Τεχνολογικό Εκπαιδευτικό Ίδρυμα Ηπείρου
Τμήμα Μηχανικών Λογισμικού

Πτυχιακή εργασία Εαρινού εξαμήνου 2016
Θέμα:

Ανάλυση και προστασία των εφαρμογών Android από επιθέσεις εισαγωγής κώδικα

Φοιτητής: Κιοσέογλου Γρηγόριος (ΑΜ: 13526)
Επιβλέπων καθηγητής: Λιάγκου Βασιλική

Περιεχόμενα

Πίνακας Σχημάτων

1. Περίληψη

Οι κινητές συσκευές μπήκαν στην καθημερινότητα των ανθρώπων πριν από πολλά χρόνια, όμως τα τελευταία χρόνια έχουν γίνει καθημερινή συνήθεια, έχοντας ξεπεράσει ακόμα και την χρήση των υπολογιστών. Τα κυριότερα λειτουργικά της σημερινής εποχής, Android και iOS. Με την νέα τεχνολογία και τα εργαλεία που υπάρχουν διαθέσιμα στο διαδίκτυο ο καθένας μπορεί να δημιουργήσει μία εφαρμογή για Android ή iOS ή ακόμη και ένα απλό πρόγραμμα για Windows. Παρ' όλα αυτά κάποιος που δεν ξέρει μπορεί να αφήσει μια πίσω πόρτα ανοιχτή, στο λογισμικό του, την οποία κάποιος με κακό σκοπό, θα την χρησιμοποιήσει. Έτσι υλικό θα εκτεθεί σε χέρια που δεν πρέπει και επίσης θα εκτεθεί δημόσια και πολύς κόσμος.

Λέξεις κλειδιά: SQLinjection, Android, OS, Java, App, Security, PHP

2. Μέθοδος συλλογής πληροφοριών

Για την συλλογή πληροφοριών χρησιμοποίησα το την μηχανή αναζήτησης της Google βάζοντας της λέξεις κλειδιά. Από τα αποτελέσματα που μου έβγαλε πήρα και άλλες πηγές για περισσότερο υλικό. Επίσης μελετήθηκαν και άρθρα τα οποία μου δόθηκαν από την επιβλέπων καθηγήτρια.

Τα κομμάτια κώδικα τα οποία θα επιδειχθούν στην εργασία αυτή, θα έχουν γραμματοσειρά Courier New και πλάγια γραφή όπως ακολουθεί:

```
παράδειγμα κώδικα για την εργασία
```

Με σκοπό να ξεχωρίζει, ο αναγνώστης, τον κώδικα από το κυρίως κείμενο.

3. Εισαγωγή

Δεν είναι συνήθεις φαινόμενο να δεχόμαστε επίθεση από hackers, όσο περιηγούμαστε στο διαδίκτυο, με σκοπό της κλοπή των προσωπικών μας δεδομένων. Παρ' όλα αυτά δεν είναι και απίθανο να συμβεί στον οποιοδήποτε Σχεδόν τα δύο τρίτα των ενηλίκων της Αμερικής κατέχουν κινητό τύπου smartphone (1) . Οι περισσότεροι, από αυτούς, τα χρησιμοποιούν ως σημείο εισόδου στο διαδίκτυο. Τα κινητά αυτά είναι γεμάτα με προσωπικές πληροφορίες που κάποιος θα μπορούσε να κλέψει για να τα χρησιμοποιήσει εναντίων τους ή για προσωπικό του κέρδος. Επίσης μπορεί να κλαπούν και απόρρητες πληροφορίες από εταιρίες, με τους ειδικούς να υποστηρίζουν, ότι οι περισσότερες κλοπές τέτοιων πληροφοριών σε ασφαλές δίκτυο γίνονται είναι από USB. Το 33% των επαγγελματιών IT security, ανησυχεί ότι τα δεδομένα κλέβονται μέσω χρήσης συσκευών USB(2), κατα κύριο λόγο από κάποιον εργάτη ή πρώην εργάτη, που θέλει να βλάψει την εταιρία ή να κερδίσει κάποια εξτρα χρήματα πουλώντας τις πληροφορίες αυτές. Το Android με την επιλογή για λειτουργία USB, που έχει, θα μπορούσε να ανήκει σε αυτή την κατηγορία.

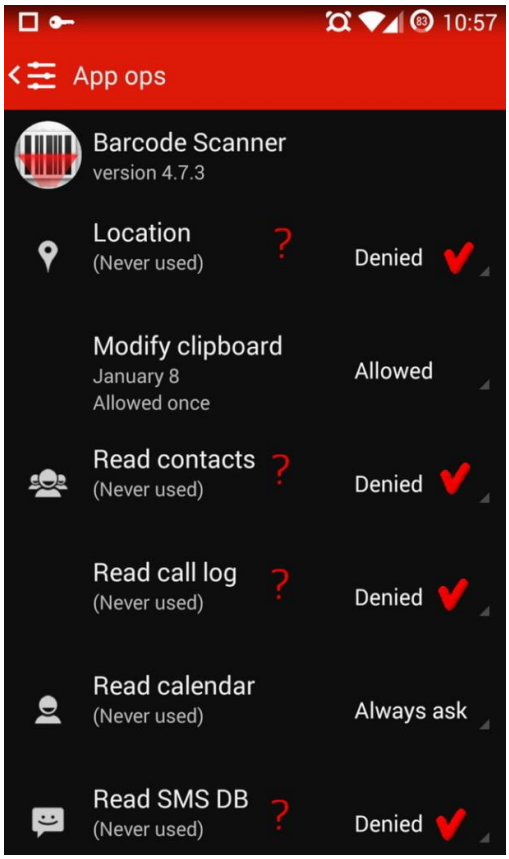
Το Android είναι ένα λειτουργικό σύστημα κινητών συσκευών, που βασίζεται στον πυρήνα του Linux και αναπτύσσεται από την Google (4). Η διεπαφή του είναι σχεδιασμένη για οθόνες αφής φορητών συσκευών όπως smartphones και υπολογιστές tablet. Επίσης έχει εξειδικευμένες διεπαφές για τηλεοράσεις, αυτοκίνητα και ρολόγια χειρός (Android wear). Το λειτουργικό χρησιμοποιεί την οθόνη αφής για να καταλάβει κινήσεις όπως το σύρσιμο ή χτύπημα για να χειριστεί τα αντικείμενα της οθόνης και το εικονικό πληκτρολόγιο. Παρά το γεγονός ότι έχει δημιουργηθεί για οθόνες αφής, μπορούμε να βρούμε λειτουργικό σύστημα Android σε παιχνιδομηχανές, ψηφιακές φωτογραφικές κάμερες και άλλες ηλεκτρικές συσκευές. Το λειτουργικό Android είναι το πιο ευρέως διαδεδομένο λειτουργικό κινητών συσκευών του 2013, και το 2014 οι πωλήσεις του ήταν περισσότερες από τις πωλήσεις των συσκευών Windows και iOS μαζί, σύμφωνα με την Jay Yarow (3).

Ο λόγος που το Android είναι τόσο ευάλωτο σε επιθέσεις injection είναι ότι δεν υπάρχει κάποιος τρόπος παρακολούθησης του τρέχον νήματος στην CPU. Σε αυτή την πτυχιακή θα γίνει αναφορά εάν μπορούν να γίνουν αυτές οι επιθέσεις σε εξωτερικό server εφαρμογής, χρησιμοποιώντας την ίδια την εφαρμογή και το αν ή πως μπορεί να γίνει ο κώδικας πιο ασφαλές με χρήση PHP και XML.

Σε αυτήν την εργασία έχει επιλεχθεί σαν αντικείμενο μελέτης το λειτουργικό σύστημα του Android, διότι είναι το διαδεδομένο λειτουργικό σύστημα κινητών συσκευών. Παρ' ότι έχει όμως τόσο μεγάλη διάδοση, λόγο του ότι είναι αρκετά προσαρμοζόμενο, έχει αρκετά κενά ασφαλείας. Στην εργασία, μελετούνται μερικά από τα μεγαλύτερα κενά ασφαλείας που μαστίζουν το λειτουργικό σύστημα του Android. Επίσης μελετούνται οι τρόποι αποτροπής των επιθέσεων καθώς και μερικά μέτρα ασφαλείας κατά αυτών των επιθέσεων. Έχει δοθεί ιδιαίτερη μελέτη στην επίθεση τύπου SQL injection, με χρήση της εφαρμογής για την είσοδο των μη ασφαλών δεδομένων και σε τρόπους που μπορεί να αντιμετωπιστεί. Για τις ανάγκες της παρουσίασης έχει, δημιουργηθεί εφαρμογή Android η οποία προσομοιώνει τον τρόπο της επίθεσης και με χρήση PHP scripts την αποτρέπει.

3.1 Ασφάλεια στην πλατφόρμα του Android

Η ασφάλεια των συσκευών Android δεν πρέπει να θεωρείτε δεδομένη. Επειδή είναι σε κινητή πλατφόρμα δεν σημαίνει ότι υπάρχει λιγότερος κίνδυνος αν ανοιχτεί ένα e-mail με άγνωστο αποστολέα. Το θετικό με την πλατφόρμα του Android είναι ότι όλες οι εφαρμογές έχουν ένα μοναδικό user-id, με το οποίο είναι ανεβασμένες στο Google Play Store, ή στην περίπτωση που δεν είναι ανεβασμένο, το παίρνουν κατά την διάρκεια της ανάπτυξης τους. Εκτός από το user-id απαιτούν και δικαιώματα χρήσης, τα οποία τα βάζει ο προγραμματιστής κατά την διάρκεια της ανάπτυξης της εφαρμογής και τα δέχεται ο χρήστης εάν θέλει και δει ότι είναι αναγκαίο. Βεβαίως υπάρχουν περιπτώσεις που αν ο χρήστης δεν δεχθεί τα δικαιώματα δεν θα γίνει η εγκατάσταση της εφαρμογής (4) και άλλες που απλά δεν θα είναι ενεργά κάποια στοιχεία της εφαρμογής, π.χ δεν θα δουλεύει η κάμερα ή το μικρόφωνο. Σύμφωνα με το Εργαστήριο ασφαλείας υπολογιστικών συστημάτων ESET, πολλές εφαρμογές ζητούν περισσότερα δικαιώματα από αυτά που πραγματικά χρειάζονται για να λειτουργήσουν (5) . Ακόμη μερικές εφαρμογές, μπορεί να έχουν κρυφά δικαιώματα τα οποία δεν εμφανίζονται κατά την διάρκεια της εγκατάστασης της εφαρμογής (7).



Σχήμα 1: Δικαιώματα μη ασφαλούς εφαρμογής (4)

Με αυτό τον τρόπο, μια εφαρμογή ραδιοφώνου θα μπορούσε να στέλνει την θέση του χρήστη της εφαρμογής, χωρίς να το ξέρει, σε διάφορους διαφημιστές ή ακόμη και να στείλει τα μηνύματα και διάφορες άλλες προσωπικές πληροφορίες του χρήστη σε τρίτα άτομα. Μια καλή λύση για να αποφευχθούν τέτοιου είδους εφαρμογές θα ήταν η εγκατάσταση αντι-ιικού λογισμικού στην συσκευή.

3.2 Προβλήματα ασφάλειας σε android περιβάλλον

Το android είναι μία σχετικά νέα πλατφόρμα ανάπτυξης εφαρμογών. Όπως είναι λογικό με όλες τις πλατφόρμες όταν είναι στα πρώιμα στάδια τους, υπάρχουν αρκετές τρύπες λογισμικού. Οι αναλυτές κάνουν μια αρκετά μεγάλη προσπάθεια για να βρουν τις βασικές και πιο επικίνδυνες απειλές ώστε να μπορέσουν να τις αντιμετωπίσουν. Μέχρι στιγμής έχουν βρεί κάποιες από αυτές και τις έχουν μοντελοποιήσει. Σύμφωνα με τον OWASP οι 10 πιο επικίνδυνες απειλές του 2014 (31) είναι:

- M1: Weak Server Side Controls
- M2: Insecure Data Storage
- M3: Insufficient Transport Layer Protection
- M4: Unintended Data Leakage
- M5: Poor Authorization and Authentication
- M6: Broken Cryptography
- M7: Client Side Injection
- M8: Security Decisions Via Untrusted Inputs
- M9: Improper Session Handling
- M10: Lack of Binary Protections

Θα δούμε τα προβλήματα όπως και τρόποι αντιμετώποισης τους σε σε μεγαλύτερο βάθος σε αργότερα κεφάλαια.

3.3 Αναγκαιότητα χρήσης μέτρων - εργαλείων ασφάλειας σε android

Λόγο του ότι το Android είναι ένα λογισμικό για κινητές συσκευές, είναι αρκετά ευαίσθητες οι πληροφορίες που εισέρχονται και εξέρχονται. Αυτό με την σειρά του σημαίνει ότι σε μερικά χέρια αυτές οι πληροφορίες θα είναι αρκετά κερδοφόρες. Για να αποτραπούν κλοπές τέτοιον πληροφοριών, ακόμη και σε αυτό το στάδιο μιας νέας πλατφόρμας λογισμικού, θα πρέπει να παρθούν μέτρα για την ασφάλεια των πληροφοριών ή να δημιουργηθούν κάποια εργαλεία. Ακόμη και αν δεν είναι 100% αποτελεσματικά στην αρχή, οι κλοπές πληροφοριών θα μειωθούν δραστικά, διότι από τις μηδενικές άμυνες, θα εμφανιστούν κάποιες. Με την πάροδο του χρόνου αυτές οι μέθοδοι και τα εργαλεία θα μπορέσουν να τελοιοποιηθούν και να σταματήσει για πάντα η κλοπή πληροφοριών με στόχο το κέρδος.

3.3.1 Επιθέσεις και Τρύπες Ασφαλείας

Πιο πάνω στην εργασία είδα ονομαστικά τα 10 πιο επικίνδυνα προβλήματα που έχουν βρεθεί για την πλατφόρμα του Android. Σε αυτό το κεφάλαιο θα δούμε πιο αναλυτικά που συναντώνται και πώς μπορεί να γίνει επίθεση χρησιμοποιώντας την κάθε τρωτότητα της πλατφόρμας.

M1: Weak Server Side Controls: Όπως είναι εμφανές και από το όνομα, αυτή η τρωτότητα εμφανίζεται, έξω από την συσκευή του κινητού, στους servers που κάνουν εξυπηρέτηση της

εφαρμογής. Η απειλή περιλαμβάνει οποιαδήποτε οντότητα που δρα ως πηγή αναξιόπιστης εισόδου στο API ή το backend της εφαρμογής. Παράδειγμα τέτοιας οντότητας είναι ένας χρήστης ή κάποιο κακόβουλο λογισμικό. Η δυνατότητα εκμετάλλευσης είναι αρκετά εύκολη και αρκετά μεγάλη. Περιλαμβάνει, επίθεση Brute force: αργή και εξαντλητική, περιλαμβάνει όλες τις πιθανές τιμές που θα μπορούσε να χρησιμοποιήσει ένας χρήστης. Cache Poisoning: Επίθεση που επιδιώκει να εισάγει ψευδές ή κακόβουλο δεδομένο σε μία μνήμη web cache, συνήθως με την χρήση του πρωτοκόλλου HTTP και DNS Poisoning: Επίθεση η οποία προσπαθεί να εισάγει ψευδής πληροφορίες σε έναν διακομιστή DNS ώστε να πάει σε διαφορετική σελίδα από αυτή που θα έπρεπε. Σελίδα όπου ανήκει σε άλλους χρήστες και επιτρέπει την συλλογή πληροφοριών. Παράδειγμα τέτοιας επίθεσης είναι το Phishing.

M2: Insecure Data Storage: Τρωτότητα η οποία χτυπάει τις βάσεις δεδομένων μέσα στη συσκευή Android με στόχο την κλοπή των δεδομένων. Η απειλή συνήθως περιλαμβάνει μία εφαρμογή - κλώνο μιας εμπορικής εφαρμογής η οποία ενεργή για λογαριασμό κάποιου τρίτου. Ακόμη θα μπορούσε να είναι και κάποιος άνθρωπος που έχει βρει ένα χαμένο ή κλεμμένο κινητό. Η δυνατότητα εκμετάλλευσης αυτής της τρωτότητας είναι αρκετά εύκολη, ειδικά στην 2η περίπτωση που κάποιος κλέβει ένα κινητό. Ο δράστης συνδέει το κινητό σε έναν υπολογιστή και με την χρήση ελεύθερων 3ων (3rd party applications) εφαρμογών, μπορεί να μπει, να δει και να αλλάξει καταλόγους στους οποίους υπάρχουν τα αποθηκευμένα δεδομένα του χρήστη.

M3: Insufficient Transport Layer Protection: Σε μια εφαρμογή για κινητές συσκευές είναι συνήθης φαινόμενο τα δεδομένα να ανταλλάσσονται με ένα μοντέλο client - server. Όταν μία πλευρά θα θελήσει να μεταδώσει δεδομένα, αυτά θα πρέπει να διασχίσουν το δίκτυο κινητής τηλεφωνίας και το Internet για να φτάσουν στον προορισμό τους. Όσο τα δεδομένα διανύουν την απόσταση μέσα από τα καλώδια των τηλεφωνικών εταιριών, μπορεί να υπάρξει η απειλή, κάποιος να εκμεταλλευτεί κάποια τρωτά σημεία και να υποκλέψει ευαίσθητα δεδομένα. Συνήθως η απειλή περιλαμβάνει, έναν κακόβουλο χρήστη να μοιράζεται το τοπικό δίκτυο που είναι παραβιασμένο ή παρακολουθείτε, συσκευές δικτύου, π.χ ρούτερ ή κεραίες μεταφοράς, ή τέλος Malware στην κινητή συσκευή. Φυσικά και είναι πολύ δύσκολο κάποιος να παρακολουθεί και να υποκλέψει πληροφορία από ένα τόσο μεγάλο δίκτυο όπως ένα δίκτυο κινητών τηλεπικοινωνιών Όμως η παρακολούθηση ενός τοπικού δικτύου Wi-Fi είναι αρκετά εύκολο.

