExternalStoragePublicDirectory”. Αυτή η μέθοδος επιστρέφει τα public directories ώστε να επιλέξει μετέπειτα ο προγραμματιστής που θα κάνει την αποθήκευση των δεδομένων. Και σε αυτή την περίπτωση δίνεται μια προειδοποίηση για έλεγχο των δεδομένων που αποθηκεύονται σε αρχεία.

4.2.4 Χρήση του DeviceID

Για να ελεγχθεί αν μία εφαρμογή χρησιμοποιεί το DeviceID, η αναζήτηση στον κώδικα γίνεται με το “ TelephonyManager.getDeviceId()”. Στην περίπτωση που εντοπιστεί, ο προγραμματιστής θα πρέπει να ελέγξει τον κώδικά του και να αντικαταστήσει το DeviceID με άλλους πιο ασφαλείς τρόπους ταυτοποίησης των χρηστών και του αριθμού εγκαταστάσεων της εφαρμογής του.

4.2.5 Χρήση αδύναμων αλγορίθμων κρυπτογράφησης ή και συνδυασμών αυτών (AES/CBC/NoPadding, AES/ECB, MD5, DES)

Οι έλεγχοι που πραγματοποιούνται είναι για τους παρακάτω αλγορίθμους και τους συνδυασμούς τους: MD5, DES, AES/ECB, AES/CBC/NoPadding.

Για κάθε έναν από αυτούς τους αλγορίθμους ο έλεγχος γίνεται με λέξεις κλειδιά το όνομα του κάθε αλγόριθμου (MD5, DES, AES/ECB, AES/CBC/NoPadding).

4.2.6 Εσφαλμένη χρήση secure random

Σύμφωνα με τις οδηγίες της google [14], η χρήση του seeded constructor ή του setSeed(byte[]) μπορεί να δημιουργήσει πρόβλημα καθώς ενδέχεται να οδηγήσει σε μία προβλεπόμενη σειρά αριθμών. Γι' αυτό προτείνεται η αποφυγή της χρησιμοποίησης του setSeed. Η αναζήτηση για αυτού του τύπου το πρόβλημα γίνεται με τις λέξεις κλειδιά "SecureRandom(.*" και "setSeed".

4.2.7 Χρήση Webview

Η χρήση του webview δεν εντάσεται σε κάποιο σύνολο εσφαλμένων πρακτικών. Το σημείο στο οποίο όμως πρέπει να προσέξει ο προγραμματιστής είναι η javascript. Θα πρέπει ο προγραμματιστής να την απενεργοποιεί για λόγους ασφαλείας καθώς επίσης να ενεργοποιεί τα: setAllowFileAccess & setAllowContentAccess. Η αναζήτηση για το webview γίνεται με κλειδί το "WebView".

4.2.8 Λάθος flags ως προς την πρόσβαση σε αρχεία

Γενικότερα, ότι δεδομένα αποθηκεύονται στην εσωτερική μνήμη της εφαρμογής μας είναι προσβάσιμα μόνο στην ίδια την εφαρμογή. Αυτή η δικλείδα ασφαλείας μπορεί να αλλάξει με την χρησιμοποίηση των MODE_WORLD_READABLE και MODE_WORLD_WRITEABLE [8] [16]. Η αναζήτηση για τα λάθος αυτά flags γίνεται με την χρήση των ονομάτων τους.

4.2.9 Προσθήκη πιστοποιητικών πέραν του συστήματος

Το Android επιτρέπει σε όποιον προγραμματιστή θέλει να προσθέσει πιστοποιητικά ασφαλείας πέραν από αυτά του συστήματος. Με αυτό τον τρόπο υπάρχει πιθανότητα να γίνει εισαγωγή κακόβουλων πιστοποιητικών ή στην καλύτερη περίπτωση πιστοποιητικών που δεν πληρούν τις προδιαγραφές ασφαλείας. Ο έλεγχος για τις προσθήκες τέτοιων πιστοποιητικών γίνεται με τις εξής λέξεις κλειδιά:

- 'KeyStore.getInstance'
- 'SSLContext.getInstance'
- 'getTrustManagers'
- 'getSocketFactory'
- 'HttpsURLConnection'
- 'CertificateFactory.getInstance'
- 'TrustManagerFactory.getDefaultAlgorithm'

Σε αυτή την περίπτωση η αναζήτηση που γίνεται θα πρέπει να εντοπίσει όλα αυτά τα κλειδιά στο ίδιο αρχείο. Μόνο τότε θα υπάρξει προειδοποίηση για την προσθήκη του πιστοποιητικού.