M4: Unintended Data Leakage: Η πιο συνηθισμένη απειλή για κινητές συσκευές. Οι τρόποι που μπορεί να επιτευχθεί η επίθεση, περιλαμβάνουν, ένα malware κινητής συσκευής ή την φυσική πρόσβαση ενός χρήστη στην συσκευή του θύματος. Ακόμη ένα σενάριο επίθεσης περιλαμβάνει τροποποιημένες εκδόσεις εμπορικών εφαρμογών. Ο χρήστης που έχει την ελεύθερη φυσική πρόσβαση μπορεί να χρησιμοποιήσει τα πολλά ελεύθερα εργαλεία, για να κάνει την δουλειά του. Αν γίνει η επίθεση γίνει μέσω κακόβουλου κώδικα, ο χρήστης θα χρησιμοποιήσει το πλήρες επιτρεπτό και τεκμηριωμένο API κάποιας εφαρμογής, ώστε να επιτευχθεί η επίθεση.

M5: Poor Authorization and Authentication: Οι παράγοντες εκμεταλλεύονται τα τρωτά σημεία ελέγχου ταυτότητας, η επίθεση γίνεται συνήθως με αυτοματοποιημένες μεθόδους ή/ και με διαθέσιμα εργαλεία τα οποία προσαρμόζονται εύκολα. Για να γίνει μια τέτοια επίθεση, ο επιτιθέμενος πρέπει να έχει καταλάβει πως γίνεται η διαδικασία της αυθεντικοποίησης, έτσι ώστε με υποβολή ορισμένων αιτήσεων στον server, να παρακάμψει την διαδικασία.

M6: Broken Cryptography: Η συγκεκριμένη απειλή περιλαμβάνει μία οντότητα η οποία έχει πρόσβαση στα δεδομένα, τα οποία, δεν έχουν κρυπτογραφηθεί σωστά. Ο τρόπος την

επίθεσης είναι η αποκρυπτογράφηση των δεδομένων σε φυσικό επίπεδο ή κακόβουλες εφαρμογες που έχουν πρόσβαση σε αυτά τα δεδομένα.

M7: Client Side Injection: Ένα παράδειγμα αυτής της τρωτότητας είναι η αποστολή μη αξιόπιστου κώδικα απο κάποιον χρήστη. Αυτό περιλαμβάνει εξωτερικούς χρήστες, εσωτερικούς χρήστες, την ίδια την εφαρμογή ή κάποιο 3ό λογισμικό, συνήθως κακόβουλο, μέσα στο κινητό. Ο επιτιθέμενος εκμεταλλεύεται την τρωτότητα κάποιου διερμηνέα μέσα στην κινητή εφαρμογή. Σχεδόν κάθε πηγή δεδομένων σε μία κινητή συσκευή, μπορεί να είναι φορέας του Injection. Αυτό περιλαμβάνει ακόμη και την ίδια την εφαρμογή.

M8: Security Decisions Via Untrusted Inputs: Και σε αυτή την τρωτότητα, η απειλη περιλαμβάνει την εισαγωγή μη ασφαλούς κώδικα, όμως αυτή την φορά σε ζωτικές, για την εφαρμογή κλήσεις μεθόδων. Αυτή η τρωτότητα περιλαμβάνει σαν επιτιθέμενες οντότητες, χρήστες και malware, όμως δεν περιορίζονται μόνο σε αυτές. Όπως είναι λογικό, ένας εισβολέας με πρόσβαση στην εφαρμογή, μπορεί να υποκλέψει πληροφορίες ή να χειραγωγήσει τα αποτελέσματα μέσω αλλοίωσης των δεδομένων.

M9: Improper Session Handling: Αυτό είναι μια αρκετά γενική τρωτότητα, διότι περιλαμβάνει οποιαδήποτε εφαρμογή και οποιονδήποτε χρήστη που έχει πρόσβαση στο HTTP/S πρωτόκολλο και στα cookies. Ο επιτιθέμενος χρειάζεται να έχει πρόσβαση στην φυσική συσκευή ή να παρακολουθεί το δίκτυο, έτσι ώστε να αλλάξει τα δεδομένα του cookie, που υπάρχει για την αυθεντικοποίηση και να μπορεί ο επιτιθέμενος να υποκλέπτει πληροφορίες όταν υπάρχει ανταλλαγή με τον διακομιστή.

M10: Lack of Binary Protections: Ο επιτιθέμενος θα προσπαθήσει να αναλύσει και να αποσυμπιλήσει, χρήση αντίστροφης μηχανικής , την εφαρμογή. Από εκεί και πέρα θα του είναι εύκολο να τροποποιήσει τον κώδικα της εφαρμογής και να προσθέσει κάποια κρυφή λειτουργικότητα. Ο επιτιθέμενος συνήθως χρησιμοποιεί κάποιο αυτοματοποιημένο εργαλείο για να αναστρέψει την λειτουργικότητα της εφαρμογής και να την προγραμματίσει έτσι ώστε να εκτελεί κρυφές λειτουργείες.

3.3.2 Εργαλεία για ενίσχυση επιπέδου ασφάλειας και αποτροπή επιθέσεων

Μερικές από τις παραπάνω τρωτότητες είναι πολύ εύκολο να χρησιμοποιηθούν, ακόμη και ο πιο άπειρος χρήστης να πάρει πληροφορίες ανενόχλητος. Από την άλλη πλευρά, αυτές οι τρωτότητες είναι πολύ εύκολο να καλυφθούν από διάφορα εργαλεία ασφαλείας, ώστε να αποφευχθεί η απώλεια δεδομένων. Κάθε εφαρμογή δεν έχει τις ίδιες ανάγκες, οπότε το καλύτερο εργαλείο που μπορεί να χρησιμοποιήσει κάποιος, είναι ο προγραμματιστής, ή η ομάδα προγραμματιστών, που έχει γράψει την εφαρμογή. Παρακάτω θα δούμε μερικούς τρόπους αποφυγής έκθεσης της εφαρμογής στις παραπάνω τρωτότητες.

M1: Για να εμφανιστεί αυτό το πρόβλημα, θα πρέπει η εταιρία ή ομάδα να βγάλει μία εφαρμογή web ή API έτοιμη για χρήση από τις κινητές συσκευές. Για την αποφυγή εισαγωγής μη αξιόπιστων δεδομένων, πρέπει να χρησιμοποιηθούν πρακτικές ασφαλής κωδικοποίηση και διαμόρφωση στους όλους τους διακομιστές της εφαρμογής του κινητού.

M2: Η εμφάνιση αυτού του προβλήματος γίνεται όταν η ομάδα ανάπτυξης υποθέσει ότι κανείς δεν μπορεί να μπει στο σύστημα αρχείων του κινητού. Αυτό είναι ένας μεγάλο λάθος. Το σύστημα αρχειοθέτησης είναι εύκολα προσβάσιμο σε όλους. Ο πιο αυστηρός κανόνας, που δεν πρέπει να σπάει ποτέ, στη ανάπτυξη εφαρμογών για κινητές συσκευές είναι, ότι ποτέ δεν πρέπει να αποθηκεύονται δεδομένα στο κινητό, εκτός εάν είναι απολύτως

απαραίτητο. Ο προγραμματιστής θα πρέπει να υποθέσει ότι τα δεδομένα είναι εκτεθειμένα με το που φθάσουν στο τηλέφωνο. Μία τεχνική αποφυγής αυτού του προβλήματος, για Android, είναι η επιβολή κρυπτογράφησης στις τοπικές αρχειοθήκες του κινητού με χρήση της μεθόδου "setStorageEncryption" ή να κάνει χρήση της βιβλιοθήκης "javax.crypto". Για το iOS υπάρχει επίσης βιβλιοθήκη κρυπτογράφησης δεδομένων, ονομάζεται "CommonCrypto".

M3: Πέρα από την διαδικασία αυθεντικοποίησης, οι εφαρμογές κινητών συσκευών δεν χρησιμοποιούν κρυπτογράφηση για τα υπόλοιπα δεδομένα που μεταφέρονται μέσα στο δίκτυο. Γενικές τεχνικές για την αποφυγή αυτής της τρωτότητας είναι, η ομάδα ανάπτυξης να θεωρίσει ότι το επίπεδο δικτύου είναι εκτεθειμένο και μπορούν να υποκλαπούν πληροφορίες. Με την χρήση SSL/TLS για την μετάδοση ευαίσθητων πληροφοριών, token των συνεδριών ή άλλων πληροφοριών προς τους διακομιστές της εφαρμογής, θα μπορούσε επίσης να αποφευχθεί αυτό το πρόβλημα. Φυσικά θα πρέπει να βγει και ο παραπανίσιος αχρησιμοποίητος κώδικας, ο οποίος θα μπορούσε να είναι επισφαλές για την σύνδεση.

M4: Ακούσια διαρροή των δεδομένων συμβαίνει όταν ένας προγραμματιστής τοποθετεί λάθος ευαίσθητες πληροφορίες ή δεδομένα σε μια τοποθεσία στην κινητή συσκευή που είναι εύκολα προσβάσιμο από άλλες εφαρμογές. Είναι σημαντικό να γίνει ένα μοντέλο απειλής για την εφαρμογή, με βάση τα παρακάτω:

- Προσωρινή αποθήκευση URL (Τόσο αιτήματος και απόκρισης)
- Προσωρινή αποθήκευση πατημένου κουμπιού
- Προσωρινή αποθήκευση buffer για Αντιγραφή / Επικόλληση
- Παρασκήνιο της εφαρμογή
- Logging
- Αποθήκευση δεδομένων HTML5
- Αντικείμενα Browser cookie
- Analytics αποστέλλονται σε 3ους

Ακόμη, είναι σημαντικό, να γίνει κατανοητό το, τι κάνει από μόνο του το λογισμικό και να εφαρμοστούν στοιχεία ασφαλείας εκεί που πρέπει.

M5: Κακά ή εκλιπών συστήματα πιστοποίησης αφήνουν έναν επιτιθέμενο να εκτελέσει κώδικα ανενόχλητος και κατά κύριο λόγο, ανώνυμα. Η μορφή της πλατφόρμας για κινητά τηλέφωνα, ενθαρρύνει τον χρήστη να χρησιμοποιεί μικρούς κωδικούς, όπως για παράδειγμα PIN 4ων ψηφίων. Οι προγραμματιστές θα πρέπει να υποθέσουν ότι, όλες οι διαδικασίες αυθεντικοποίησης μπορούν να παραβιαστούν από κακόβουλους χρήστες. Αυτές οι διαδικασίες θα πρέπει να ελεγχθούν και να αλλαχθούν όπου είναι απαραίτητο. Λόγο των απαιτήσεων για χρήση και εκτός δικτύου, θα πρέπει να γίνονται και περισσότεροι έλεγχοι ασφαλείας και μέσα στην εφαρμογή.

M6: Η χρήση όχι αρκετά δυνατών αλγορίθμων κρυπτογράφησης, θα πρέπει να αποφευχθεί, διότι ο επιτιθέμενος θα μπορέσει να επαναφέρει τα δεδομένα στην αρχική τους κατάσταση και να κλέψει. Μερικοί από αυτούς τους αλγορίθμους, που έχει αποδειχθεί ότι έχουν σημαντικές αδυναμίες, είναι: RC2, MD4, MD5, SHA1. Δεν υπάρχει εύκολος τρόπος για την αποφυγή αυτής της τρωτότητας. Καλό θα ήταν οι προγραμματιστές να χρησιμοποιήσουν δικούς τους αλγορίθμους κρυπτογράφησης ή πρωτόκολλα.

M7:Client-side injection είναι το αποτέλεσμα κατά την εκτέλεση κακόβουλου κώδικα για την κινητή συσκευή μέσω μιας εφαρμογής για κινητά. Σε γενικές γραμμές, μπορούν να μπουν περισσότεροι έλεγχοι όπου υπάρχει είσοδος και έξοδος δεδομένων. Πιο συγκεκριμένα για Android, όταν υπάρχουν δυναμικά ερωτήματα ή φορείς παροχής περιεχομένου, πρέπει η ομάδα ανάπτυξης να βεβαιωθεί ότι χρησιμοποιούνται παραμετροποιημένα ερωτήματα. Επίσης να επαληθεύει όλα τα δεδομένα μέσω Φίλτρων για όλες τις δραστηριότητες (οθόνες).

M8: Οι προγραμματιστές χρησιμοποιούν γενικά κρυφά πεδία και τιμές ή κάποια κρυμμένη λειτουργικότητα για να διακρίνει τους χρήστες υψηλότερου επίπεδου από τους χρήστες χαμηλότερου επίπεδου. Ο επιτιθέμενος μπορεί να υποκλέψει και να τροποποιήσει αυτές τις ευαίσθητες παραμέτρους. Για την πλατφόρμα του Android δεν υπάρχουν ακόμη κάποια μέτρα ασφαλείας ως προς την συγκεκριμένη απειλή. Γενικότερα, οι προγραμματιστές θα πρέπει να επαληθεύουν τα δεδομένα και όταν κληθεί κάποιο URL, να μην το ανοίγουν χωρίς πρώτα να εξετάσουν εάν είναι έμπιστο.
M9: Προκειμένου να διευκολυνθεί μια καταστασιακή συναλλαγή μεταξύ του χρήστη και ενός backend διακομιστή κινητής εφαρμογής, τα κινητά τηλέφωνα χρησιμοποιούν κλειδιά συνόδου να διατηρηθεί η συνεδρία μέσω των πρωτοκόλλων όπως HTTP ή SOAP. Αφού γίνει η αυθεντικοποίηση του χρήστη, αποθηκεύεται ένα cookie, στο κινητό, με τα στοιχεία του. Εκείνες τις πληροφορίες έχει στόχο ο επιτιθέμενος. Για να γίνει σωστή διαχείρηση των συνεδριών, θα πρέπει να υπάρχει απόλυτη βεβαιότητα από τους προγραμματιστές, ότι η εφαρμογή μπορεί να δημιουργεί, να συντηρεί και να καταστρέφει τα tokens κατά την διάρκεια του κύκλου ζωής της εφαρμογής, στην συνεδρία του χρήστη.
M10: Η έλλειψη δυαδικής προστασίας μέσα σε μια εφαρμογή για κινητά, εκθέτει την εφαρμογή και τον ιδιοκτήτης της σε μια μεγάλη ποικιλία τεχνικών και επιχειρηματικών κινδύνων. Είναι δύσκολο να εντοπιστεί ο δράστης που έχει χρησιμοποιήσει αντίστροφη μηχανική σε μία εφαρμογή. Συνήθως ο κάτοχος της αρχικής εφαρμογής, το μαθαίνει όταν ένας κλώνος της εφαρμογής τους, εμφανισθεί σε κάποιο app store. Για να μπορέσει κάποιος να αποφύγει την κατάσταση, στην οποία η εφαρμογή του έχει αποσυμπιληστεί, θα πρέπει να πάρει μέτρα για τα ακόλουθα στοιχεία:

- Έλεγχο Ανίχνευσης Jailbreak
- Αθροίσματα Ελέγχου
- Έλεγχο Βεβαίωσης Πιστοποιητικου
- Έλεγχο Ανίχνευσης Debugger

Ακόμη θα πρέπει να ελεγχθεί εάν το κινητό είναι σε κατάσταση Root. Για να γίνει αυτό, θα πρέπει να γίνει έλεγχος στα αρχεία πόρων του συστήματος, εάν υπάρχει υπογραφή γνήσιας έκδοσης λογισμικού, δηλαδή developer build ή εάν είναι ελεύθερο, custon ROM.

4. Επιθέσεις Injection και Προστασία

Η συγκεκριμένη εργασία επικεντρώνεται στον τύπο επίθεσης SQL injection. Στο κεφάλαιο αυτό θα δούμε πώς γίνεται μια τέτοια επίθεση και με ποιους τρόπους θα μπορούσε κάποιος θα την αποφύγει και να την αποτρέψει.

4.1 Θεωρία των επιθέσεων με Injection

Ο πιο αποτελεσματικός τρόπος για να αποφύγει κάποιος μια επίθεση Injection ή να την αποτρέψει, είναι να ορίσει μία θεωρία για αυτές. Μία επίθεση SQL injection αποτελείται από την εισαγωγή ή "ένεση" ενός ερωτήματος SQL μέσω δεδομένων εισόδου από τον πελάτη στην εφαρμογή. Μια επιτυχημένη ένεση SQL, ο επιτιθέμενος μπορεί να διαβάσει τα ευαίσθητα δεδομένα από τη βάση δεδομένων, να τροποποιήσει τα δεδομένα της βάσης δεδομένων (Εισαγωγή / Ενημέρωση / Διαγραφή), διενεργήσει πράξεις διαχείρισης της βάσης δεδομένων, όπως διακοπή λειτουργίας του DBMS (Database Management System), την ανάκτηση του περιεχομένου ενός συγκεκριμένου παρόντος φακέλου στο αρχείο DBMS

σύστημα και σε ορισμένες περιπτώσεις να δώσει εντολές στο λειτουργικό σύστημα. Σύμφωνα με την λίστα Common Weakness Enumeration (CWE) (8) της εταιρίας MITRE, η SQL injection είναι η νούμερο ένα ευπάθεια από το 2011. Άλλα παραδείγματα επιθέσεων με injection είναι το Cross-Site Scripting, εν συντομεία XSS, ένας τύπος HTML injection και OS Command Injection. Όλες αυτές οι επιθέσεις έχουν κοινό στόχο, ο κακόβουλος χρήστης να μπορέσει να βρεί ένα ευπαθές κομμάτι της εφαρμογής ή προγράμματος και μέσο αυτού να μπορέσει να δει, να κλέψει ή να διαγράψει ευαίσθητα δεδομένα.