Το σύνολο των παραπάνω αναζητήσεων ουσιαστικά αποτελεί μία στατική ανάλυση του κώδικα. Επειδή η ανάλυση γίνεται με λέξεις κλειδιά είναι πιθανό να υπάρχουν false-positive εγγραφές στο αρχείο αποτελεσμάτων. Γι' αυτό το λόγο να τονίσουμε ότι όλα τα αποτελέσματα αποτελούν προειδοποιήσεις για χειροκίνητο έλεγχο και όχι απαραίτητα ευρήματα.

4.3 Προαπαιτούμενα για την εκτέλεση της εφαρμογής μας

Για να εκτελεστεί επιτυχώς η εφαρμογή που υλοποιήθηκε θα πρέπει ο χρήστης να έχει εγκατεστημένη την Java (χρησιμοποιείται από τους decompilers) και την Python (χρησιμοποιείται από τα script της εφαρμογής). Όσον αφορά την java χρησιμοποιήθηκε η τελευταία έκδοση του openjdk 1.7.0_79 και για την python η 2.7.9.

Κεφάλαιο 5ο

5 Τα αποτελέσματα των ελέγχων

Για την δοκιμή του project κατά την διάρκεια της ανάπτυξης δημιουργήθηκαν κάποιες dummy εφαρμογές οι οποίες είχαν μία από τις ελεγχόμενες λειτουργίες η κάθε μία. Εφόσον ελέγχθηκε ότι κάθε αποτέλεσμα το οποίο εντοπιζόταν καταγραφόταν σωστά στο αρχείο αποτελεσμάτων προχωρήσαμε στον έλεγχο εφαρμογών τρίτων.

Οι εφαρμογές που αποφασίστηκε να ελεγχθούν είναι οι εξής:

- Grindr
- Hornet
- immomo
- pof
- chaton
- singlesaroundme
- skout
- tagged
- tinder
- sayhi
- waplog
- zoosk
- meetme
- lovoo
- i-am
- wechat

Τα apk αρχεία αντιγράφηκαν στον φάκελο apk της εφαρμογής μας και έπειτα κάναμε το decom pile και την ανάλυση.

Τα αποτελέσματα στα οποία καταλήξαμε αναλύονται παρακάτω:

Σε κάθε εφαρμογή που εξετάσαμε θα παρουσιάζεται ένας συγκεντρωτικός πίνακας με τα αποτελέσματά της. Ο παρακάτω πίνακας εξηγεί τι σημαίνει η κάθε χρωματική απόχρωση στους πίνακες αποτελεσμάτων.

	false positive
	not found
	found
	third-party library

Πίνακας 1 Επεξηγηματικός Πίνακας

Όσον αφορά τα RawQuery, SharedPreferences και WebView αυτά εντοπίστηκαν σε όλες τις εφαρμογές. Τις θεωρούμε positive διότι η προειδοποίηση στον χρήστη της εφαρμογής μας είναι γενική ώστε να κοιτάξει ο ίδιος για ενδεχόμενα προβλήματα. Σε κάθε μία από τις παρακάτω αναλύσεις θα σχολιαστούν ευρήματα εκτός των τριών που προαναφέρθηκαν.

Για τα ευρήματα που έχουν καταγραφεί σε 3rd Party βιβλιοθήκες, θα πρέπει να εξεταστεί από τον προγραμματιστή αν επηρεάζουν την ασφάλεια των δεδομένων της εφαρμογής του

5.1 Grindr

Τα αποτελέσματα της ανάλυσης για το Grindr είναι τα εξής:

	Grindr
RawQuery	
SharedPreferences	
FileSave	
DeviceId	
AESCBC	
AESECB	
MD5	
DES	
SecureRandomSeeded	
SecureRandomSetSeed	
WebView	
ModeWorldReadable	
ModeWorldWritable	
CertificateManager	

Πίνακας 2 Αποτελέσματα Grindr

Τα προβλήματα που εντοπίστηκαν αφορούν την γεννήτρια τυχαίων αριθμών στην οποία δόθηκε seed και την χρήση επισφαλών αλγορίθμων κρυπτογράφησης.

5.2 Hornet

Τα αποτελέσματα της ανάλυσης για το Hornet είναι τα εξής:

	Hornet
RawQuery	
SharedPreferences	
FileSave	
DeviceId	
AESCBC	
AESECB	
MD5	
DES	
SecureRandomSeeded	
SecureRandomSetSeed	
WebView	
ModeWorldReadable	
ModeWorldWritable	
CertificateManager	

Πίνακας 3 Αποτελέσματα Hornet

Τα προβλήματα που καταγράφηκαν από την εφαρμογή μας είναι η χρήση μη ασφαλών αλγορίθμων κρυπτογράφησης (σε βιβλιοθήκες τρίτων). Επίσης, εντοπίσθηκε αποθήκευση σε αρχεία αλλά μετά από έλεγχο διαπιστώθηκε ότι πρόκειται για την αποθήκευση φωτογραφιών.

5.3 Immomo

Τα αποτελέσματα της ανάλυσης για το Immomo είναι τα εξής:

	Immomo
RawQuery	
SharedPreferences	
FileSave	
DeviceId	
AESCBC	
AESECB	
MD5	
DES	
SecureRandomSeeded	

SecureRandomSetSeed	
WebView	
ModeWorldReadable	
ModeWorldWritable	
CertificateManager	

Πίνακας 4 Αποτελέσματα Immono

Τα προβλήματα που καταγράφηκαν είναι η χρήση μη ασφαλών αλγορίθμων κρυπτογράφησης. Ιδιαίτερη προσοχή πρέπει να δοθεί στην προσθήκη μέσω του Certificate Manager πιστοποιητικών πέραν αυτών του συστήματος.

5.4 Pof

Τα αποτελέσματα της ανάλυσης για το Pof είναι τα εξής:

	Pof
RawQuery	
SharedPreferences	
FileSave	
DeviceId	
AESCBC	
AESECB	
MD5	
DES	
SecureRandomSeeded	
SecureRandomSetSeed	
WebView	
ModeWorldReadable	
ModeWorldWritable	
CertificateManager	

Πίνακας 5 Αποτελέσματα Pof

Τα προβλήματα που εντοπίστηκαν αφορούν στην χρήση μη ασφαλών αλγορίθμων κρυπτογράφησης και στην χρήση seed στην γεννήτρια τυχαίων αριθμών.

5.5 Chaton

Τα αποτελέσματα της ανάλυσης για το Chaton είναι τα εξής:

	Chaton
RawQuery	
SharedPreferences	
FileSave	
DeviceId	
AESCBC	
AESECB	
MD5	
DES	
SecureRandomSeeded	
SecureRandomSetSeed	
WebView	
ModeWorldReadable	
ModeWorldWritable	
CertificateManager	

Πίνακας 6 Αποτελέσματα Chaton

Τα προβλήματα που εντοπίστηκαν αφορούν στην χρήση του DeviceID για αναγνωριστικό της συσκευής, χρήση του MD5 αλγορίθμου για κρυπτογράφηση και τέλος στην χρήση seed στην γεννήτρια τυχαίων αριθμών. Τέλος, εντοπίστηκε η αποθήκευση σε αρχεία αλλά μετά από έλεγχο διαπιστώθηκε ότι αφορά την αποθήκευση φωτογραφιών.

5.6 Singlesaroundme

Τα αποτελέσματα της ανάλυσης για το SinglesAroundMe είναι τα εξής:

	Singlesaroundme
RawQuery	
SharedPreferences	
FileSave	
DeviceId	
AESCBC	
AESECB	
MD5	
DES	

SecureRandomSeeded	
SecureRandomSetSeed	
WebView	
ModeWorldReadable	
ModeWorldWritable	
CertificateManager	

Πίνακας 7 Αποτελέσματα SinglesAroundMe

Τα προβλήματα που εντοπίστηκαν αφορούν στην χρήση μη ασφαλών αλγορίθμων κρυπτογράφησης (βιβλιοθήκες τρίτων).

5.7 Skout

Τα αποτελέσματα της ανάλυσης για το Skout είναι τα εξής:

	Skout
RawQuery	
SharedPreferences	
FileSave	
DeviceId	
AESCBC	
AESECB	
MD5	
DES	
SecureRandomSeeded	
SecureRandomSetSeed	
WebView	
ModeWorldReadable	
ModeWorldWritable	
CertificateManager	

Πίνακας 8 Αποτελέσματα Skout

Τα προβλήματα που εντοπίστηκαν αφορούν στην χρήση μη ασφαλών αλγορίθμων κρυπτογράφησης (βιβλιοθήκες τρίτων). Επίσης, εντοπίστηκε η αποθήκευση σε αρχεία αλλά μετά από έλεγχο διαπιστώθηκε ότι αφορά την αποθήκευση φωτογραφιών.

5.8 Tagged

Τα αποτελέσματα της ανάλυσης για το Tagged είναι τα εξής:

	Tagged
RawQuery	
SharedPreferences	
FileSave	
DeviceId	
AESCBC	
AESECB	
MD5	
DES	
SecureRandomSeeded	
SecureRandomSetSeed	
WebView	
ModeWorldReadable	
ModeWorldWritable	
CertificateManager	

Πίνακας 9 Αποτελέσματα Tagged

Τα προβλήματα που εντοπίστηκαν αφορούν στην χρήση μη ασφαλών αλγορίθμων κρυπτογράφησης (βιβλιοθήκες τρίτων). Επίσης, εντοπίστηκε η αποθήκευση σε αρχεία αλλά μετά από έλεγχο διαπιστώθηκε ότι αφορά την αποθήκευση φωτογραφιών.