4.2 Τρωτά σημεία και μη εξουσιοδοτημένες είσοδοι

Δεν υπάρχει κανόνας ο οποίος να λέει πότε πρέπει να βάλει λίγο περισσότερη ασφάλεια σε μία εφαρμογή ένας προγραμματιστής και πότε όχι. Όλα είναι στην κρίση του και στο πώς έχει σκεφτεί αυτός την εφαρμογή. Παρακάτω δίνεται ένα μη ασφαλές ερώτημα SQL.

```
SELECT * FROM user WHERE USERNAME = '$_POST[username]' AND password
= '$_POST[pass]'
```

Σχήμα 2: Κώδικας τρωτού ερωτήματος SQL

Ο παραπάνω κώδικα είναι μια επίδειξη από ερώτημα αυθεντικοποίησης χρήστη στην SQL. Επιλέγει όλα τα δεδομένα που έχουν σχέση με, το όνομα χρήστη (username) και τον κωδικό πρόσβασης (pass) που θα του δώσει ο χρήστης της εφαρμογής. Στην περίπτωση που είναι λανθασμένα τα δεδομένα δεν επιστρέφει τίποτα, αλλά εάν κάποιος εισάγει ένα όνομα χρήστη και το παρακάτω αλφαριθμητικό για κωδικό:

```
'                    OR                    '1'='1
```

το σύστημα θα του επιτρέψει την είσοδο. Εδώ πρέπει να προσέξει κάποιος ότι στον τελευταίο άσσο, δεν χρειάζεται να κλείσουν τα εισαγωγικά, διότι το αρχικό ερώτημα κλείνει στο τέλος με εισαγωγικό, οπότε αν μπει θα είναι διπλό εισαγωγικό και θα εμφανιστεί σαν συντακτικό λάθος στην βάση δεδομένων. Το παραπάνω ερώτημα, μετά την εισαγωγή του αλφαριθμητικού, θα είναι μοιάζει κάπως έτσι:

```
SELECT * FROM user WHERE USERNAME = '$_POST[username]' AND password
= '' OR '1'='1'
```

Σχήμα 3: Κώδικας ερωτήματος SQL μετά απο injection

Με πιο απλά λόγια, η λογική μεταβλητη password, θα είναι πάντα αληθής, διότι το 1=1 είναι πάντα αληθής και δεν μπορεί να αλλάξει. Έτσι το μόνο που πρέπει να κάνει ένας κυβερνο-εγκληματίας είναι να βρει το όνομα χρήστη που χρησιμοποιεί το υποψήφιο θήμα, ώστε να του κάνει κακό ή να του πάρει ευαίσθητες πληροφορίες, που αργότερα μπορεί να χρησιμοποιήσει προς όφελος του. Είτε αυτό είναι οικονομικό είτε όχι. Επιπλέον σε μία βάση δεδομένων η πρώτη γραμμή του πίνακα που χρησιμοποιείτε για να γίνει αυθεντικοποίηση των δεδομένων πρόσβασης ενός χρήστη, είναι συνήθως ο κωδικός του Διαχειριστή. Οπότε εάν ο χρήστης βάλει δύο φορές των κώδικα '1'='1', υπάρχει μεγάλη πιθανότητα να συνδεθεί ώς διαχειριστής της εφαρμογής, με αποτέλεσμα να έχει, αυτός, μεγαλύτερη δύναμη και ευελιξία να συλλέξει πληροφορίες ή να τις διαγράψει, χωρίς κανείς να τον καταλάβει. Στις

μέρες μας, οι περισσότερες εφαρμογές χρησιμοποιούν σαν όνομα χρήστη, το e-mail του χρήστη που τις χρησιμοποιεί. Έτσι μπορούν να έχουν μοναδικούς χρήστες χωρίς να ανησυχούν για διπλούς λογαριασμούς. Το παραπάνω αλφαριθμητικό δεν είναι ο μόνος τρόπος για να έχει κάποιος εξωτερικός παράγοντας, μη εξουσιοδοτημένη είσοδο στον λογαριασμό κάποιου άλλου, διότι από μεταφραστή σε μεταφραστή έχει διαφορά το συντακτικό. Επιπλέον κάποιος μπορεί να χρησιμοποιήσει και το "--" στο τέλος του αλφαριθμητικού, για να κάνει σχόλια τυχόν έξτρα παραμέτρους που έχει το ερώτημα SQL (12).

Επίθεση injection δεν γίνεται μόνο μέσω κάποιας εφαρμογής. Μπορεί να γίνει και μέσω ενός απλού φυλλομετρητή ιστού (web browser). Εύκολα μπορεί κάποιος να βρει μια σελίδα που να ενδέχεται να είναι ευάλωτη σε επίθεση injection, το μόνο που έχει να προσέξει είναι η γραμμή διεύθυνσης, το URL να έχει κάποια είσοδο για μεταβλητές, όπως "id=" ή "username=". Ένα παράδειγμα τέτοιου URL είναι:

```
https://www.ce.teiep.gr/index.php?id=1&language=gr
```

Σχήμα 4: Παράδειγμα μη ασφαλούς URL

Εάν ο κακόβουλος χρήστης δει ότι η σελίδα μπορεί να παραβιαστεί, αλλά δεν έχει λογαριασμούς χρηστών, μπορεί να βάλει μέσω της γραμμής διευθύνσεων του φυλλομετρητή ιστού, ένα πεδίο UNION, με το οποίο θα μπορέσει να κάνει δευτερεύων ερώτημα στην βάση και να περάσει μεταβλητές στο ερώτημα, τις οποίες δεν είχε το αρχικό ερώτημα.

```
http://www.example.com/product.php?id=10 UNION SELECT 1 FROM DUAL--
```

Σχήμα 5: Παράδειγμα χρήσης Union σε URL

Τέλος στην περίπτωση που η βάση χτυπάει συνέχεια συντακτικό λάθος θα μπορούσε ο κακόβουλος χρήστης, με χρήση του UNION να τρέξει ερώτημα με την συνάρτηση version() και να δεί ποια ακριβώς έκδοση και ποιον μεταφραστή χρησιμοποιεί η βάση, ώστε να κάνει την επίθεση του πιο σίγουρη για επιτυχία.

4.3 Αποτροπή επιθέσεων Injection

Όπως είναι φυσικό, οι άνθρωποι που δουλεύουν στην ασφάλεια πληροφοριών για εταιρείες πληροφορικής, και όχι μόνο, έβαλαν τα δυνατά τους για να σταματήσουν αυτού του είδους τις επιθέσεις και να προσπαθήσουν να προστατέψουν τις πληροφορίες τις κάθε επιχείρησης.

4.3.1 Define and Bind

Στην αρχή, ίσως κανείς να μην προέβλεψε ότι θα μπορούσε με ένα απλό κόλπο να έχει πρόσβαση σε λογαριασμούς διαχειριστών ή ακόμη και σε λογαριασμούς πελατών κάθε λογής. Σαν καινούργιος τομέας για την ασφάλεια πληροφοριών, δεν υπήρχαν αρκετοί τρόποι, για την αποτροπή τέτοιων επιθέσεων, όπως δεν υπήρχε και αρκετή γνώση για το πώς ή το πού μπορεί να γίνει τέτοια επίθεση. Υπήρχαν όμως αρκετές γλώσσες προγραμματισμού, από τις οποίες θα μπορούσαν να πάρουν σαν παράδειγμα για μερικούς τρόπους αποτροπής επιθέσεων εισαγωγής κώδικα.

Έτσι για την ασφάλεια των αρχικών ερωτημάτων στις βάσεις δεδομένων, οι μεταβλητές δηλώνονταν στην αρχή του προγράμματος, μαζί με τον τύπο τους, ώστε

αργότερα όταν του ανατεθεί μία τιμή, να μην γίνει λανθασμένη χρήση από τον μεταφραστή της βάσης δεδομένων. Αυτό δεν είχε πάντα το επιθυμητό αποτέλεσμα, διότι υπήρχαν περιπτώσεις όπου και πάλι περνούσε ο κακόβουλος κώδικας. Αυτό, επίσης, σήμαινε ότι ο προγραμματιστής θα είχε ένα πεπερασμένο αριθμό μεταβλητών που θα μπορούσε να δουλέψει, διότι όσο περισσότερες μεταβλητές χρησιμοποιούσε, τόσο περισσότερη μνήμη και υπολογιστική ισχύ δέσμευε από τον υπολογιστή. Αυτό έκανε το πρόγραμμα βαρύ για τους τότε υπολογιστές, με αποτέλεσμα να τους κάνει ακόμη πιο αργούς.

Αργότερα δημιουργήθηκαν και συναρτήσεις, οι οποίες "έδεναν" μεταβλητές σε συγκεκριμένο τύπο. Με πιο απλά λόγια, δεν ήταν αναγκαστικό για τον προγραμματιστή, να δηλώσει τον τύπο της μεταβλητής στην αρχή του προγράμματος, αλλά μπορούσε αργότερα και όταν ήταν αναγκαίο να δημιουργήσει την μεταβλητή να της δώσει σε έναν τύπο, όπως για παράδειγμα αλφαριθμητικό (sting), ακέραιο (integer) ή πραγματικό (float). Και σε αυτή την περίπτωση υπήρχε η πιθανότητα να περαστούν κομμάτια επικίνδυνου κώδικα υπό ορισμένες συνθήκες, αλλά το πρόγραμμα θα ήταν πιο ελαφρύ και πιο γρήγορο σε σχέση με την προηγούμενη μέθοδο που χρησιμοποιούσαν.

4.3.2 PHP

Με την πάροδο των χρόνων, την ανάπτυξη της τεχνολογίας και τον ερχομό του διαδικτύου στο απλό καθημερινό σπίτι και αφού πλέον μπορούσε ο καθένας να περιηγηθεί στο διαδίκτυο χωρίς να πληρώνει μια περιουσία στις εταιρείες τηλεφωνίας, ήρθε και η ανάπτυξη διάφορων γλωσσών προγραμματισμού, οι οποίες είχαν σαν μοναδικό στόχο την ανάπτυξη ιστοσελίδων και την διακόσμησή τους. Εννοείτε ότι στην αρχή δεν είχαν την ευχέρεια για να χρησιμοποιήσουν πολύπλοκες τεχνικές ασφαλείας για δεδομένα. Μία από αυτές της γλώσσες ήταν η PHP. Κατασκευασμένη με χρήση της γλώσσας C και αρκετή ομοιότητα με την γλώσσα Perl στην αρχή και φυσικά με πιο περιορισμένες λειτουργίες απ' ότι έχει τώρα. Η κυριότερη έκδοση της PHP ήταν η PHP 3, στην οποία η γλώσσα έγινε αντικειμενοστραφής (15). Το 2004 παρουσιάστηκε, μετά από πολλές δοκιμές και αρκετή ανάπτυξη, η PHP 5, η οποία έχει όλες τι λειτουργίες που γνωρίζουμε.

Από μόνη της η PHP δεν θα μπορούσε να συνδεθεί με μία βάση δεδομένων, και να αλλάξει ή να προσθέσει ή ακόμη και να σβήσει δεδομένα από αυτήν. Έτσι γίνεται ένας συνδυασμός την γλώσσας PHP με την γλώσσα SQL ώστε να γίνουν εφικτά. Για την δήλωση των μεταβλητών, η γλώσσα PHP, απαιτεί να μπει το σήμα "$" μπροστά από το όνομα της μεταβλητής, όμως δεν απαιτεί να γίνει καμία δήλωση ως προς τον τύπο των μεταβλητών που χρησιμοποιούνται σε κάποιο πρόγραμμα. Ο τύπος αλλάζει δυναμικά, και όταν ή όπου χρειαστεί. Με άλλα λόγια, μία μεταβλητή με περιεχόμενο τον αριθμό 1, μπορεί να χρησιμοποιηθεί ως αλφαριθμητικό αλλά και ως ακέραιος αριθμός.

Αυτό φυσικά άφηνε αρκετές τρύπες για εισαγωγές μη ασφαλούς κώδικα και αρκετά δεδομένα χρηστών είχαν κλαπεί. Έτσι στην PHP 5 δημιουργήθηκαν συναρτήσεις τύπου strval(), που επιστρέφει το αλφαριθμητικό της μεταβλητής που είναι μέσα στις παρενθέσεις. Εδώ αξίζει να σημειωθεί ότι, η συνάρτηση intval(), επιστρέφει ακέραιο μέρος της μεταβλητής, η οποία είναι μέσα στις παρενθέσεις, όμως το σύστημα του αριθμού που θα επιστραφεί είναι ανάλογο με το πώς είναι γραμμένη η μεταβλητή εισαγωγής. Στην περίπτωση που είναι κάποιο αλφαριθμητικό, με αρχή 0χ, τότε η συνάρτηση θα επιστρέψει τον ακέραιο αριθμό με βάση το 16δικό σύστημα. Εάν ξεκινάει μόνο με 0, τότε επιστρέφει τον ακέραιο αριθμό με βάση το 8δικό σύστημα. Σε διαφορετική περίπτωση επιστρέφει με βάση στο 10δικό σύστημα

(16). Υπό ορισμένες συνθήκες ακόμη και η χρήση αυτών των ειδικών συναρτήσεων, δεν μπορούσε να σταματήσει τις επιθέσεις εισαγωγής κώδικα.

Ένας αρκετά ασφαλές κωδικός χρήστη έχει τουλάχιστον 16 ψηφία ή περισσότερα, τα οποία αποτελούνται από γράμματα ή από αριθμούς ή από ειδικούς χαρακτήρες. Ακριβώς επειδή μπορεί να αποτελεστεί από ειδικούς χαρακτήρες, δεν υπάρχει τρόπος να αποφευχθούν τελείως οι επιθέσεις εισαγωγής κώδικα, μπορούν όμως να περιοριστούν μόνο σε πεδία όπως ένας κωδικός. Στους πρώτους γνωστούς κωδικούς, η εισαγωγή ειδικών χαρακτήρων δεν ήταν εφικτή. Για αυτό τον λόγο δημιουργήθηκε η συνάρτηση mysql_real_escape_string. Αυτή η συνάρτηση παίρνει σαν είσοδο μία μεταβλητή και αφού την μετατρέψει σε αλφαριθμητικό, αφαιρεί όλους τους ειδικούς χαρακτήρες και επιστρέφει το αλφαριθμητικό, χωρίς αυτούς. Με τον ερχομό της PHP 5, όλοι και λιγότεροι προγραμματιστές χρησιμοποιούσαν αυτή την συνάρτηση, διότι οι ανάγκες ασφαλείας είχαν αυξηθεί αρκετά. Πολλοί χρήστες που είχαν ειδικούς χαρακτήρες στους κωδικούς τους ήταν ανίκανοι να συνδεθούν στον λογαριασμό τους, λόγο της συνάρτησης που εισήγαγε λάθος αλφαριθμητικό στην βάση, με αποτέλεσμα να γίνει λάθος το ερώτημα και μην υπάρχουν τα επιθυμητά αποτελέσματα.

Για να καλυφθεί αυτό το πρόβλημα, οι προγραμματιστές δημιούργησαν την PDO, είναι το αρκτικόλεξο για τις λέξεις PHP Data Objects. Η PDO, είναι μία προέκταση της κλασσικής PHP και περιέχει διάφορους οδηγούς και αρκετές συναρτήσεις για πολλές βάσης δεδομένων που χρησιμοποιούνται, όμως δεν μπορεί από μόνη της να κάνει τα πάντα σε μία βάση δεδομένων. Επίσης μπορεί να χρησιμοποιήσει τις ίδιες συναρτήσεις για να μαζέψει δεδομένα από όλες τις βάσεις δεδομένων. Με την PDO, χρησιμοποιεί έτοιμα ερωτήματα (prepared statement) για τις βάσεις δεδομένων. Αυτό βοηθάει τόσο στην ασφάλεια όσο και στην ταχύτητα εκτέλεσης του προγράμματος. Στην ταχύτητα βοηθάει, διότι ένα έτοιμο ερώτημα **(/**/)** φορτώνετε μία φορά στην μνήμη και μπορεί να εκτελεστεί όσες φορές χρειαστεί, χωρίς να επιβαρύνει το σύστημα. Το αντίστοιχο απλό ερώτημα που χρησιμοποιούνταν τα περασμένα χρόνια, φορτώνονταν στην μνήμη κάθε φορά που ήταν να τρέξει.

Με PDO δύο συνεχόμενα ερωτήματα θα γίνουν:

```
$stmt = $con>prepare("INSERT INTO EMPLOYERS(name, id) VALUES
(?, ?)");
$stmt->bindParam(1, $name);
$stmt->bindParam(2, $id);

//εισαγωγή 1ης γραμμής
$name = 'one';
$id = 1;
$stmt->execute();
// εισαγωγή δεύτερης σειράς, με διαφορετικές τιμές.
$name = 'two';
$id = 2;
$stmt->execute();
```

Σχήμα 6α: Σύγκρηση mysql query και PDO prepared statement

Ενώ με απλό ερώτημα της mysql θα δείχνει κάπως έτσι:

```
//εισαγωγή 1ης γραμμής
$name='one';
```

```
$id=1;
mysql_query("INSERT INTO EMPLOYERS(name, id) VALUES ($name,
$id)");

//εισαγωγή 2ης γραμμής
$name='two';
$id=2;
mysql_query("INSERT INTO EMPLOYERS(name, id) VALUES ($name,
$id)");
```

Σχήμα 6β: Σύγκρηση mysql query και PDO prepared statement

Επίσης για να αποφευχθεί η πιθανότητα για επίθεση εισαγωγής κώδικα, από κακόβουλο χρήστη, η PDO έχει συνάρτηση bindValue, που είναι δένει μία μεταβλητή με έναν συγκεκριμένο τύπο και μία συγκεκριμένη θέση στο ερώτημα που πάει να γίνει στην βάση δεδομένων.

```
$stmt= $con->prepare("SELECT * FROM user WHERE USERNAME =
:username AND password = :pwd");
$stmt->bindValue(':username', $username, PDO::PARAM_STR);
$stmt->bindValue(':pwd', $pwd, PDO::PARAM_STR);
```

Σχήμα 7: PDO bindValue

Σε αντίθεση με την συνάρτηση τις άλλες συναρτήσεις, η bindvalue δεν βγάζει κανένα χαρακτήρα και επιστρέφει την τιμή σαν τύπο που όρισε ο προγραμματιστής. Αν, για παράδειγμα, γίνει εισαγωγή του μη ασφαλούς κώδικα, που έγινε παράδειγμα πιο πάνω, για μη ορθή είσοδο χρήστη, αντί να περαστεί κενό και μια Boolean η οποία θα είναι πάντα αληθής, θα περαστεί όλο αυτό σαν αλφαριθμητικό, το οποίο θα είναι και λάθος, οπότε θα αποτραπεί η είσοδος στον λάθος χρήστη. Αυτή η συνάρτηση δεν έχει μόνο σκοπό να επιστρέφει αλφαριθμητικά, μπορεί ο οποιοσδήποτε προγραμματιστής να την χρησιμοποιήσει για να “δέσει” ακέραιους και πραγματικούς αριθμούς ή ακόμη και μεγάλα αντικείμενα πολλά ακόμη (19).