5.9 Tinder

Τα αποτελέσματα της ανάλυσης για το Tinder είναι τα εξής:

	Tinder
RawQuery	
SharedPreferences	
FileSave	
DeviceId	
AESCBC	
AESECB	
MD5	
DES	
SecureRandomSeeded	
SecureRandomSetSeed	
WebView	
ModeWorldReadable	
ModeWorldWritable	

CertificateManager	

Πίνακας 10 Αποτελέσματα Tinder

Τα προβλήματα που εντοπίστηκαν αφορούν στην χρήση μη ασφαλών αλγορίθμων κρυπτογράφησης (βιβλιοθήκες τρίτων). Επίσης, εντοπίστηκε η αποθήκευση σε αρχεία αλλά μετά από έλεγχο διαπιστώθηκε ότι αφορά την αποθήκευση φωτογραφιών.

5.10 sayHi

Τα αποτελέσματα της ανάλυσης για το sayHi είναι τα εξής:

	sayHi
RawQuery	
SharedPreferences	
FileSave	
DeviceId	
AESCBC	
AESECB	
MD5	
DES	
SecureRandomSeeded	
SecureRandomSetSeed	
WebView	
ModeWorldReadable	
ModeWorldWritable	
CertificateManager	

Πίνακας 11 Αποτελέσματα sayHi

Τα προβλήματα που εντοπίστηκαν σε αυτή την εφαρμογή αφορούν στην χρήση αδύναμων αλγορίθμων κρυπτογράφησης (βιβλιοθήκες τρίτων) και τη χρήση seed στην γεννήτρια τυχαίων αριθμών.

5.11 Waplog

Τα αποτελέσματα της ανάλυσης για το Waplog είναι τα εξής:

	Waplog
RawQuery	
SharedPreferences	
FileSave	
DeviceId	
AESCBC	
AESECB	
MD5	
DES	

SecureRandomSeeded	
SecureRandomSetSeed	
WebView	
ModeWorldReadable	
ModeWorldWritable	
CertificateManager	

Πίνακας 12 Αποτελέσματα Waplog

Τα προβλήματα που εντοπίστηκαν σε αυτή την εφαρμογή αφορούν στην χρήση του αλγόριθμου κρυπτογράφησης MD5 (βιβλιοθήκες τρίτων).

5.12 Zoosk

Τα αποτελέσματα της ανάλυσης για το Zoosk είναι τα εξής:

	Zoosk
RawQuery	
SharedPreferences	
FileSave	
DeviceId	
AESCBC	
AESECB	
MD5	
DES	
SecureRandomSeeded	
SecureRandomSetSeed	
WebView	
ModeWorldReadable	
ModeWorldWritable	
CertificateManager	

Πίνακας 13 Αποτελέσματα Zoosk

Τα προβλήματα που εντοπίστηκαν σε αυτή την εφαρμογή αφορούν στην χρήση DeviceID ως αναγνωριστικού της συσκευής, την χρήση του αλγορίθμου κρυπτογράφησης AES/CBC χωρίς την χρήση Padding και την χρήση και άλλων μη ασφαλών αλγορίθμων κρυπτογράφησης (βιβλιοθήκες τρίτων).

5.13 Meetme

Τα αποτελέσματα της ανάλυσης για το MeetMe είναι τα εξής:

	Meetme
RawQuery	
SharedPreferences	
FileSave	
DeviceId	
AESCBC	
AESECB	
MD5	
DES	
SecureRandomSeeded	
SecureRandomSetSeed	
WebView	
ModeWorldReadable	
ModeWorldWritable	
CertificateManager	

Πίνακας 14 Αποτελέσματα Meetme

Τα προβλήματα που εντοπίστηκαν σε αυτή την εφαρμογή αφορούν στην χρήση DeviceID ως αναγνωριστικού της συσκευής και στην χρήση κι άλλων μη ασφαλών αλγορίθμων κρυπτογράφησης (βιβλιοθήκες τρίτων). Τέλος, εντοπίστηκε η αποθήκευση σε αρχεία αλλά μετά από έλεγχο διαπιστώθηκε ότι αφορά την αποθήκευση φωτογραφιών.

5.14 Lovoo

Τα αποτελέσματα της ανάλυσης για το Lovoo είναι τα εξής:

	Lovoo
RawQuery	
SharedPreferences	
FileSave	
DeviceId	
AESCBC	
AESECB	
MD5	
DES	
SecureRandomSeeded	
SecureRandomSetSeed	

WebView	
ModeWorldReadable	
ModeWorldWritable	
CertificateManager	

Πίνακας 15 Αποτελέσματα Lovoo

Ο έλεγχος της εφαρμογής μας κατέληξε στα εξής ευρήματα: Χρήση αδύναμου αλγορίθμου κρυπτογράφησης σε βιβλιοθήκη τρίτου, χρήση seed στην γεννήτρια τυχαίων αριθμών και την αποθήκευση σε αρχεία. Στην τελευταία περίπτωση μετά από έλεγχο διαπιστώθηκε ότι πρόκειται για αποθήκευση φωτογραφιών.

5.15 I-am

Τα αποτελέσματα της ανάλυσης για το I-am είναι τα εξής:

	I-am
RawQuery	
SharedPreferences	
FileSave	
DeviceId	
AESCBC	
AESECB	
MD5	
DES	
SecureRandomSeeded	
SecureRandomSetSeed	
WebView	
ModeWorldReadable	
ModeWorldWritable	
CertificateManager	

Πίνακας 16 Αποτελέσματα I-am

Τα προβλήματα που εντοπίστηκαν σε αυτή την εφαρμογή αφορούν την χρήση του αλγόριθμου κρυπτογράφησης MD5 (βιβλιοθήκες τρίτων). Όσον αφορά την αποθήκευση σε αρχεία, μετά από έλεγχο διαπιστώθηκε ότι πρόκειται για αποθήκευση φωτογραφιών.

5.16 Wechat

Τα αποτελέσματα της ανάλυσης για το Wechat είναι τα εξής:

	Wechat
RawQuery	
SharedPreferences	
FileSave	
DeviceId	
AESCBC	
AESECB	
MD5	
DES	
SecureRandomSeeded	
SecureRandomSetSeed	
WebView	
ModeWorldReadable	
ModeWorldWritable	
CertificateManager	

Πίνακας 17 Αποτελέσματα Wechat

Τα αποτελέσματα μετά τον έλεγχο από την εφαρμογή μας είναι τα εξής: Χρήση μη ασφαλών αλγορίθμων κρυπτογράφησης και χρήση seed στην γεννήτρια τυχαίων αριθμών. Όσον αφορά την αποθήκευση σε αρχεία μετά από έλεγχο διαπιστώθηκε ότι πρόκειται για την αποθήκευση φωτογραφιών.

Κεφάλαιο 6ο

6 Έλεγχος των εφαρμογών από πλατφόρμα τρίτων

Για να βγάλουμε πιο ασφαλή αποτελέσματα, τρέξαμε τεστ στις εφαρμογές μέσα από την πλατφόρμα Mobile Security Framework (MobSF https://github.com/ajinabraham/Mobile-Security-Framework-MobSF) [17]. Χρειάστηκε να εγκαταστήσουμε την πλατφόρμα σε δικό μας server και να εξαχθούν τα αποτελέσματα από εκεί. Ανάμεσα σε αυτή την πλατφόρμα και την δική μας εφαρμογή υπάρχουν κοινά σημεία τα οποία σε πολλές περιπτώσεις επιβεβαίωσαν και τα αποτελέσματα που αναφέρθηκαν παραπάνω. Η πλατφόρμα MobSF έχει την δυνατότητα βέβαια να κάνει έναν πιο γενικό έλεγχο, αναλύοντας τα SSL πιστοποιητικά, τις άδειες και το androidmanifest (γίνεται ανάλυση xml αρχείων της εφαρμογής), τον κώδικα της εφαρμογής, τα api που χρησιμοποιούνται, τα url κ.α. Στην περίπτωση της παρούσας διπλωματικής μας ενδιέφερε η ανάλυση κώδικα και η επιβεβαίωση όσο περισσότερων αποτελεσμάτων μας.

Και στην περίπτωση του MobSF, όπως και στην δική μας εφαρμογή, καταγράφηκαν γενικά ευρήματα τα οποία βρέθηκαν σε όλες τις ελεγχόμενες εφαρμογές. Για παράδειγμα στην δική μας εφαρμογή τέτοια ευρήματα ήταν οι έλεγχοι για RawQuery, SharedPrefernces και WebView. Όσον αφορά το MobSF παρατηρούμε το ίδιο μοτίβο στα εξής ευρήματα:

- The App logs information. Sensitive information.should never be logged.
- Files may contain hardcoded sensitive informations like usernames, passwords, keys etc.
- App uses SQLite Database. Sensitive Information should be encrypted.
- App can read/write to External Storage. Any App can read data written to External Storage.