5. Κρυπτογραφικά Εργαλεία

Η κρυπτογραφία αναφέρεται στην υλοποίηση μεθόδων τροποποίησης των μεταδιδομένων πληροφοριών, έτσι ώστε να γίνονται κατανοητά μόνο από τον προβλεπόμενο παραλήπτη ή παραλήπτες. Είναι μια διαδικασία που μπορεί να εκτελεστεί τόσο σε hardware όσο και σε software. Η ενσωμάτωση των μεθόδων της κρυπτογραφίας σε hardware επιταχύνει σε μεγάλο βαθμό την διεκπεραίωση της. Επίσης, οι χρήστες δεν γνωρίζουν, ούτε καν αντιλαμβάνονται την παρουσία της και πραγματοποιούν ανενόχλητοι τις εργασίες τους. Το γεγονός ότι ο χρήστης δεν ανακατεύεται καθόλου στις διαδικασίες της κρυπτογραφίας, αυξάνει την αποτελεσματικότητα του εργαλείου στην παρεχόμενη ασφάλεια. (32). Στην εφαρμογή της συγκεκριμένης εργασίας, έχει γίνει η χρήση του κρυπτογραφικού αλγορίθμου MD5 για την αποθήκευση του αλφαριθμητικού κλειδιού στην βάση δεδομένων, με στόχο τόσο την ασφάλεια κατά της αυθεντικοποίηση του χρήστη ενάντια σε επιθέσεις SQL, όσο και για την ακεραιότητα των δεδομένων που λαμβάνονται από το σύστημα.

5.1 Απλός κατακερματισμός

Κατακερματισμός είναι η μετατροπή μιας σειράς χαρακτήρων σε μια, συνήθως, μικρότερου μήκους σταθερή τιμή - κλειδί που αντιπροσωπεύει τους αρχικούς χαρακτήρες. Ο κατακερματισμός χρησιμοποιείται για να επιδειχθούν και να ανακτηθούν στοιχεία σε μια βάση δεδομένων, επειδή είναι πιο γρήγορα να βρεθεί μία εγγραφή στοιχείου χρησιμοποιώντας το μικρότερο κατακερματισμένο κλειδί, από ό,τι να βρεθεί χρησιμοποιώντας την αρχική τιμή. Χρησιμοποιείται επίσης σε πολλούς αλγόριθμους κρυπτογράφησης. Τέτοιου είδους συναρτήσεις ονομάζονται μονομερής συναρτήσεις, διότι είναι εύκολο να δημιουργήσουμε πληροφορία ή να τσεκ-άρουμε δεδομένα, χρησιμοποιώντας μία συνάρτηση κατακερματισμού, αλλά είναι δύσκολο να μεταβούμε από την κατακερματισμένη πληροφορία στην αρχική. Επίσης χρησιμοποιούνται για την δημιουργία πινάκων, που αργότερα θα χρησιμοποιηθούν ως μέσω επαληθεύσεις πληροφοριών και για την κρυπτογράφηση δεδομένων, που θα δούμε στο επόμενο υπό κεφάλαιο. Στην επιστήμη της πληροφορικής και ιδιαίτερα στον τομέα του κατακερματισμού, υπάρχουν συγκρούσεις (21). Με τον όρο συγκρούσεις, εννοείτε το ίδιο αποτέλεσμα κατακερματισμού, για δύο διαφορετικές πληροφορίες. Αυτές οι συγκρούσεις συναντώνται περισσότερο, όταν συγκρίνουμε δύο αρκετά μεγάλα σύνολα πληροφοριών, όπως για παράδειγμα δύο αρχεία με ονόματα πελατών, και το κατακερματισμένο κλειδί είναι μικρό σε έκταση, για παράδειγμα 32bit. Οι συγκρούσεις είναι αναπόφευκτες ότι και να κάνει ο προγραμματιστής, το πώς τις χειρίζεται όμως είναι αυτό που μετράει. Υπάρχουν συναρτήσεις κατακερματισμού, οι οποίες αυξάνουν την πιθανότητα για σύγκρουση, οι οποίες χρησιμοποιούνται για την ταυτοποίηση δακτυλικών αποτυπωμάτων ή αρχείων ήχου και άλλες συναρτήσεις οι οποίες μειώνουν αρκετά την πιθανότητα των συγκρούσεων, τα λεγόμενα Checksums ή αθροίσματα ελέγχου (20).

5.2 Checksums

Τα αθροίσματα ελέγχου είναι μικρές ποσότητες πληροφοριών με στόχο για την επαλήθευση πληροφοριών, για τυχών λάθη που δημιουργήθηκαν κατά την μετάδοση. Αρκετοί server στον κόσμο χρησιμοποιούν αυτά τα αθροίσματα ελέγχου, για να επαληθεύσουν ότι οι πληροφορίες έχουν φτάσει επιτυχώς στον στόχο και είναι όλες σωστές. Ένας καλός κώδικας που βγάζει τέτοια αθροίσματα, θα πρέπει να μπορεί να βγάλει τελείως διαφορετικές τιμές ακόμη και αν η είσοδος είναι σχεδόν ίδια (22) (23).

5.3 Κρυπτογραφικός Κατακερματισμός

Εδώ πρέπει να τονιστεί ότι ο κρυπτογραφικός κατακερματισμός δεν έχει καμία σχέσει με την κρυπτογράφηση των δεδομένων. Είναι φυσικό να μπερδευτεί κάποιος με τον τίτλο που κατέχει αυτή η μέθοδος. Οι συναρτήσεις κρυπτογραφίας, μετατρέπουν τα δεδομένα, για παράδειγμα το κείμενο “Hello world” σε κείμενο με διαφορετικά γράμματα, όμως τον ίδιο αριθμό γραμμάτων για κάθε λέξη. Επίσης κάθε κρυπτογραφικός αλγόριθμος έχει ένα κλειδί, το οποίο το έχουν και οι δύο πλευρές που αποστέλλουν μηνύματα. Έτσι γίνεται κατανοητό το μήνυμα μόνο στα άτομα που πρέπει να το διαβάσουν. Αυτό είναι ένα κλασσικό

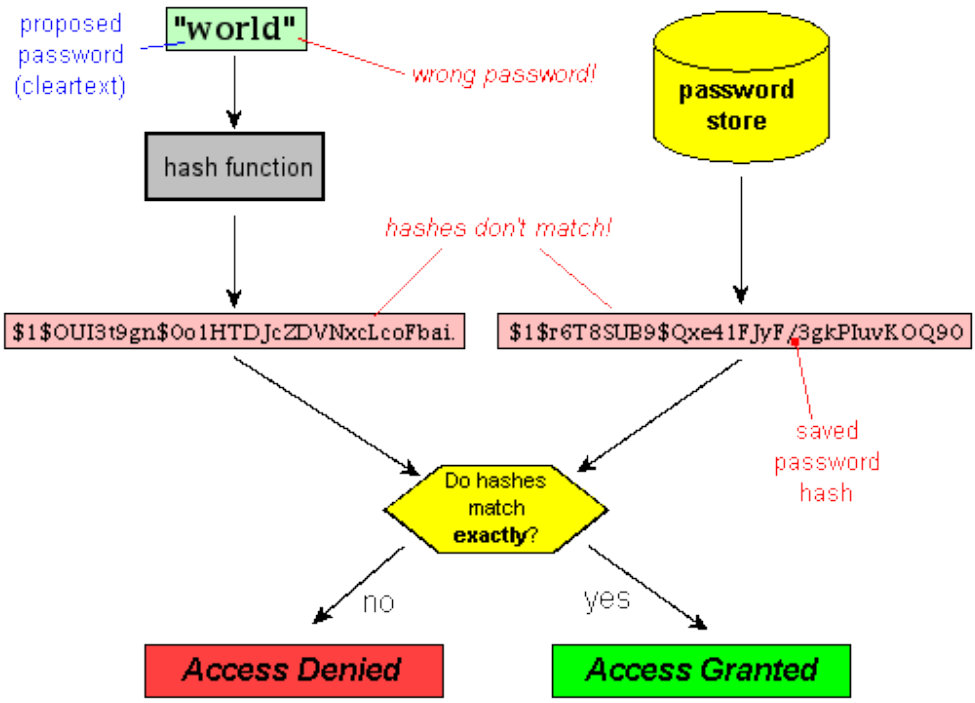
παράδειγμα αμφίδρομων αλγορίθμων. Μπορούν να αναστραφούν τα αποτελέσματα, μόνο με το σωστό κλειδί. Οι κρυπτογραφικοί αλγόριθμοι είναι μονόδρομοι αλγόριθμοι, όπως ακριβώς είναι και ο απλός κατακερματισμός, χρησιμοποιούνται για να δημιουργηθεί μία τιμή, που θα είναι τελείως μη αντιστρέψιμη και θα βοηθήσει στην επαλήθευση των δεδομένων και κωδικών. Πιο ειδικά αυτού του είδους οι αλγόριθμοι, χρησιμοποιούνται για να συνοψίσουν μεγάλες ποσότητες δεδομένων, σε σχετικά μικρούς αριθμούς, για την βοήθεια της επαλήθευσης (24). Ακόμη είναι σημαντικό να γνωρίζει κανείς ότι, το μήκος του κλειδιού παραμένει σταθερό, αναλόγως με το είδος του κρυπτογραφικού αλγορίθμου που χρησιμοποιήθηκε για την δημιουργία του, ακόμη και αν τα δεδομένα έχουν τεράστια διαφορά στο μέγεθός τους. Επίσης μπορεί δύο κλειδιά θα μοιάζουν αρκετά μεταξύ τους, όμως στην περίπτωση που δεν είναι ακριβώς όμοιοι, ακόμη και αν έχουν διαφορά ένα bit, υπάρχει η περίπτωση τα δεδομένα να είναι τελείως διαφορετικά μεταξύ τους (24).

Όσο αφορά τον κατακερματισμό κωδικών, είναι καλό να μην αφήνουμε τους κωδικούς σε κοινή θέα, ακόμη και μέσα σε βάση δεδομένων με τρόπο που να φαίνονται ακριβώς ποιοι είναι. Δεν είναι απόλυτα ασφαλές και μπορεί κάποιος κακόβουλος χρήστης να επιτεθεί στην βάση και να τους κλέψει, με αποτέλεσμα αργότερα να κάνει ζημιά. Όπως αναφέρθηκε και πιο πάνω, οι περισσότερες ληστείες δεδομένων γίνονται μέσα από μία επιχείρηση παρά από εξωτερικές επιθέσεις. Με την χρήση κρυπτογραφικών αλγόριθμων κατακερματισμού, μπορεί κάποιος να αποτρέψει τέτοια προσπάθεια. Δημιουργώντας ένα κατακερματισμένο κλειδί και αποθηκεύοντας αυτό αντί για τον κωδικό στην βάση, ακόμη και ένας μικρός κωδικός 6 χαρακτήρων, μόνο με πεζά γράμματα, μπορεί να γίνει πιο ασφαλής. Αυτό οφείλετε στο γεγονός ότι, ακόμη και για 6 χαρακτήρες θα δημιουργηθεί ένα μοναδικό κλειδί 30-40 χαρακτήρων, που μόνο ο κατακερματισμός της λέξης κλειδί θα είναι εφικτό να παράγει. Σαν παράδειγμα μπορούμε να δούμε το παρακάτω:

```
MD4          ("")          =          31d6cfe0d16ae931b73c59d7e0c089c0
MD4          ("a")         =          bde52cb31de33e46245e05fbdbd6fb24
MD4 ("abc") = a448017aaf21d8525fc10ae87aa6729d
```

Σχήμα 8: Παράδειγμα κατακερματισμού κωδικών.

Οι παραπάνω σχέσεις είναι ορισμένες στο αρχείο The MD4 Message Digest Algorithm (26). Αυτό με την σειρά του σημαίνει ότι υπάρχει μόνο μία λέξη από της άπειρες που υπάρχουν ανά τον κόσμο, που θα βγάλει το συγκεκριμένο αλφαριθμητικό και θα δώσει στον χρήστη πρόσβαση στην βάση.



Σχήμα 9: Παράδειγμα επαλήθευσης κλειδιού από την βάση (25).

Με την πάροδο των χρόνων, τα μοντέλα αλγορίθμων για κρυπτογραφικό κατακερματισμό, έχουν αλλάξει, αρκετά χρησιμοποιούνται όμως και τώρα, παρ' ότι θεωρούνται "σπασμένα" από ειδικούς.

MD4: Ο MD4, αρκτικόλεξο για τις λέξεις Message-Digest είναι μια κρυπτογραφική συνάρτηση κατακερματισμού που εφευρέθηκε και αναπτύχθηκε από τον Ronald Rivest το 1990 (26) (27). Το μήκος του κλειδιού που παρείχε στον χρήστη είναι 128 bits. Ήταν ο αλγόριθμος βάση για τις μετέπειτα γενιές κρυπτογραφικών αλγορίθμων όπως MD5 και SHA1. Το 1995 ο κατακερματισμός MD4, επλήγη σοβαρά από επίθεση συγκρούσεων, από τότε υπάρχουν αρκετές περιπτώσεις και δημοσιεύσεις και αναφορές για για τέτοιου είδους επιθέσεις. Το 1995 ο Hans Dobbertin, έκανε την πρώτη επιτυχημένη απόπειρα να σπάσει την αλγόριθμο, και του πήρε μόλις μερικά δευτερόλεπτα (28).

MD5: Ο αλγόριθμος MD5 είναι ένας ευρέως χρησιμοποιούμενος αλγόριθμος για κατακερματισμό παράγει ένα κλειδί κατακερματισμού μήκους 128-bit. Ο MD5 σχεδιάστηκε αρχικά για να χρησιμοποιηθεί ως μια συνάρτηση κρυπτογραφικού κατακερματισμού, αλλά βρέθηκε ότι είχε εκτεταμένα τρωτά σημεία. Μπορεί ακόμη να χρησιμοποιηθεί ως άθροισμα ελέγχου για την επαλήθευση της ακεραιότητας των δεδομένων, αλλά μόνο αν υπάρχει ακούσια περίπτωση καταστροφής ή διαφθοράς των δεδομένων. Δημιουργήθηκε από τον Ronald Rivest, μετά από φόβο ότι ο MD4 ήταν τρωτός σε επιθέσεις. Το 1996, βρέθηκε μία αδυναμία στον αλγόριθμο και από τότε οι κρυπτογράφοι, άρχισαν να συνιστούν την χρήση άλλων αλγορίθμων όπως ο SHA-1. Το 2004 μετά από επίθεση επίδειξης, αποδείχθηκε ότι ο αλγόριθμος δεν ήταν αρκετά ανθεκτικός σε συγκρούσεις (29).

SHA-1: Ο SHA-1, συντομογραφία του Secure Hash Algorithm 1, είναι ένας αλγόριθμος κατακερματισμού, δημιουργημένος απο την NSA (National Security Agency) και δημοσιευμένη απο το NIST (National Institute of Standards and Technology) των Ηνωμένων Πολιτειών της Αμερικής. Ο αλγόριθμος αυτός δημιουργεί ένα κλειδί μήκους 160bit. Συνήθως είναι σε 16αδικό σύστημα και έχει 40 χαρακτήρες. Ο SHA-1 δεν θεωρίτε ασφαλής έναντι σε

δυνατούς αντιπάλους. Το 2005, κρυπταναλυτές, βρήκαν επιθέσεις εναντίον του αλγορίθμου, οι οποίες υποδήλωναν ότι δεν είναι ασφαλής η χρήση του (30).

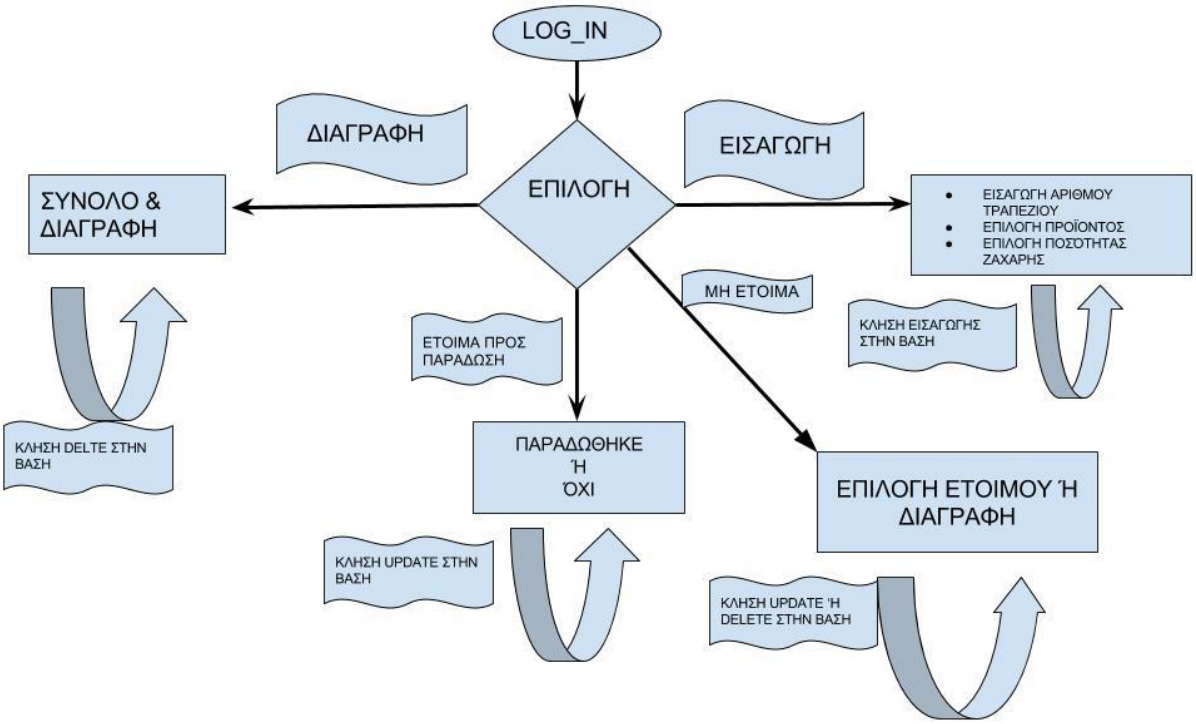
SHA-2: Ο SHA-2 περιλαμβάνει σημαντικές αλλαγές από τον προκάτοχό του, SHA-1. Η οικογένεια SHA-2 αποτελείται από έξι συναρτήσεις κατακερματισμού, που έχουν μήκος κλειδιού 224, 256, 384 ή 512 bits. Αυτές οι συναρτήσεις είναι γνωστές ως SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA -512 / 256.

6. Η Εφαρμογή

Η εφαρμογή προσομοιώνει ένα PoS, Point of Sale, μιας καφετέριας. Για να ξεκινήσει κάποιος να την χρησιμοποιεί, θα πρέπει να αυθεντικοποιήθει. Αφού αυθεντικοποιηθεί, θα μπορεί να δώσει παραγγελία στο κατάστημα. Σαν επιλογές υπάρχουν, να δώσει κάποιος παραγγελία, να δει το κάποιος τι παραγγελίες έχουν δοθεί, τι παραγγελίες έχουν ετοιμαστεί και είναι προς παράδοση και τι ποσό είναι να πληρώσει το κάθε τραπέζι. Πάνω σε αυτές τις επιλογές, θα επιδειχθεί το πώς μπορεί κάποιος να κάνει επίθεση SQL injection, για να αυθεντικοποιηθεί ως χρήστης της εφαρμογής και να διαγράψει τα δεδομένα της βάσης. Επίσης με την χρήση διάφορων script, θα επιδειχθεί το πώς θα γίνει η αποφυγή της επίθεσης και θα μείνουν τα δεδομένα ασφαλείς από τον κακόβουλο χρήστη.

Για την εφαρμογή που θα γίνει επίδειξη, χρησιμοποιήθηκε το Android Studio και το χρήση του SDK (Software Development Kit) 23. Σαν API χρησιμοποιήθηκε το API 18, που επιτρέπει την εφαρμογή να τρέξει σε λογισμικά android με έκδοση 4.0.3 και μετά. Επίσης έγινε η εισαγωγή της εξωτερικής βιβλιοθήκης `cz.msebera.android` για τη εισαγωγή του Apache HTTP server, που έχει βγει από το SDK 22 και μετά.

Το παρακάτω σχήμα δίνει μια γρήγορη επισκόπηση της εφαρμογής και των λειτουγιών της:



Σχήμα 10: Επισκόπηση των λειτουργιών της εφαρμογής

Ακόμη πάρθηκαν μερικά ακόμη μέτρα ασφαλείας για να σιγουρευτούν ότι μπορεί να γίνει επίθεση ακόμη και στις πιο ασφαλής εφαρμογές, εάν υπάρχει κάποια τρύπα ασφαλείας.

6.1 Εργαλεία

Τα εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη και την δοκιμή του λογισμικού, εφαρμογής, είναι:

Android Studio
Το Android Studio είναι ένα περιβάλλον ανάπτυξης εφαρμογών (IDE), το οποίο είναι σχεδιασμένο από την Google για αποκλειστική ανάπτυξη εφαρμογών στην πλατφόρμα του Android OS (33).

XAMPP
Το XAMPP είναι ένας δωρεάν και open source cross-platform web server που αναπτύχθηκε από την Apache Friends. Αποτελείτε κυρίως από έναν Apache HTTP server, βάση δεδομένων τύπου MariaDB και μπορεί να διαβάσει scripts γραμμένα σε PHP και Perl. Για την δημιουργία της βάσης δεδομένων της εφαρμογής και για την διαχείριση της, έγινε χρήση της MySQL και του PHPmyadmin που υπάρχουν μέσα στο XAMPP (34).

Sublime
Το Sublime είναι κειμενογράφος που σηκώνει αρκετές γλώσσες προγραμματισμού. Χρησιμοποιήθηκε για την γραφή των scripts σε γλώσσα PHP (35).

6.2 Μέτρα ασφαλείας

Σαν μέτρα ασφαλείας για την συγκεκριμένη έρευνα, πάρθηκαν τα εξής:

1. Όλες οι συνδέσεις με τον server, έχουν γίνει από δευτερεύων διαχειριστή της βάσης δεδομένων, ο οποίος έχει δικαιώματα μόνο στην βάση android.
2. Όλες οι συνδέσεις με τον server, γίνονται με χρήση HTTP.

6.3 User Stories

Ακόμη δημιουργήθηκαν μερικές ιστορίες χρήσης λογισμικού, που η εφαρμογή ακολουθεί και μπορεί να εκτελέσει. Οι περιπτώσεις χρήσεις είναι οι εξής:

1. Σαν χρήστης της εφαρμογής θα μπορώ να αυθεντικοποιηθώ για την χρήση της εφαρμογής.
2. Σαν χρήστης της εφαρμογής θα μπορώ να επιλέξω ένα από τα διαθέσιμα προϊόντα.
3. Σαν χρήστης της εφαρμογής θα μπορώ να επιλέξω την τιμή της ζάχαρης.
4. Σαν χρήστης της εφαρμογής θα μπορώ να δώσω παραγγελία.
5. Σαν χρήστης της εφαρμογής θα μπορώ να βλέπω όλες τις παραγγελίες που δεν είναι έτοιμες προς παράδοση.
6. Σαν χρήστης της εφαρμογής θα μπορώ να ακυρώσω μία ή περισσότερες παραγγελίες
7. Σαν χρήστης της εφαρμογής θα μπορώ να βλέπω όλες τις παραγγελίες που είναι έτοιμες προς παράδοση.
8. Σαν χρήστης της εφαρμογής θα μπορώ να δω το συνολικό κόστος κάθε τραπεζιού.

9. Σαν χρήστης της εφαρμογής θα μπορώ να διαγράψω παραγγελίες μετά την πληρωμή.

6.4 PHP Scripts

Πάνω σε μερικές από αυτές τις λειτουργίες θα γίνει η επίθεση εισαγωγής κώδικα, και θα γίνει και αποτροπή αυτών των επιθέσεων. Ακόμη, για την επίδειξη έχουν δημιουργηθεί 2 σετ από php scripts, τα οποία υπάρχουν στον server και η εφαρμογή, κάνει χρήση για να λειτουργήσει και να μπορέσει να συνδεθεί και να κάνει αλλαγές στην βάση. Το πρώτο σετ με php scripts, είναι ευάλωτο σε επιθέσεις injection, και δίνει μη εξουσιοδοτημένη είσοδο όταν δεχθεί επίθεση, επίσης χρησιμοποιεί τον χρήστη root ως διαχειριστή της βάσης. Ως root, δεν απαιτείται κωδικός για την σύνδεση με την βάση ή για τυχών αλλαγές σε πίνακες τις βάσης δεδομένων. Το δεύτερο σετ με php scritps, δεν είναι ευάλωτο σε επιθέσεις injection, κάνει χρήση του χρήστη fantomias, που είναι ο δευτερεύων διαχειριστής της βάσης. Για την σύνδεση με την βάση, ως δευτερεύων διαχειριστής, απαιτείται κωδικός, σε αυτή την περίπτωση ο κωδικός για τον χρήστη fantomias είναι “fantomakos”. Η εφαρμογή έχει αποθηκευμένο τον κωδικό μέσα της και τον στέλνει αυτόματα στην βάση στην αρχή, μόλις πάει να δίνει σύνδεση με την βάση. Σημαντικό είναι να τονιστεί ότι ο χρήστης της εφαρμογής δεν μπορεί να τον δει τον κωδικό και δεν μπορεί να τον αλλάξει σε καμία περίπτωση, όπως και ότι δεν γνωρίζει ότι γίνεται σύνδεση ως δευτερεύων διαχειριστής. Επίσης, ο δευτερεύων διαχειριστής fantomias, έχει το δικαίωμα να αλλάξει, να εισάγει και διαγράψει δεδομένα, μόνο στην βάση δεδομένων που χρησιμοποιείται για την λειτουργία της εφαρμογής.

6.5 Βάση δεδομένων

Για την βάση δεδομένων χρησιμοποιήθηκαν 4 πίνακες, ο πρώτος έχει όνομα user, και χρησιμοποιείτε στην αυθεντικοποίηση του χρήστη της εφαρμογής.

Όνομα πεδίου	Ρόλος στην βάση	Είδος
Username	το όνομα χρήστη που έχει ο χρήστης Είναι το πρωτεύον κλειδί του πίνακα	varchar (26)
Password	ο κωδικός χρήστη	varchar (26)

Ο δεύτερος πίνακας ονομάζεται user2, και χρησιμοποιείτε για την επίδειξη αυθεντικοποίησης πελάτη, με την χρήση της κρυπτογραφικής συνάρτησης κατακερματισμού MD5.

Όνομα πεδίου	Ρόλος στην βάση	Είδος
Username	το όνομα χρήστη που έχει ο χρήστης Είναι το πρωτεύον κλειδί του πίνακα	varchar (26)

Password	κατακερματισμένο κλειδί του κωδικού χρήστη	varchar (46)

Ο τρίτος πίνακας έχει τίτλος proionta και η εφαρμογή τον χρησιμοποιεί για να πάρει αυτόματα τα διαθέσιμα προϊόντα από την βάση.

Όνομα πεδίου	Ρόλος στην βάση	Είδος
name	όνομα του προϊόντος Είναι το πρωτεύον κλειδί του πίνακα	varchar (26)
timi	τιμή του προϊόντος	decimal (5,2)
eidos	είδος του προϊόντος	varchar (26)

Τέταρτος και τελευταίος πίνακας είναι ο paraggelies. Σε αυτό τον πίνακα αποθηκεύονται οι παραγγελίες που χειρίζεται η εφαρμογή.

Όνομα πεδίου	Ρόλος στην βάση	Είδος
aa	αύξων αριθμός της παραγγελίας Είναι το πρωτεύον κλειδί του πίνακα	integer (11)
NoTrap	ο αριθμός του τραπεζιού που έγινε και πρέπει να πάει η παραγγελία	integer (11)
proion	όνομα του προϊόντος που παραγγέλθηκε	varchar (26)
Zaxari	τιμές της ποσότητας ζάχαρης που θα έχει το προϊον	varchar (26)
etoimo	αν είναι έτοιμο το προϊόν	tinyint(1)
Parad	αν έχει παραδοθεί το προϊόν στον πελάτη	tinyint(1)

Με την χρήση αυτών των τεσσάρων πινάκων η εφαρμογή, μπορεί να κάνει αυθεντικοποίηση, εισαγωγή δεδομένων, αλλαγή δεδομένων και την διαγραφή τους.

6.6 Activities

Σε αυτό το υπο - κεφάλαιο θα εμφανιστεί ο κώδικας της εφαρμογής ανά activity που χρησιμοποιεί το android. Πρώτα θα εμφανιστεί ο κώδικας XML, δηλαδή ο κώδικας για το γραφικό περιβάλλον που βλέπει ο χρήστης και μετά οι μέθοδοι για την λειτουργικότητα.

Activity main_activity:

XML:

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context="gr.greg.ptuxiaki.MainActivity"
    tools:showIn="@layout/activity_main">

    <TextView
        android:text="Log In"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/Tlog"
        android:textSize="36dp"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="76dp" />

    <EditText
        android:layout_width="200dp"
        android:layout_height="30dp"
        android:id="@+id/usr"
        android:hint="Username"
        android:layout_marginTop="37dp"
        android:layout_below="@+id/Tlog"
        android:layout_centerHorizontal="true"
        android:gravity="center"
        android:background="#ffffff" />

    <EditText
        android:layout_width="200dp"
        android:layout_height="30dp"
        android:inputType="textPassword"
        android:ems="5"
        android:id="@+id/pwd"
        android:hint="Password"
        android:layout_centerVertical="true"
        android:layout_alignLeft="@+id/usr"
        android:layout_alignStart="@+id/usr"
```

```
        android:gravity="center"
        android:background="#ffffff" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="GO"
        android:id="@+id/button"
        android:layout_below="@+id/pwd"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="69dp" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Sign Up"
        android:id="@+id/button2"
        android:layout_below="@+id/button"
        android:layout_centerHorizontal="true" />


</RelativeLayout>
```

Java:

```
package gr.greg.ptuxiaki;

import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;

import android.os.StrictMode;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import org.json.JSONException;
import org.json.JSONObject;

import java.io.BufferedWriter;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.ProtocolException;
```

```
import java.net.URL;



public class MainActivity extends AppCompatActivity {
    private Global gl=new Global();
    EditText ETuser,ETpwd;
    Button btn,btn2;
    InputStream is;
    OutputStream os;
    URL url;
    StringBuffer sb;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);


        StrictMode.ThreadPolicy         policy         =         new
StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);
        ETuser=(EditText) findViewById(R.id.usr);
        ETpwd=(EditText) findViewById(R.id.pwd);
        btn=(Button) findViewById(R.id.button);
        btn2=(Button) findViewById(R.id.button2);


        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                HttpURLConnection conn = null;
                try {
                    url = gl.login();
                    conn = (HttpURLConnection) url.openConnection();
                    conn.setDoInput(true);
                    conn.setDoOutput(true);
                    conn.setUseCaches(false);
                } catch (IOException e) {
                    e.printStackTrace();
                }
                try {
                    conn.setRequestMethod("POST");
                } catch (ProtocolException e) {
                    e.printStackTrace();
                }
```

```
                conn.setRequestProperty("Content-Type",
"application/x-www-form-urlencoded"); //header
                Uri.Builder builder = new Uri.Builder()
                        .appendQueryParameter("pwd",
ETpwd.getText().toString())
                        .appendQueryParameter("user",
ETuser.getText().toString());
                //kwdikopoiw
                String query = builder.build().getEncodedQuery();


                BufferedWriter writer = null;
                try {
                    writer       =       new       BufferedWriter(new
OutputStreamWriter(conn.getOutputStream(), "UTF-8"));
                } catch (IOException e){e.printStackTrace();}
                //pernaw tis metablites sto php arxeio
                try {

                        writer.write(query);
                        writer.flush();
                        writer.close();


                } catch(NullPointerException h){
                    h.printStackTrace();
                }catch(IOException e){e.printStackTrace();}

                try {
                    is=conn.getInputStream();
                    int ch;
                    sb = new StringBuffer();
                    while ((ch = is.read()) != -1) {
                        sb.append((char) ch);
                    }
                    Log.d("response", sb.toString());

                    if (is != null)
                        is.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
                try {
                    JSONObject jOb = new JSONObject(sb.toString());
                    if (jOb.getBoolean("found")) {
                        Toast.makeText(getApplicationContext(),
"Redirection...", Toast.LENGTH_SHORT).show();
```

```
                        Intent intent = new
Intent(MainActivity.this, Btn_Activity.class);
                        startActivity(intent);
                    } else {
                        Toast.makeText(getApplicationContext(),
"Wrong Credentialz...", Toast.LENGTH_SHORT).show();
                    }
                } catch (JSONException e) {
                    e.printStackTrace();
                }
            }
        });

        btn2.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                Toast.makeText(getApplicationContext(),
"Redirection...", Toast.LENGTH_SHORT).show();
                Intent intent = new Intent(MainActivity.this,
newUser.class);
                startActivity(intent);
            }
        });

    }


}
```

Activity Btn_activity:

XML

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context="gr.greg.ptuxiaki.Btn_Activity"
    tools:showIn="@layout/activity_btn">

    <Button
        android:layout_width="200dp"
```

```
        android:layout_height="wrap_content"
        android:text="Εισαγωγη"
        android:id="@+id/btn_input"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="53dp" />

    <Button
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:text="not ready"
        android:id="@+id/btn_notRdy"
        android:layout_below="@+id/btn_input"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="60dp" />

    <Button
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:text="Ready"
        android:id="@+id/ready"
        android:layout_below="@+id/btn_notRdy"
        android:layout_alignLeft="@+id/plero"
        android:layout_alignStart="@+id/plero"
        android:layout_marginTop="60dp" />

    <Button
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:text="Pay"
        android:id="@+id/plero"
        android:layout_below="@+id/ready"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="60dp" />

</RelativeLayout>
```

Java:

```
package gr.greg.ptuxiaki;

import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;

public class Btn_Activity extends AppCompatActivity {
```

```
    Button insert,showNotReady,showReady,plero;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_btn);
        Init(); //klisi sunartishs arxikopoihshs
    }

    private void Init() {
        insert    =    (Button)    this.findViewById(R.id.btn_input);
// init to 1o koumpi
        insert.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                Intent   intent   =   new   Intent(Btn_Activity.this,
order.class);
                startActivity(intent);
            }
        });

        showNotReady = (Button) this.findViewById(R.id.btn_notRdy);
        showNotReady.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                Intent   intent   =   new   Intent(Btn_Activity.this,
showData.class);
                startActivity(intent);
            }
        });

        showReady = (Button) this.findViewById(R.id.ready);
        showReady.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent   intent   =   new   Intent(Btn_Activity.this,
readyData.class);
                startActivity(intent);
            }
        });

        plero= (Button) this.findViewById(R.id.plero);
        plero.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent   intent   =   new   Intent(Btn_Activity.this,
exterminatus.class);
                startActivity(intent);
            }
```

```
        });
    }

}
```

Activity Order:
XML:

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context="gr.greg.ptuxiaki.order"
    tools:showIn="@layout/activity_order">

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true">

        <LinearLayout
            android:orientation="horizontal"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="10dp">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"

android:textAppearance="?android:attr/textAppearanceMedium"
                android:text="Αριθμός Τραπεζιού"
                android:id="@+id/Trap"
                android:layout_marginRight="10px" />