Πρόκειται για γενικά ευρήματα που προειδοποιούν για ενδεχόμενο πρόβλημα το οποίο πρέπει να εξεταστεί από τον προγραμματιστή.

Το πρώτο και το δεύτερο από τα παραπάνω αναφέρονται σε αρχεία τα οποία ίσως να έχουν αποθηκευμένα ευαίσθητα δεδομένα. Για να ελέγξουμε την ορθότητα των αποτελεσμάτων αυτών, εγκαταστήσαμε τις εφαρμογές σε rootαρισμένη συσκευή και αντιγράψαμε όλα τα δεδομένα της εκάστοτε εφαρμογής. Δεν βρέθηκαν ευαίσθητα δεδομένα στα αρχεία αυτά, τουλάχιστον σε σχέση με αυτά που αναφέρονται στην περιγραφή. Δεν υπήρχαν username, ούτε κωδικοί, κλειδιά κτλ. Τα πιο ανησυχητικά ευρήματα ήταν κάποια token που αφορούσαν τον τρόπο με τον οποίο είχε γίνει login (συγκεκριμένα fb-login) και session-tokens.

Όσον αφορά τις SQLite βάσεις δεδομένων, αυτό το εύρημα υπάρχει σε όλα τα report των εφαρμογών. Με τον ίδιο τρόπο εμείς είχαμε ως εύρημα την χρήση του rawQuery που προϋποθέτει την ύπαρξη βάσης δεδομένων. Όπως εξηγήσαμε και σε προηγούμενο κεφάλαιο, τα ευαίσθητα δεδομένα θα πρέπει όχι μόνο να κρυπτογραφούνται αλλά αυτό να γίνεται με ασφαλείς αλγορίθμους.

Τέλος, όσον αφορά το δικαίωμα για εγγραφή στην κάρτα sd, επίσης αναφέρθηκε παραπάνω το πρόβλημα με τα ευαίσθητα δεδομένα.

Παρακάτω δεν γίνεται αναφορά στα προαναφερθέντα ευρήματα του MobSF καθώς αποτελούν πολύ γενικές διαπιστώσεις.

6.1 Grindr

Σε αυτή την εφαρμογή επιβεβαιώθηκε η χρήση ανασφαλούς γεννήτριας τυχαίων αριθμών. Η πλατφόρμα όμως δεν μπόρεσε να εκτελέσει κάποιον έλεγχο για μη ασφαλείς αλγορίθμους κρυπτογράφησης.

6.2 Hornet

Στο Hornet επιβεβαιώθηκε η χρήση webview η οποία μπορεί να οδηγήσει σε προβλήματα ασφαλείας.

6.3 Immomo

Η πλατφόρμα MobSF επιβεβαίωσε την ύπαρξη webview και το ενδεχόμενο να υπάρχει πρόβλημα ασφαλείας σε αυτό καθώς επίσης το πρόβλημα με την χειροκίνητη εισαγωγή πιστοποιητικών.

6.4 Pof

Η πλατφόρμα MobSF επιβεβαίωσε την ύπαρξη webview και το ενδεχόμενο να υπάρχει πρόβλημα ασφαλείας σε αυτό καθώς επίσης το πρόβλημα με την γεννήτρια τυχαίων αριθμών.

6.5 Chaton

Η πλατφόρμα MobSF επιβεβαίωσε το πρόβλημα με το webview καθώς και με την γεννήτρια τυχαίων αριθμών. Αντιθέτως, δεν κατάφερε να καταγράψει τα προβλήματα που υπάρχουν με την χρήση του DeviceID και την χρήση του ανασφαλούς MD5.

6.6 Singlesaroundme

Σε αυτή την εφαρμογή δεν βρέθηκαν ιδιαίτερες συγκλίσεις στα αποτελέσματα πέραν των γενικών προειδοποιήσεων.

6.7 Skout

Στο skout επιβεβαιώθηκε η χρήση webview η οποία μπορεί να οδηγήσει σε προβλήματα ασφαλείας.

6.8 Tagged

Στο Tagged επιβεβαιώθηκε η χρήση webview η οποία μπορεί να οδηγήσει σε προβλήματα ασφαλείας.

6.9 Tinder

Στο tinder επιβεβαιώθηκε η ύπαρξη προβλήματος στην γεννήτρια τυχαίων αριθμών.

6.10 Sayhi

Στο SayHI επιβεβαιώθηκε η χρήση webview η οποία μπορεί να οδηγήσει σε προβλήματα ασφαλείας και το πρόβλημα με την γεννήτρια τυχαίων αριθμών.

6.11 Waplog

Στο Waplog επιβεβαιώθηκε η χρήση webview η οποία μπορεί να οδηγήσει σε προβλήματα ασφαλείας.

6.12 Zoosk

Σε αυτή την εφαρμογή δεν βρέθηκαν ιδιαίτερες συγκλίσεις στα αποτελέσματα πέραν των γενικών προειδοποιήσεων. Επιπρόσθετα, στην εφαρμογή μας βρέθηκαν σημαντικά προβλήματα τα οποία δεν ελέγχει η πλατφόρμα MobSF και συγκεκριμένα εντοπίστηκε η χρήση DeviceID καθώς και χρήση αλγορίθμου κρυπτογράφησης χωρίς padding.

6.13 Meetme

Στο Meetme επιβεβαιώθηκε η χρήση webview η οποία μπορεί να οδηγήσει σε προβλήματα ασφαλείας. Αντίθετα δεν μπόρεσε να επιβεβαιωθεί η χρήση του DeviceID.

6.14 Lovoo

Η πλατφόρμα MobSF επιβεβαίωσε την ύπαρξη webview και το ενδεχόμενο να υπάρχει πρόβλημα ασφαλείας σε αυτό καθώς επίσης το πρόβλημα με την γεννήτρια τυχαίων αριθμών.

6.15 I-am

Στο I-am επιβεβαιώθηκε η χρήση webview η οποία μπορεί να οδηγήσει σε προβλήματα ασφαλείας.

6.16 WeCHat

Στο WeChat επιβεβαιώθηκε η χρήση seed στην γεννήτρια αριθμών και η ύπαρξη webview που μπορεί να αποτελέσει πρόβλημα ασφαλείας. Αντίθετα, δεν μπόρεσε το MobSF να επιβεβαιώσει την χρήση αδύναμων αλγορίθμων κρυπτογράφησης.

Κεφάλαιο 7ο

7 Συμπεράσματα

Καταρχάς τα προβλήματα τα οποία καταφέρνει να εντοπίσει η εφαρμογή αποτελούν warnings ώστε ο χρήστης να αξιολογήσει το κόστος που μπορεί να έχει μία ενδεχόμενη απώλεια/διαρροή προσωπικών του δεδομένων και ενδεχομένως να σταματήσει να χρησιμοποιεί την android εφαρμογή. Επίσης, αποτελεί ένα σημαντικό εργαλείο ώστε ο δημιουργός της κάθε mobile εφαρμογής να μπορεί να τρέχει κάποια αυτοματοποιημένα τεστ ώστε να διορθώσει τα προβλήματα που μπορεί να υπάρχουν. Τα προβλήματα αυτά αφορούν κυρίως προβλήματα εμπιστευτικότητας και ειδικότερα την αποθήκευση δεδομένων της εφαρμογής τοπικά. Τα πιο σημαντικά ευρήματα από το σύνολο των ελεγχόμενων είναι η χρήση μη ασφαλών αλγορίθμων κρυπτογράφησης και η χρήση seed στην γεννήτρια τυχαίων αριθμών.

Με την πρόοδο του λειτουργικού συστήματος και των API, την εξέλιξη της τεχνολογίας και την αύξηση της χρήσης από τους χρήστες προκύπτουν συνεχώς νεότερα προβλήματα ασφάλειας που πρέπει να επιλυθούν και καινούρια εργαλεία για να γίνεται ο εντοπισμός τους. Είναι χαρακτηριστικά τα παραδείγματα με τις υποκλοπές μέσω των NFC chip τα οποία είναι by default ενεργοποιημένα και θα μπορούσε κάποιος εξ ’επαφής να υποκλέψει δεδομένα. Θα πρέπει να υπάρχει authentication με κάποιον ασφαλή τρόπο ώστε να γίνεται κάποια ενέργεια μέσω NFC και οπωσδήποτε τα ευαίσθητα δεδομένα να είναι κρυπτογραφημένα. Επίσης, είναι γνωστά τα προβλήματα με τους αισθητήρες ανάγνωσης δακτυλικών αποτυπωμάτων όπου πολλοί κατασκευαστές κινητών αποθήκευαν τα βιομετρικά στοιχεία ακρυπτογράφητα (προσβάσιμα από κακόβουλους χρήστες/εφαρμογές). Επιπλέον, φαινομενικά “αθώες” εφαρμογές θα μπορούσαν να αποκτήσουν πρόσβαση σε δεδομένα στα οποία κανονικά δεν θα έπρεπε. Τέλος, όπως αναφέρθηκε και στον πρόλογο και βασιζόμενοι σε προηγούμενες εργασίες υπάρχει σοβαρό πρόβλημα με την λήψη των στοιχείων της τοποθεσίας και πως διαχειρίζονται αυτές οι πληροφορίες. Παραδείγματος χάρη δεν είναι αναγκαία η αποστολή της τοποθεσίας [18] [19] [20] κάποιου ώστε να υπολογιστεί η απόσταση του από κάποιον άλλο χρήστη ή σημείο ενδιαφέροντος αλλά υπάρχουν αλγόριθμοι που μπορούν να το κάνουν χωρίς να κινδυνεύει η εμπιστευτικότητα δεδομένων.