            <EditText
                android:layout_width="match_parent"
                android:layout_height="30dp"
                android:id="@+id/arT"
                android:inputType="number"
```

```
                android:background="#ffffff" />

        </LinearLayout>

        <Spinner
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/spinner"
            android:background="#ffffff" />

        <Spinner
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/spinner2"
            android:background="#ffffff" />

    </LinearLayout>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Παραγγελία"
        android:id="@+id/input"
        android:layout_gravity="center_horizontal"
        android:layout_centerVertical="true"
        android:layout_centerHorizontal="true" />

</RelativeLayout>
```

Java:

```
package gr.greg.ptuxiaki;

import android.content.DialogInterface;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Bundle;

import android.os.StrictMode;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.Toast;
```

```
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.io.UnsupportedEncodingException;
import java.net.HttpURLConnection;
import java.net.ProtocolException;
import java.net.URL;
import java.util.ArrayList;
import java.util.List;

import cz.msebera.android.httpclient.HttpEntity;
import cz.msebera.android.httpclient.HttpResponse;
import cz.msebera.android.httpclient.NameValuePair;
import cz.msebera.android.httpclient.client.HttpClient;
import
cz.msebera.android.httpclient.client.entity.UrlEncodedFormEntity;
import cz.msebera.android.httpclient.client.methods.HttpPost;
import cz.msebera.android.httpclient.impl.client.DefaultHttpClient;
import cz.msebera.android.httpclient.message.BasicNameValuePair;


@SuppressWarnings( "deprecation" )
public class order extends AppCompatActivity {
    EditText noT; //ari8mos trapeziou
    Spinner proion, zax; //proion basis + zaxari apo to strings.xml
    Button in;
    Global gl=new Global();
    URL url;

    ArrayList<String> itemList=new ArrayList<>();
    ArrayList<String> kategList=new ArrayList<String>();
    ArrayAdapter<String> adapter;
    ArrayAdapter<String> Zadapter;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_order);
```

```
        StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);//energopoioume to strict
mode gia na mhn xrisimopoihsoume kati p den uparxei


        //Arikopoihsh twn metablitwn mas
        noT=(EditText) findViewById(R.id.arT);
        proion=(Spinner) findViewById((R.id.spinner));
        zax=(Spinner) findViewById((R.id.spinner2));
        in=(Button) findViewById(R.id.input);

        //Etoimazoume ton adapter pou 8a dextei ta proionta ths
baseis kai ton bazoume sto spinner
        adapter=new
ArrayAdapter<String>(this,R.layout.support_simple_spinner_dropdown_i
tem,itemList);
        //pairnw to Array List apo to res/values/string poy exei tis
times tis zaxaris
        Zadapter=new ArrayAdapter(this,
android.R.layout.simple_spinner_item,getResources().getStringArray(R
.array.list));

Zadapter.setDropDownViewResource(android.R.layout.simple_spinner_dro
pdown_item);

        proion.setAdapter(adapter);
        zax.setAdapter(Zadapter);

        //stin periptosh pou to proion einai sthn katigoria "rofima"
energopoihte to spinner gia epilogi zaxaris
        //alliws den fainetai kai h timi tou einai panta h 1 sthn
lista, diladi "oxi"
        proion.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener() {

            public void onItemSelected(AdapterView<?> parentView,
View selectedItemView, int position, long id) {
                if (kategList.get(position).equals("rofima")) {
                    zax.setVisibility(View.VISIBLE);
                } else {
                    zax.setSelection(0);
                    zax.setVisibility(View.INVISIBLE);
                }
            }

            @Override
            public void onNothingSelected(AdapterView<?> parent) {
```

```
            }

        });



        //Otan pati8ei to koympi gia Eisagwgh
        in.setOnClickListener(new View.OnClickListener(){

            InputStream is = null;
            OutputStream os = null;

            @Override
            public void onClick(View v) {

                if (Integer.valueOf(noT.getText().toString())> 0){
                    String arTrap = noT.getText().toString();
                    String                    pro                    =
proion.getSelectedItem().toString();

                    // Stelnw dedomena me methodo POST
                    HttpURLConnection conn = null;

                    try {
                        url = gl.getPostUrl();
                        conn          =          (HttpURLConnection)
url.openConnection();
                        conn.setDoInput(true);
                        conn.setDoOutput(true);
                        conn.setUseCaches(false);
                    } catch (IOException e) {
                        e.printStackTrace();
                    }
                    try {
                        conn.setRequestMethod("POST");
                    } catch (ProtocolException e) {
                        e.printStackTrace();
                    }
                    conn.setRequestProperty("Content-Type",
"application/x-www-form-urlencoded"); //den kserw ti einai alla to
xreiazetai
                    // bazw parametrous kai times gia to arxeio
                    Uri.Builder builder = new Uri.Builder()
                            .appendQueryParameter("pass",
gl.getPass())
                            .appendQueryParameter("NoTrap", arTrap)
```

```
                            .appendQueryParameter("proion", pro)
                            .appendQueryParameter("zaxari",
zax.getSelectedItem().toString());
                    //kwdikopoiw
                    String                    query                    =
builder.build().getEncodedQuery();

                    try {
                        os = conn.getOutputStream();
                    } catch (IOException e) {
                        e.printStackTrace();
                    }
                    BufferedWriter writer = null;
                    try {
                        writer     =     new     BufferedWriter(new
OutputStreamWriter(os, "UTF-8"));
                    } catch (UnsupportedEncodingException e) {
                        e.printStackTrace();
                    }
                    //pernaw tis metablites sto php arxeio
                    try {
                        writer.write(query);
                        writer.flush();
                        Toast.makeText(order.this,       "Inserted",
Toast.LENGTH_SHORT).show();
                        writer.close();
                        os.close();
                    } catch (IOException e) {
                        e.printStackTrace();
                        Toast.makeText(order.this, "Error Occurred",
Toast.LENGTH_SHORT).show();
                    }

                    try {
                        is = conn.getInputStream();
                        int ch;
                        StringBuffer sb = new StringBuffer();
                        while ((ch = is.read()) != -1) {
                            sb.append((char) ch);
                        }
                        Log.d("response", sb.toString());
                        if (is != null)
                            is.close();
                    } catch (IOException e) {
                        e.printStackTrace();
                    }

                }else{
```

```
                    new AlertDialog.Builder(order.this)
                            .setTitle("Προσοχή")
                            .setMessage("Βάλατε μη δεκτό αριθμό
τραπεζιού")
                            .setPositiveButton(android.R.string.yes,
new DialogInterface.OnClickListener() {
                                public void onClick(DialogInterface
dialog, int which) {
                                }
                            })

.setIcon(android.R.drawable.ic_dialog_alert)
                            .show();
                }
            }
        });
    }

    protected  void onStart(){
        super.onStart();
        BackTask bt=new BackTask();
        bt.execute();
    }
    private class BackTask extends AsyncTask<Void,Void,Void> {

        ArrayList<String> list,alist;

        protected void onPreExecute(){

            super.onPreExecute();
            list= new ArrayList<>();
            alist=new ArrayList<>();
        }

        @Override
        protected Void doInBackground(Void... params) {
            InputStream is = null;
            String result = "";
            try {

                HttpClient client = new DefaultHttpClient();
                HttpPost postm = new HttpPost(gl.getSpinUrl());
                List<NameValuePair>        list        =        new
ArrayList<NameValuePair>(1);
                list.add(new
BasicNameValuePair("pass",gl.getPass()));
                postm.setEntity(new
UrlEncodedFormEntity(list));//kwdikopoiw
```

```
                HttpResponse                response                =
client.execute(postm);//ektelesi tou php
                HttpEntity entity = response.getEntity();
                is = entity.getContent();

            } catch (IOException e) {
                e.printStackTrace();
            }
            //metatropi tis apantisis se string
            try{

                BufferedReader   br   =   new   BufferedReader(new
InputStreamReader(is,"utf-8"));
                String line="";
                while ((line=br.readLine()) != null)
                {
                    result+=line;
                }
                is.close();

            } catch(IOException e){ e.printStackTrace();}

            //analisi tou json
            try{
                JSONArray array= new JSONArray(result);

                for (int i=0; i<array.length(); i++)
                {
                    JSONObject jOb= array.getJSONObject(i);
                    list.add(jOb.getString("name"));
                    alist.add(jOb.getString("kateg"));
                }

            } catch(JSONException e){ e.printStackTrace();}
            return null;
         }


        protected void onPostExecute(Void result){
            itemList.addAll(list);
            kategList.addAll(alist);
            adapter.notifyDataSetChanged();//to  xreiazetai  gia  na
mpoyn kainoyrgia proionta

        }
    }
}
```

Για τις επόμενες activities, έχουν χρησιμοποιηθεί και βοηθιτικές κλάσεις. Οι κλάσεις τύπου Connector, που κάνουν την συναλλαγή πληροφοριών με τους server και οι κλάσεις Adapter που ετοιμάζουν και ομαδοποιούν τα δεδομένα με συγκεκριμένο τρόπο ώστε να τα δει με όμορφο τρόπο ο χρήστης. Για αυτού του είδους τις Activities, έχει χρησιμοποιηθεί μόνο ένα XML αρχείο για εμφάνιση γραφικών στο χρήστη.

Activity showData:

XML:

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context="gr.greg.ptuxiaki.showData">

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:gravity="center_horizontal"
        android:id="@+id/linearLayout3"
        android:paddingRight="10dp">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"

android:textAppearance="?android:attr/textAppearanceMedium"
            android:text="Trapezi"
            android:id="@+id/textView3"
            android:layout_marginLeft="0dp" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"

android:textAppearance="?android:attr/textAppearanceMedium"
            android:text="Medium Text"
            android:id="@+id/noTrap"
            android:paddingRight="10dp" />
```

```
    </LinearLayout>

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="140dp"
        android:layout_height="match_parent"
        android:gravity="center_horizontal"
        android:id="@+id/linearLayout2"
        android:layout_alignParentTop="true"
        android:layout_toRightOf="@+id/linearLayout3"
        android:layout_toEndOf="@+id/linearLayout3">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"

android:textAppearance="?android:attr/textAppearanceMedium"
            android:text="Proion"
            android:id="@+id/textView4" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"

android:textAppearance="?android:attr/textAppearanceMedium"
            android:text="Medium Text"
            android:id="@+id/name"
            android:paddingRight="10dp" />

    </LinearLayout>

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="120dp"
        android:layout_height="match_parent"
        android:gravity="center_horizontal"
        android:id="@+id/linearLayout4"
        android:layout_alignParentTop="true"
        android:layout_toRightOf="@+id/linearLayout2"
        android:layout_toEndOf="@+id/linearLayout2">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"

android:textAppearance="?android:attr/textAppearanceMedium"
            android:text="Zaxari"
```

```
            android:id="@+id/TextView7" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"

android:textAppearance="?android:attr/textAppearanceMedium"
            android:text="Medium Text"
            android:id="@+id/zaxari" />
    </LinearLayout>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Done"
        android:id="@+id/doneButt"
        android:layout_alignParentTop="true"
        android:layout_toRightOf="@+id/linearLayout4"
        android:layout_toEndOf="@+id/linearLayout4"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Cancel"
        android:id="@+id/cancer"
        android:layout_alignParentTop="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true" />
</RelativeLayout>
```

Java:

showData.java:

```
package gr.greg.ptuxiaki;

import android.content.pm.ActivityInfo;
import android.os.AsyncTask;
import android.os.Bundle;
import android.os.Handler;
import android.support.v7.app.AppCompatActivity;
import android.widget.ListView;

import org.json.JSONArray;

import java.util.Timer;
import java.util.TimerTask;
```

```
public class showData extends AppCompatActivity {


    private ListView theList;
    Timer timer=new Timer();

        final Handler handler = new Handler();
        TimerTask task = new TimerTask() {
            @Override
            public void run() {
                handler.post(new Runnable() {
                    public void run() {
                        new              fetchNotReady().execute(new
showDataCon());
                    }
                });
            }
        };



    @Override
    protected void onCreate(Bundle savedInstanceState){
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_show_data);

setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE);//
//bazw thn o8oni na einai plageia
        theList=(ListView) this.findViewById(R.id.theList);

        timer.schedule(task, 0, 3000); //ekteleite ka8e 3000ms

    }


    public void setListAdapter(JSONArray array) {
        theList.setAdapter(new showDataAdapter(array, this));
    }


    //ksekinaw ksexwristo nima me thn trofodosia ths listas twn
proiontwn
    public           class           fetchNotReady           extends
AsyncTask<showDataCon,Long,JSONArray> {
        @Override
        protected JSONArray doInBackground(showDataCon... params) {
            return params[0].getAllData();
        }
```

```
        @Override
        protected void onPostExecute(JSONArray jsonArray) {
            setListAdapter(jsonArray);
        }
    }

    @Override
    public void onBackPressed() {
        timer.cancel(); /* an den to stamatisw tote to timertask 8a
sunexisei na leitourgei akomi kai otan 8a kleisei h Activity
        mexri na kleisei oli h efarmogh*/
        timer=null;
        handler.removeCallbacks(null);
        super.onBackPressed();
    }

}
```

showDataCon.java:

```
package gr.greg.ptuxiaki;

import android.net.Uri;
import android.util.Log;
import android.widget.Toast;

import org.json.JSONArray;
import org.json.JSONException;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.io.UnsupportedEncodingException;
import java.net.HttpURLConnection;
import java.net.ProtocolException;
import java.net.URL;

/**
 * Created by Greg on 7/3/2016.
 */

//trabaei ta dedomena olwn oswn den einai etoima apo thn basi kai ta
bazei se ena JSON
```

```
public class showDataCon {

    public JSONArray getAllData() {
        JSONArray array;
        Global gl = new Global();
        InputStream is ;
        OutputStream os = null;
        URL url;
        HttpURLConnection conn = null;
        String result = "";

        try {
            url = gl.getNotReady();
            conn = (HttpURLConnection) url.openConnection();
            conn.setDoInput(true);
            conn.setDoOutput(true);
            conn.setUseCaches(false);

        } catch (IOException e) {
            e.printStackTrace();
        }
        try {
            conn.setRequestMethod("POST");
        } catch (ProtocolException e) {
            e.printStackTrace();
        }
        conn.setRequestProperty("Content-Type",  "application/x-www-
form-urlencoded"); //den kserw ti einai alla to xreiazetai
        // bazw parametrous kai times gia to arxeio
        Uri.Builder builder = new Uri.Builder()
                .appendQueryParameter("pass", gl.getPass());
        //kwdikopoiw
        String query = builder.build().getEncodedQuery();

        try {
            os = conn.getOutputStream();
        } catch (IOException e) {
            e.printStackTrace();
        }
        BufferedWriter writer = null;
        try {
            writer  =  new  BufferedWriter(new  OutputStreamWriter(os,
"UTF-8"));
        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
        }
        //pernaw tis metablites sto php arxeio
        try {
```

```
            writer.write(query);
            writer.flush();
            //            Toast.makeText(order.this,      "Inserted",
Toast.LENGTH_SHORT).show();
            writer.close();
            os.close();
        } catch (IOException e) {
            e.printStackTrace();
            //                        Toast.makeText(order.this,"Error
Occurred",Toast.LENGTH_SHORT).show();
        }

        try {
            is = conn.getInputStream();
            BufferedReader    br    =    new    BufferedReader(new
InputStreamReader(is, "utf-8"));
            String line = "";
            while ((line = br.readLine()) != null) {
                result += line;
            }
            is.close();

        } catch (IOException e) {
            e.printStackTrace();
        }
        try {
            array = new JSONArray(result);
            return array;
        } catch (JSONException e) {
            e.printStackTrace();
        }
        return null;
    }
}
```

showDataAdapter:

```
package gr.greg.ptuxiaki;

import android.net.Uri;
import android.util.Log;
import android.widget.Toast;

import org.json.JSONArray;
import org.json.JSONException;

import java.io.BufferedReader;
import java.io.BufferedWriter;
```

```
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.io.UnsupportedEncodingException;
import java.net.HttpURLConnection;
import java.net.ProtocolException;
import java.net.URL;

/**
 * Created by Greg on 7/3/2016.
 */

//trabaei ta dedomena olwn oswn den einai etoima apo thn basi kai ta
bazei se ena JSON
public class showDataCon {

    public JSONArray getAllData() {
        JSONArray array;
        Global gl = new Global();
        InputStream is ;
        OutputStream os = null;
        URL url;
        HttpURLConnection conn = null;
        String result = "";

        try {
            url = gl.getNotReady();
            conn = (HttpURLConnection) url.openConnection();
            conn.setDoInput(true);
            conn.setDoOutput(true);
            conn.setUseCaches(false);

        } catch (IOException e) {
            e.printStackTrace();
        }
        try {
            conn.setRequestMethod("POST");
        } catch (ProtocolException e) {
            e.printStackTrace();
        }
        conn.setRequestProperty("Content-Type",  "application/x-www-
form-urlencoded"); //den kserw ti einai alla to xreiazetai
        // bazw parametrous kai times gia to arxeio
        Uri.Builder builder = new Uri.Builder()
                .appendQueryParameter("pass", gl.getPass());
        //kwdikopoiw
```

```
        String query = builder.build().getEncodedQuery();

        try {
            os = conn.getOutputStream();
        } catch (IOException e) {
            e.printStackTrace();
        }
        BufferedWriter writer = null;
        try {
            writer = new BufferedWriter(new OutputStreamWriter(os,
"UTF-8"));
        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
        }
        //pernaw tis metablites sto php arxeio
        try {
            writer.write(query);
            writer.flush();
            //           Toast.makeText(order.this,      "Inserted",
Toast.LENGTH_SHORT).show();
            writer.close();
            os.close();
        } catch (IOException e) {
            e.printStackTrace();
            //                       Toast.makeText(order.this,"Error
Occurred",Toast.LENGTH_SHORT).show();
        }

        try {
            is = conn.getInputStream();
            BufferedReader    br    =    new    BufferedReader(new
InputStreamReader(is, "utf-8"));
            String line = "";
            while ((line = br.readLine()) != null) {
                result += line;
            }
            is.close();

        } catch (IOException e) {
            e.printStackTrace();
        }
        try {
            array = new JSONArray(result);
            return array;
        } catch (JSONException e) {
            e.printStackTrace();
        }
        return null;
```

```
    }
}
```