Η λύση για κάθε πρόβλημα που υφίσταται στο παρόν ή θα προκύψει στο μέλλον είναι οι προγραμματιστές να ακολουθούν τις οδηγίες του δημιουργού του εκάστοτε λειτουργικού και SDK (στην περίπτωσή μας το android). Μόνο μέσα από τις βέλτιστες αυτές πρακτικές θα μπορεί ο προγραμματιστής να νιώθει όσο το δυνατόν πιο σίγουρος για την ποιότητα και την ασφάλεια της εφαρμογής του, ενώ θα είναι πιο εύκολο σε περίπτωση προβλήματος να ακολουθήσει τις αναβαθμισμένες οδηγίες του δημιουργού ώστε να επαναφέρει την εφαρμογή του στα ανώτερα επίπεδα ασφάλειας.

8 Βιβλιογραφικές Πηγές

[1] R. Stevens, «Investigating user privacy in android ad libraries.,» Workshop on Mobile Security Technologies (MoST), 2012.

[2] B. Theodore, A. Pridgen και D. S. Wallach, Longitudinal analysis of android ad library permissions, arXiv, 2013.

[3] W. Zhou, Y. Zhou, X. Jiang και P. Ning, «Detecting repackaged smartphone applications in third-party android marketplaces,» Proceedings of the second ACM conference on Data and Application Security and Privacy, 201.

[4] C. Patsakis, A. Zigomitros και A. Solanas, «Analysis of Privacy and Security Exposure in Mobile Dating Applications,» International Conference on Mobile, Secure and Programmable Networking (MSPN'2015), Paris, June 15-17 2015.

[5] Q. Qin, C. Patsakis και M. Bouroche, «Playing Hide and Seek with Mobile Dating Applications,» 29th IFIP SEC 2014 International Conference ICT Systems on Security and Privacy Protection, Marrakech, 2-4 June 2014.

[6] G. Clint, «AndroidLeaks: automatically detecting potential privacy leaks in android applications on a large scale,» Springer Berlin Heidelberg, 2012.

[7] P. P. Chan, L. C. Hui και S.-M. Yiu, «Droidchecker: analyzing android applications for capability leak,» Proceedings of the fifth ACM conference on Security and Privacy in Wireless and Mobile Networks, 2012.

[8] [Ηλεκτρονικό]. Available: http://developer.android.com/training/articles/security-tips.html.

[9] [Ηλεκτρονικό]. Available: http://www.androidauthority.com/how-to-hide-your-api-key-in-android-600583/.

[10] [Ηλεκτρονικό]. Available: https://www.owasp.org/index.php/Guide_to_Cryptography.

[11] [Ηλεκτρονικό]. Available: http://tozny.com/blog/encrypting-strings-in-android-lets-make-better-mistakes/.

[12] «https://pdos.csail.mit.edu/papers/cryptobugs:apsys14.pdf,» [Ηλεκτρονικό]. Available: https://pdos.csail.mit.edu/papers/cryptobugs:apsys14.pdf.

[13] [Ηλεκτρονικό]. Available: https://blog.skullsecurity.org/2013/padding-oracle-attacks-in-depth.

[14] [Ηλεκτρονικό]. Available: http://android-developers.blogspot.gr/2013/08/some-securerandom-thoughts.html.

[15] [Ηλεκτρονικό]. Available: http://android-developers.blogspot.gr/2011/03/identifying-app-installations.html.

[16] [Ηλεκτρονικό]. Available: https://appvigil.co/blog/secure-android-app-development-best-practices/.

[17] [Ηλεκτρονικό]. Available: https://github.com/ajinabraham/Mobile-Security-Framework-MobSF.

[18] P. Kotzanikolaou, C. Patsakis, E. Magkos και M. Korakakis, «Lightweight Private Proximity Testing for Geospatial Social Networks, Computer Communications».

[19] A. Narayanan, «Location Privacy via Private Proximity Testing,» NDSS, 2011.

[20] C. Patsakis, P. Kotzanikolaou και M. Bouroche, «Private Proximity Testing on Steroids: An NTRU-based protocol,» 11th International Workshop on Security and Trust Management (STM 2015), Vienna, September 21-22, 2015.

9 Παράρτημα