Activity readyData:

XML:

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context="gr.greg.ptuxiaki.readyData">
    <LinearLayout
        android:orientation="vertical"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:gravity="center_horizontal"
        android:id="@+id/linearLayout3"
        android:paddingRight="10dp">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"

android:textAppearance="?android:attr/textAppearanceMedium"
            android:text="Trapezi"
            android:id="@+id/textView3"
            android:layout_marginLeft="0dp" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"

android:textAppearance="?android:attr/textAppearanceMedium"
            android:text="Medium Text"
            android:id="@+id/noTrap"
            android:paddingRight="10dp" />

    </LinearLayout>

    <LinearLayout
```

```
        android:orientation="vertical"
        android:layout_width="140dp"
        android:layout_height="match_parent"
        android:gravity="center_horizontal"
        android:id="@+id/linearLayout2"
        android:layout_alignParentTop="true"
        android:layout_toRightOf="@+id/linearLayout3"
        android:layout_toEndOf="@+id/linearLayout3">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"

android:textAppearance="?android:attr/textAppearanceMedium"
            android:text="Proion"
            android:id="@+id/textView4" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"

android:textAppearance="?android:attr/textAppearanceMedium"
            android:text="Medium Text"
            android:id="@+id/name"
            android:paddingRight="10dp" />

    </LinearLayout>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Gone"
        android:id="@+id/goneButt"
        android:layout_alignParentTop="true"
        android:layout_toRightOf="@+id/linearLayout2"
        android:layout_toEndOf="@+id/linearLayout2" />
</RelativeLayout>
```

readyData.java:

```
package gr.greg.ptuxiaki;

import android.os.AsyncTask;
import android.os.Bundle;
import android.os.Handler;
import android.support.v7.app.AppCompatActivity;
import android.widget.ListView;
```

```
import org.json.JSONArray;

import java.util.Timer;
import java.util.TimerTask;


public class readyData extends AppCompatActivity {
    ListView theList;
    Timer timer = new Timer();

    final Handler handler = new Handler();
    TimerTask task = new TimerTask() {
        @Override
        public void run() {
            handler.post(new Runnable() {
                public void run() {
                    new fetchReady().execute(new readyDataCon());
                }
            });
        }
    };


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_ready_data);

        this.theList = (ListView) this.findViewById(R.id.aList);

        timer.schedule(task, 0, 3000); //ekteleite ka8e 3000ms
    }

    public void setListAdapter(JSONArray array) {
        this.theList.setAdapter(new readyDataAdapter(array, this));
    }

    public class fetchReady extends AsyncTask<readyDataCon, Long,
JSONArray> {
        @Override
        protected JSONArray doInBackground(readyDataCon... params) {
            return params[0].getReadyData();
        }

        @Override
        protected void onPostExecute(JSONArray jsonArray) {
            setListAdapter(jsonArray);
        }
```

```
    }
}
```

readyDataCon.java:

```
package gr.greg.ptuxiaki;

import android.net.Uri;

import org.json.JSONArray;
import org.json.JSONException;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.io.UnsupportedEncodingException;
import java.net.HttpURLConnection;
import java.net.ProtocolException;
import java.net.URL;

/**
 * Created by Greg on 21/3/2016.
 */
public class readyDataCon {
    public JSONArray getReadyData(){

        JSONArray array;
        Global gl = new Global();
        InputStream is;
        OutputStream os = null;
        URL url;
        HttpURLConnection conn = null;
        String result = "";

        try {
            conn.setDoInput(true);
            conn.setDoOutput(true);
            conn.setUseCaches(false);
        } catch (NullPointerException e) {
            e.printStackTrace();
        }
        try {
            url = gl.getReady();
            conn = (HttpURLConnection) url.openConnection();
```

```
        } catch (IOException e) {
            e.printStackTrace();
        }
        try {
            conn.setRequestMethod("POST");
        } catch (ProtocolException e) {
            e.printStackTrace();
        }
        conn.setRequestProperty("Content-Type",  "application/x-www-
form-urlencoded"); //den kserw ti einai alla to xreiazetai
        // bazw parametrous kai times gia to arxeio
        Uri.Builder builder = new Uri.Builder()
                .appendQueryParameter("pass", gl.getPass());
        //kwdikopoiw
        String query = builder.build().getEncodedQuery();

        try {
            os = conn.getOutputStream();
        } catch (IOException e) {
            e.printStackTrace();
        }
        BufferedWriter writer = null;
        try {
            writer  =  new  BufferedWriter(new  OutputStreamWriter(os,
"UTF-8"));
        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
        }
        //pernaw tis metablites sto php arxeio
        try {
            writer.write(query);
            writer.flush();
                        writer.close();
            os.close();
        } catch (IOException e) {
            e.printStackTrace();

        }
        try {
            is = conn.getInputStream();
            BufferedReader    br    =    new    BufferedReader(new
InputStreamReader(is, "utf-8"));
            String line = "";
            while ((line = br.readLine()) != null) {
                result += line;
            }
            is.close();
```

```
        } catch (IOException e) {
            e.printStackTrace();
        }

        try {
            array = new JSONArray(result);
            return array; //epistrefw ton pinaka
        } catch (JSONException e) {
            e.printStackTrace();
        }
        return null; //se periptwsi poy uparxei problima, epistrefw
null, logika crash h efarmogi
    }
}
```

readyDataAdapter.java:

```
package gr.greg.ptuxiaki;

import android.app.Activity;
import android.content.Context;
import android.os.StrictMode;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.IOException;
import java.io.InputStream;
import java.util.ArrayList;
import java.util.List;

import cz.msebera.android.httpclient.HttpEntity;
import cz.msebera.android.httpclient.HttpResponse;
import cz.msebera.android.httpclient.NameValuePair;
import cz.msebera.android.httpclient.client.ClientProtocolException;
import cz.msebera.android.httpclient.client.HttpClient;
import
cz.msebera.android.httpclient.client.entity.UrlEncodedFormEntity;
import cz.msebera.android.httpclient.client.methods.HttpPost;
import cz.msebera.android.httpclient.impl.client.DefaultHttpClient;
```

```
import cz.msebera.android.httpclient.message.BasicNameValuePair;

/**
 * Created by Greg on 21/3/2016.
 */
public class readyDataAdapter extends BaseAdapter {
    private JSONArray dataArray;
    Global gl=new Global();
    private Activity act;
    private LayoutInflater inflater=null;
    public int a=0;
    InputStream is= null;

    public readyDataAdapter (JSONArray array, Activity ab){
        this.dataArray=array;
        this.act=ab;
        inflater=(LayoutInflater)
this.act.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    }
    @Override
    public int getCount() {
        //sthn periptosh pou den yparxei kati poy na mhn einai
etoimo, 8a epistrepsei 0
        //alliws epistrefei Null kai crasharei h efarmogi
        try{
            return dataArray.length();
        }catch(NullPointerException e){
            return 0;
        }
    }

    @Override
    public Object getItem(int position) {
        return null;
    }

    @Override
    public long getItemId(int position) {
        return 0;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup
parent) {
        ListCell cell;
        StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);
```

```
        if(convertView==null){

            convertView=
inflater.inflate(R.layout.content_ready_data,null);
            cell= new ListCell();

            cell.noTrap=(TextView)
convertView.findViewById(R.id.noTrap);
            cell.name=(TextView)
convertView.findViewById(R.id.name);
            cell.Bdone=(Button)
convertView.findViewById(R.id.goneButt);

            convertView.setTag(cell);
        }else{
            cell=(ListCell) convertView.getTag();
        }

        try {
            JSONObject jOb = this.dataArray.getJSONObject(position);
            cell.noTrap.setText(jOb.getString("NoTrap"));
            cell.name.setText(jOb.getString("proion"));
            cell.isDone= jOb.getInt("aa");
            a=cell.isDone;


        }catch (JSONException e) {
            e.printStackTrace();
        }
        cell.Bdone.setOnClickListener(new View.OnClickListener() {

            String  swap=  String.valueOf(a);//metatrepw  ton  ari8mo
trapezioy se String

            @Override
            public void onClick(View v) {

                try{
                    //dimioyrgw  lista  typou  "name+key"  kai  pernaw
tis metablites
                    List<NameValuePair>       list       =       new
ArrayList<NameValuePair>(1);
                    list.add(new
BasicNameValuePair("pass",gl.getPass()));
                    list.add(new BasicNameValuePair("art", swap));
```

```
                    HttpClient client= new DefaultHttpClient();

                    HttpPost                postm=                  new
HttpPost(gl.postGone());//epilegw to script pou 8elw


                    postm.setEntity(new
UrlEncodedFormEntity(list));//kwdikopoiw

                    HttpResponse              response              =
client.execute(postm);//ektelw

                    Toast.makeText(v.getContext(),           "Gone",
Toast.LENGTH_SHORT).show();

                    HttpEntity entity=  response.getEntity();

                    is=entity.getContent();

                }catch(ClientProtocolException
e){e.printStackTrace();
                }catch(IOException e){e.printStackTrace();}
            }

        });
        return convertView;
    }
}
```

Activity exterminatus:

XML:

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context=".exterminatus"
    tools:showIn="@layout/activity_exterminatus">

    <TextView
        android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:text="Αριθμός Τραπεζιού"
    android:id="@+id/textView7"
    android:layout_alignParentTop="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_marginTop="5dp" />

<EditText
    android:layout_width="120dp"
    android:layout_height="30dp"
    android:ems="10"
    android:id="@+id/numb"
    android:layout_alignTop="@+id/textView7"
    android:layout_alignRight="@+id/theList"
    android:layout_alignEnd="@+id/theList"
    android:inputType="number"
    android:background="#FFFFFF" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Bgale"
    android:id="@+id/print"
    android:layout_below="@+id/textView7"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="46dp" />

<ListView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/theList"
    android:headerDividersEnabled="false"
    android:layout_below="@+id/print"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_above="@+id/textView8" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:text="Synolo"
    android:id="@+id/textView8"
    android:layout_alignParentBottom="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />
```

```
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:text="0.00"
        android:id="@+id/sum"
        android:layout_alignParentBottom="true"
        android:layout_alignLeft="@+id/print"
        android:layout_alignStart="@+id/print" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Pliro8ike"
        android:id="@+id/pay"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:visibility="invisible" />
</RelativeLayout>
```

Java:

exterminatus.java:

```
package gr.greg.ptuxiaki;

import android.os.AsyncTask;
import android.os.Bundle;
import android.os.StrictMode;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.TextView;

import org.json.JSONArray;

import java.io.IOException;
import java.io.InputStream;
import java.util.ArrayList;
import java.util.List;

import cz.msebera.android.httpclient.HttpEntity;
import cz.msebera.android.httpclient.HttpResponse;
```

```
import cz.msebera.android.httpclient.NameValuePair;
import cz.msebera.android.httpclient.client.ClientProtocolException;
import cz.msebera.android.httpclient.client.HttpClient;
import
cz.msebera.android.httpclient.client.entity.UrlEncodedFormEntity;
import cz.msebera.android.httpclient.client.methods.HttpPost;
import cz.msebera.android.httpclient.impl.client.DefaultHttpClient;
import cz.msebera.android.httpclient.message.BasicNameValuePair;
import cz.msebera.android.httpclient.protocol.HTTP;

public class exterminatus extends AppCompatActivity {

    Button Bprint;
    static Button Bpay;
    static EditText trapN;
    static TextView sum;
    ListView theList;
    static Boolean bool=false;
    Global gl=new Global();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_exterminatus);
        Bprint=(Button)findViewById(R.id.print);
        Bpay=(Button)findViewById(R.id.pay);
        trapN=(EditText)findViewById(R.id.numb);
        sum=(TextView)findViewById(R.id.sum);
        theList= (ListView) findViewById(R.id.theList);
        StrictMode.ThreadPolicy         policy         =         new
StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);


        Bprint.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                new getPrint().execute(new exterminatusCon());
                exterminatusAdapter.total=0.00;

            }
        });

        Bpay.setOnClickListener(new View.OnClickListener() {

            InputStream is=null;
            @Override
            public void onClick(View v) {
```

```
                Bpay.setVisibility(View.INVISIBLE);
                Log.d("click",
String.valueOf(exterminatusAdapter.total));
                exterminatusAdapter.total=0.00;

Log.d("click",String.valueOf(exterminatusAdapter.total));
                try {
                    HttpClient client = new DefaultHttpClient();

                    HttpPost postm = new HttpPost(gl.exterminate());
                    //dimioyrgw  lista  typou  "name+key"  kai  pernaw
tis metablites
                    List<NameValuePair>       list       =       new
ArrayList<NameValuePair>(1);
                    list.add(new
BasicNameValuePair("pass",gl.getPass()));
                    list.add(new           BasicNameValuePair("art",
trapN.getText().toString()));

                    postm.setEntity(new
UrlEncodedFormEntity(list));//kwdikopoiw
                    postm.setHeader(HTTP.CONTENT_TYPE,
"application/x-www-form-urlencoded;charset=UTF-8");//4   no   reason
@all, alla to xreiazetai

                    HttpResponse response = client.execute(postm);

                    HttpEntity entity = response.getEntity();

                    is = entity.getContent();

                } catch (ClientProtocolException e) {
                    e.printStackTrace();
                } catch (IOException e) {
                    e.printStackTrace();
                }
                new getPrint().execute(new exterminatusCon());
                sum.setText("0.00");

            }

        });

}
    public void setListAdapter(JSONArray array) {
        this.theList.setAdapter(new       exterminatusAdapter(array,
this));
    }
```

```
    public class getPrint extends
AsyncTask<exterminatusCon,Long,JSONArray> {

        @Override
        protected JSONArray doInBackground(exterminatusCon...
params) {
            return params[0].printData();
        }

        @Override
        protected void onPostExecute(JSONArray jsonArray) {
            setListAdapter(jsonArray);
        }


    }
}
```

exterminatusCon.java:

```
package gr.greg.ptuxiaki;

import android.util.Log;

import org.json.JSONArray;
import org.json.JSONException;

import java.io.IOException;
import java.io.InputStream;
import java.util.ArrayList;
import java.util.List;

import cz.msebera.android.httpclient.HttpEntity;
import cz.msebera.android.httpclient.HttpResponse;
import cz.msebera.android.httpclient.NameValuePair;
import cz.msebera.android.httpclient.client.ClientProtocolException;
import cz.msebera.android.httpclient.client.HttpClient;
import
cz.msebera.android.httpclient.client.entity.UrlEncodedFormEntity;
import cz.msebera.android.httpclient.client.methods.HttpPost;
import cz.msebera.android.httpclient.impl.client.DefaultHttpClient;
import cz.msebera.android.httpclient.message.BasicNameValuePair;
import cz.msebera.android.httpclient.protocol.HTTP;
import cz.msebera.android.httpclient.util.EntityUtils;

/**
```

```
 * Created by Greg on 16/4/2016.
 */


public class exterminatusCon {
    Global gl=new Global();
    InputStream is=null;

    public JSONArray printData(){
        HttpEntity httpEntity=null;


        try
        {
            HttpClient client = new DefaultHttpClient();

            HttpPost postm = new HttpPost(gl.print());
            //dimioyrgw lista typou "name+key" kai pernaw tis
metablites
            List<NameValuePair> list = new
ArrayList<NameValuePair>(1);
            list.add(new BasicNameValuePair("pass",gl.getPass()));
            list.add(new BasicNameValuePair("art",
exterminatus.trapN.getText().toString()));
            //permw ap'eu8eias ton ari8mo apo thn klasi
exterminatus.java

            postm.setEntity(new
UrlEncodedFormEntity(list));//kwdikopoiw
            postm.setHeader(HTTP.CONTENT_TYPE, "application/x-www-
form-urlencoded;charset=UTF-8");//4 no reason @all, alla to
xreiazetai

            HttpResponse response = client.execute(postm);

            httpEntity = response.getEntity();

        } catch (ClientProtocolException e) {

            // An uparxei problima sto http protocol
            e.printStackTrace();


        } catch (IOException e) {
            e.printStackTrace();
        }
```

```
        // Metatropi HttpEntity se JSON Array
        JSONArray jsonArray = null;

        if (httpEntity != null) {
            try {
                String                entityResponse                =
EntityUtils.toString(httpEntity);

                Log.e("Entity Response  : ", entityResponse);

                jsonArray = new JSONArray(entityResponse);

            } catch (JSONException e) {
                e.printStackTrace();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }

        return jsonArray;
    }
}
```

exterminatusAdapter.java:

```
package gr.greg.ptuxiaki;

import android.app.Activity;
import android.content.Context;
import android.os.StrictMode;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.TextView;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

/**
 * Created by Greg on 16/4/2016.
 */


public class exterminatusAdapter extends BaseAdapter {

    String euro = "\u20ac";
```

```
    private JSONArray dataArray;
    private Activity act;
    private LayoutInflater inflater = null;
    static public Double total = 0.0;

    public exterminatusAdapter(JSONArray array, Activity ab) {
        this.dataArray = array;
        this.act = ab;
        inflater                  =                  (LayoutInflater)
this.act.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    }

    @Override
    public int getCount() {
        try {
            return this.dataArray.length();
        } catch (NullPointerException e) {
            return 0;
        }
    }

    @Override
    public Object getItem(int position) {
        return position;
    }

    @Override
    public long getItemId(int position) {
        return position;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup
parent) {
        ListCell cell;
        StrictMode.ThreadPolicy         policy         =         new
StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);


        if (convertView == null) {

            convertView                                          =
inflater.inflate(R.layout.exterminatus_help, null);
            cell = new ListCell();

            cell.name                  =                  (TextView)
convertView.findViewById(R.id.name);
```

```
            cell.timi                    =                    (TextView)
convertView.findViewById(R.id.timi);
            convertView.setTag(cell);
        } else {
            cell = (ListCell) convertView.getTag();
        }

        //fortosi dedomenwn se lista

        try {
            JSONObject jOb = this.dataArray.getJSONObject(position);
            cell.name.setText(jOb.getString("name"));
            cell.timi.setText(jOb.getString("timi"));
            total += jOb.getDouble("timi");

        } catch (JSONException e) {
            e.printStackTrace();
        }
        if (total > 0.01) {
            exterminatus.Bpay.setVisibility(View.VISIBLE);
            exterminatus.sum.setText(String.format("%.2f",  total  /
2) + euro);
        } else {
            exterminatus.sum.setText("0.00");
            exterminatus.Bpay.setVisibility(View.INVISIBLE);
        }

        return convertView;
    }
}
```

Activity newUser:

XML:

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context="gr.greg.ptuxiaki.newUser"
    tools:showIn="@layout/activity_new_user">
```

```
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/Pass"
    android:layout_marginTop="186dp"
    android:background="#ffffff"
    android:layout_alignParentTop="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="40dp"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:text="Password Strength: "
    android:id="@+id/textView"
    android:layout_below="@+id/Pass"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_marginLeft="42dp"
    android:layout_marginStart="42dp"
    android:textColor="#000000" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:id="@+id/Str"
    android:layout_alignTop="@+id/textView"
    android:layout_toRightOf="@+id/textView"
    android:layout_toEndOf="@+id/textView"
    android:text="Not Inserted"
    android:textColor="#000000" />

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/secP"
    android:background="#ffffff"
    android:layout_below="@+id/textView3"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_marginTop="15dp" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

```
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:id="@+id/match"
    android:layout_below="@+id/secP"
    android:layout_centerHorizontal="true" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Sign up"
    android:id="@+id/button"
    android:layout_marginTop="63dp"
    android:clickable="true"
    android:enabled="false"
    android:layout_below="@+id/match"
    android:layout_centerHorizontal="true" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:text="Password"
    android:id="@+id/textView2"
    android:layout_alignBottom="@+id/Pass"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_marginBottom="20dp" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:text="Retype Password"
    android:id="@+id/textView3"
    android:layout_below="@+id/textView"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/userN"
    android:layout_above="@+id/textView2"
    android:background="#ffffff"
    android:layout_marginBottom="30dp" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

```
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:text="Username"
        android:id="@+id/textView4"
        android:layout_above="@+id/userN"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />
</RelativeLayout>
```

Java:

```
package gr.greg.ptuxiaki;

import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.os.StrictMode;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.text.Editable;
import android.text.TextWatcher;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import java.io.BufferedWriter;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.io.UnsupportedEncodingException;
import java.net.HttpURLConnection;
import java.net.ProtocolException;
import java.net.URL;

public class newUser extends AppCompatActivity {

    EditText pass,sec,userN;
    TextView stre,match;
    Button butt;
    Integer num;
    URL url;
    Global gl= new Global();
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_new_user);

        StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);//energopoioume to strict
mode gia na mhn xrisimopoihsoume kati p den uparxei

        pass= (EditText) findViewById(R.id.Pass);
        sec= (EditText) findViewById(R.id.secP);
        userN= (EditText) findViewById(R.id.userN);
        stre= (TextView) findViewById(R.id.Str);
        match= (TextView) findViewById(R.id.match);
        butt=(Button) findViewById(R.id.button);
        num=0;


        pass.addTextChangedListener(new TextWatcher() {
            @Override
            public void beforeTextChanged(CharSequence s, int start,
int count, int after) {

            }

            @Override
            public void onTextChanged(CharSequence s, int start, int
before, int count) {
                num=s.toString().length();
            }

            @Override
            public void afterTextChanged(Editable s) {

                if(num<=4){ stre.setText("Too weak");}

                if(num>4){ stre.setText("Weak");}

                if(num>6){ stre.setText("Medium");}

                if(num>=10){ stre.setText("Strong");}
            }
        });

        sec.addTextChangedListener(new TextWatcher() {
            @Override
            public void beforeTextChanged(CharSequence s, int start,
int count, int after) {
```

```
            }

            @Override
            public void onTextChanged(CharSequence s, int start, int
before, int count) {

            }

            @Override
            public void afterTextChanged(Editable s) {

if(pass.getText().toString().equals(sec.getText().toString())){
                    match.setText("Match");
                    if (num>4){
                        butt.setEnabled(true);
                    }

                }
                else{
                    match.setText("Mismatch");
                    butt.setEnabled(false);}
            }
        });

        butt.setOnClickListener(new View.OnClickListener() {
            OutputStream os= null;
            InputStream is=null;
            @Override
            public void onClick(View v) {
                String name = userN.getText().toString();
                String pwd = pass.getText().toString();

                // Stelnw dedomena me methodo POST
                HttpURLConnection conn = null;
                try {
                    url = new URL(gl.sign());
                    conn = (HttpURLConnection) url.openConnection();
                    conn.setDoInput(true);
                    conn.setDoOutput(true);
                    conn.setUseCaches(false);
                    os = conn.getOutputStream();
                } catch (IOException e) {
                    e.printStackTrace();
                }
                try {
                    if (conn != null) {
                        conn.setRequestMethod("POST");
```

```
                    conn.setRequestProperty("Content-Type",
"application/x-www-form-urlencoded");
                }


            } catch (ProtocolException e) {
                e.printStackTrace();
            }
//                          conn.setRequestProperty("Content-Type",
"application/x-www-form-urlencoded"); //den kserw ti einai alla to
xreiazetai
            // bazw parametrous kai times gia to arxeio
            Uri.Builder builder = new Uri.Builder()
                    .appendQueryParameter("pwd", pwd)
                    .appendQueryParameter("userN", name);

            //kwdikopoiw
            String query = builder.build().getEncodedQuery();

            try {
                os = conn.getOutputStream();

            } catch (IOException e) {
                e.printStackTrace();
            }
            BufferedWriter writer=null;
            try {
                writer      =      new      BufferedWriter(new
OutputStreamWriter(os, "UTF-8"));
            } catch (UnsupportedEncodingException e) {
                e.printStackTrace();
            }
            //pernaw tis metablites sto php arxeio
            try {
                writer.write(query);
                writer.flush();
                Toast.makeText(newUser.this,         "Inserted",
Toast.LENGTH_SHORT).show();
                writer.close();
                os.close();
                Intent   intent   =   new   Intent(newUser.this,
MainActivity.class);
                startActivity(intent);
            } catch (IOException e) {
                e.printStackTrace();
                Toast.makeText(newUser.this,  "Error  Occurred",
Toast.LENGTH_SHORT).show();
            }
```

```
                try {
                    is = conn.getInputStream();
                    int ch;
                    StringBuffer sb = new StringBuffer();
                    while ((ch = is.read()) != -1) {
                        sb.append((char) ch);
                    }
                    Log.d("response", sb.toString());
                    if (is != null)
                        is.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }

        });
    }

}
```

Τέλος για τις ανάγκες τις εργασίας, δημιουργήθηκαν και δύο βοηθητικές κλάσεις, οι οποίες καλούνται εντός του κώδικα όποτε χρειαστούν. Δεν περιλαμβάνουν οπτικό περιβάλλον, παρά μόνο μεθόδους.

Class Global:

```
package gr.greg.ptuxiaki;


import java.net.MalformedURLException;
import java.net.URL;


//exw tis ka8olikes metablites
public class Global {

    private String strUrl = "http://192.168.2.117/";
    private URL url;
    private String pass = "fantomakos";

    public URL getPostUrl() {
        try {
            url = new URL(strUrl + "S_post.php");
        } catch (MalformedURLException e) {
            e.printStackTrace();
        }
        return url;
```

```
}

public String getSpinUrl() {
    return strUrl + "spinpop.php";
}

public String getPass() {
    return pass;
}

public URL getNotReady() {
    try {
        url = new URL(strUrl + "S_getNotReady.php");
    } catch (MalformedURLException e) {
        e.printStackTrace();
    }
    return url;
}

public URL login() {
    try {
        url = new URL(strUrl + "C_login.php");
    } catch (MalformedURLException e) {
        e.printStackTrace();
    }
    return url;
}

public String postRdy() {
    return strUrl + "S_postDone.php";
}

public String destroyNR() { //Not READY (yet)!!
    return strUrl + "S_exterminateUndone.php";
}

public URL getReady() {
    try {
        url = new URL(strUrl + "S_getReady.php");
    } catch (MalformedURLException e) {
        e.printStackTrace();
    }
    return url;
}

public String postGone() {
    return strUrl + "S_postGone.php";
}
```

```
    public String exterminate() { return strUrl + "S_Exterminatus.php"; }

    public String print(){return strUrl+ "S_print.php"; }

    public String sign(){return strUrl + "newUser.php"; }
       }
```

Class ListCell.java:

```
package gr.greg.ptuxiaki;

import android.widget.Button;
import android.widget.TextView;

/**
 * Created by Greg on 12/3/2016.
 */
public class ListCell {
    //mini-class gia boh8eia sthn taksinoisi twn ontotitwn
     TextView noTrap,name,zaxari,timi;
    int isDone=0;
    Button Bdone,Bcan; }
```

7. Συμπεράσματα

Μέρα με την ημέρα οι χρήστες υπολογιστών και κινητών συσκευών αυξάνοται σημαντικά. Μαζί με αυτή την αύξηση, αυξάνονται οι εφαρμογές που μπορεί να κάνει χρήση κάποιος άνθρωπος, όπως και τα δεδομένα που χρησιμοιποιεί. Μερικά από αυτά τα προσωπικά και ευαίσθητα δεδομένα που υπάρχουν σε αυτές τις συσκευές, θα ήταν άκρως χρήσιμα σε άτομα τα οποία θέλουν είτε να μας βλάψουν, είτε γνωρίζουν άτομα που θέλουν να μας βλάψουν και απλώς θέλουν να βγάλουν κέρδος. Υπάρχουν πολλές επιλογές για το πως μπορεί κάποιος να κλέψει δεδομένα από έναν άλλο χρήστη. Η νούμερο ένα απειλή είναι να κάνει επίθεση εισαγωγής κώδικα, ή πιο απλά SQL injection, με ελπίδα ότι θα υπάρχει κάποια τρύπα ασφαλείας ανοιχτή, που θα ξέχασε κάποιος προγραμματιστής να κλείσει ή η εταιρία δεν θα την έχει παρατηρήσει. Υπάρχουν όμως και αρκετοί τρόποι για να αποφευχθούν αυτές οι επιθέσεις, όμως λόγο της μεγάλης ζήτησης εφαρμογών από τον κόσμο, πολλοί προγραμματιστές, αφήνουν κομμάτια κώδικα με τρύπες ασφαλείας, με σκοπό να τις κλείσουν αργότερα και να βάλουν κάποιο έξτρα περιεχόμενο. Δεν υπάρχει καμία πολιτική για το πώς ή πότε θα πρέπει, ένας προγραμματιστής, να δώσει προσοχή στην ασφάλεια των δεδομένων. Κάθε ένας προγραμματιστής έχει την επιλογή και το βάρος να χρησιμοποιήσει τεχνικές αποτροπής επιθέσεων από μόνος του, όπου και όποτε μπορεί. Υπάρχουν όμως και αρκετοί προγραμματιστές που δεν γνωρίζουν από ασφάλεια δεδομένων. Αυτοί οι προγραμματιστές, είναι που αφήνουν τις περισσότερες τρύπες ασφαλείας. Αυτό οφείλετε στο ότι, δεν έχουν τις απαραίτητες γνώσεις και ότι δεν μπόρεσαν να δούνε την τρύπα ασφαλείας που υπήρχε στην εφαρμογή, ή το πρόγραμμά τους, ή απλά δεν τους ενδιέφερε να κλείσουν την τρύπα. Αρκετές δημοφιλής εφαρμογές, πιστεύω, ότι έχουν ανοιχτές πίσω πόρτες για να μπορεί κάποιος από τους προγραμματιστές, ή διαφορετικό πρόσωπο, να δει ή να αλλάξει δεδομένα τις εφαρμογής. Μία δημοφιλής τεχνική για κλείσιμο τρυπών ασφαλείας, που υιοθετήθηκε από μεγάλες εταιρίες, πρόσφατα, είναι η προσέγγιση ατόμων που έχουν εμπειρία στην εύρεση bug ή την ασφάλεια εφαρμογών και προγραμμάτων, γνωστοί ως bug bounty hunterss. Με χρηματικό, κατά κύριο λόγο, ή άλλο κίνητρο αυτοί οι “κυνηγοί” βρίσκουν, bugs και τρύπες ασφαλείας για οποιαδήποτε εφαρμογή. Επίσης αρκετές εταιρίες, χρησιμοποιούν αυτούς τους κυνηγούς για να δείξουν ότι η εφαρμογή τους είναι απόρθητη και χωρίς κανένα πρόβλημα κώδικα. Για μία καλύτερη εμπειρία στον κόσμο της πληροφορικής και μια ζωή χωρίς φόβο για τυχών κλοπές δεδομένων, που ίσως μας επιβαρύνουν, θα ήταν καλό να μάθουν όλοι μερικές βασικές τεχνικές, για την αποφυγή και την καταπολέμηση των περισσότερο διαδεδομένων τύπων επιθέσεων με σκοπό την απόσπαση πληροφοριών.

Αναφορές

(1) Pew Research Center. U.S. Smartphone Use in 2015, Απρίλιος 2015 http://www.pewinternet.org/2015/04/01/us-smartphone-use-in-2015/

(2) Somansa. Data Loss Prevention, 2013 http://www.somansatech.com/2013/solutions/solutions_07.php

(3) Yarrow, J. ,This Chart Shows Google's Incredible Domination Of The World's Computing Platforms. Μαρτιος 2014

(4) 1+ Greece. Android και Ασφάλεια. http://www.oneplusgreece.com/android-and-security/

(5) Καθημερινή. ESET: Συμβουλές ασφάλειας για τα Android smartphone http://www.kathimerini.gr/773368/article/texnologia/thlefwnia/eset-symvoyles-asfaleias-gia-ta-android-smartphone

(6) University of Torronto. Permission to Spy: An Analysis of Android Malware Targeting Tibetans Απρίλιος 2013 // https://citizenlab.org/2013/04/permission-to-spy-an-analysis-of-android-malware-targeting-tibetans/

(7) Android Enthusiasts. Google play store does not show all persmissions at app install http://android.stackexchange.com/questions/82073/google-play-store-does-not-show-all-persmissions-at-app-install

(8) Steve Christey. 2011 CWE/SANS Top 25 Most Dangerous Software Errors https://cwe.mitre.org/top25/index.html

(9) Jeff Williams. Injection Theory // https://www.owasp.org/index.php/Injection_Theory

(10) moufid. Exploit SQL Injection Using Sqlmap in Kali Linux https://codingsec.net/2016/04/sql-injection/

(11) Neil DuPaul, SQL Injection Cheat Sheet & Tutorial: Vulnerabilities & How to Prevent SQL Injection Attacks. // http://www.veracode.com/security/sql-injection

(12) Godfrey Nolan, Practical Advice for Building Secure Android Databases in SQLite Ιανουάριος 2015

(13) OWASP, 26 Απριλίου 2016 https://www.owasp.org/index.php/Testing_for_SQL_Injection_(OTG-INPVAL-005)

(14) Oracle https://docs.oracle.com/cd/A91202_01/901_doc/appdev.901/a89857/oci05bnd.htm

(15) php.net, History of PHP http://php.net/manual/en/history.php.php

(16) php.net, intval http://php.net/manual/en/function.intval.php

(17) php.net, mysql_real_escape_string http://php.net/manual/en/function.mysql-real-escape-string.php

(18) php.net, Introduction
http://php.net/manual/en/intro.pdo.php
(19) php.net, Predefined Constants
http://php.net/manual/en/pdo.constants.php
(20) wikipedia, Collision (computer science) Σεπτέμβριος 2015
(21) S.S Thomsen, Cryptographic hash functions Νοέμβριος 2008
http://orbit.dtu.dk/fedora/objects/orbit:82593/datastreams/file_5025771/content
(22) Michael Barr, CRC Series, Part 1: Additive Checksums, Νοέμβριος 1999
http://www.barrgroup.com/Embedded-Systems/How-To/Additive-Checksums
(23) Εφαρμογή #ashing
http://hashingapp.github.io/
(24) Steve Friedl, An Illustrated Guide to Cryptographic Hashes Μάιος 2005
http://www.unixwiz.net/techtips/iguide-crypto-hashes.html
(25) Dr. Richard J. Enbody, Computer and Network Security Ιούνιος 2015
http://www.cse.msu.edu/~cse825/lectures/06_crypto_hash_func.pdf
(26) R. Rivest, The MD4 Message Digest Algorithm Οκτώβριος 1990
http://tools.ietf.org/html/rfc1186
(27) Bruce Schneier, Cryptanalysis of MD5 and SHA: Time for a New Standard Αυγουστος 2004
https://www.schneier.com/essays/archives/2004/08/cryptanalysis_of_md5.html
(28) Smriti Gupta & Prof. Sandeep Kumar Yadav, A Critical Review of Cryptographic Hash Functions Σεπτέμβριος 2015
http://www.ijarcce.com/upload/2015/september-15/IJARCCE%20103.pdf
(29) Xiaoyun Wang, Hongbo Yu, How to Break MD5 and Other Hash Functions σελίδες 19-35, Μάιος 2005
http://merlot.usc.edu/csac-f06/papers/Wang05a.pdf
(30) Bruce Schneier, Cryptanalysis of SHA-1, Φεβρουάριος 2005
https://www.schneier.com/blog/archives/2005/02/cryptanalysis_o.html
(31) OWASP, Projects/OWASP Mobile Security Project - Top Ten Mobile Risks, Φεβρουάριος 2016
https://www.owasp.org/index.php/Projects/OWASP_Mobile_Security_Project_-_Top_Ten_Mobile_Risks
(32) Κ. Μάγκος , Α. Νιξαρλίδης, Ασφάλεια στο διαδύκτιο, Ιούλιος 1999
http://www.islab.demokritos.gr/gr/html/ptixiakes/kostas-aris_ptyxiakh/Phtml/kruptografia.htm
(33) Google, Meet Android Studio
https://developer.android.com/studio/intro/index.html

(34) Apache friends, XAMPP Apache + MariaDB + PHP + Perl
https://www.apachefriends.org/index.html
(35) Sublime, Sublime
http://www.sublimetext.com/